University of Mumbai

वेबसाईट – mu.ac.in
ईमेल - आयडी - dr.aams@fort.mu.ac.in
aams3@mu.ac.in

विद्याविषयक प्राधिकरणे
सभा आणि सेवा विभाग(ए.ए.एम.एस)
रूम नं. १२८ एम.जी.रोड, फोर्ट,
मुंबई – ४०० ०३२
टेलिफोन नं - ०२२ - ६८३२००३३

(नॅक पुनर्मूल्यांकनाद्वारे ३.६५ (सी.जी.पी.ए.) सह अ++ श्रेणी
विद्यापीठ अनुदान आयोगाद्वारे श्रेणी १ विद्यापीठ दर्जा)

क्र.वि.प्रा.स.से./आयसीडी/२०२५-२६/३७ दिनांक : २७ मे, २०२५

परिपत्रक:-

सर्व प्राचार्य/संचालक, संलग्नित महाविद्यालये/संस्था, विद्यापीठ शैक्षणिक विभागांचे संचालक/ विभाग प्रमुख यांना कळविण्यात येते की, राष्ट्रीय शैक्षणिक धोरण २०२० च्या अमंलबजावणीच्या अनुषंगाने **शैक्षणिक वर्ष २०२५-२६** पासून **पदवी व पदव्युत्तर अभ्यासक्रम** विद्यापरिषदेच्या दिनांक २८ मार्च २०२५ व २० मे, २०२५ च्या बैठकीमध्ये मंजूर झालेले सर्व अभ्यासक्रम मुंबई विद्यापीठाच्या www.mu.ac.in या संकेत स्थळावर NEP २०२० या टॅब वर उपलब्ध करण्यात आलेले आहेत.

(डॉ. प्रसाद कारंडे)
कुलसचिव

मुंबई – ४०० ०३२
२७ मे, २०२५

क्र.वि.प्रा.स.से.वि/आयसीडी/२०२५-२६/३७ दिनांक : २७ मे, २०२५
Desktop/ Pritam Loke/Marathi Circular/NEP Tab Circular

Copy forwarded for information and necessary action to :-	
1	The Deputy Registrar, (Admissions, Enrolment, Eligibility and Migration Dept)(AEM), dr@eligi.mu.ac.in
2	The Deputy Registrar, Result unit, Vidyanagari drresults@exam.mu.ac.in
3	The Deputy Registrar, Marks and Certificate Unit,. Vidyanagari dr.verification@mu.ac.in
4	The Deputy Registrar, Appointment Unit, Vidyanagari dr.appointment@exam.mu.ac.in
5	The Deputy Registrar, CAP Unit, Vidyanagari cap.exam@mu.ac.in
6	The Deputy Registrar, College Affiliations & Development Department (CAD), deputyregistrar.uni@gmail.com
7	The Deputy Registrar, PRO, Fort, (Publication Section), Pro@mu.ac.in
8	The Deputy Registrar, Executive Authorities Section (EA) eau120@fort.mu.ac.in He is requested to treat this as action taken report on the concerned resolution adopted by the Academic Council referred to the above circular.
9	The Deputy Registrar, Research Administration & Promotion Cell (RAPC), rapc@mu.ac.in
10	The Deputy Registrar, Academic Appointments & Quality Assurance (AAQA) dy.registrar.tau.fort.mu.ac.in ar.tau@fort.mu.ac.in
11	The Deputy Registrar, College Teachers Approval Unit (CTA), concolsection@gmail.com
12	The Deputy Registrars, Finance & Accounts Section, fort draccounts@fort.mu.ac.in
13	The Deputy Registrar, Election Section, Fort drelection@election.mu.ac.in
14	The Assistant Registrar, Administrative Sub-Campus Thane, thanesubcampus@mu.ac.in
15	The Assistant Registrar, School of Engg. & Applied Sciences, Kalyan, ar.seask@mu.ac.in
16	The Assistant Registrar, Ratnagiri Sub-centre, Ratnagiri, ratnagirisubcentar@gmail.com
17	The Director, Centre for Distance and Online Education (CDOE), Vidyanagari, director@idol.mu.ac.in
18	Director, Innovation, Incubation and Linkages, Dr. Sachin Laddha pinkumanno@gmail.com
19	Director, Department of Lifelong Learning and Extension (DLLE), dlleuniversityofmumbai@gmail.com

Copy for information :-	
1	P.A to Hon'ble Vice-Chancellor, vice-chancellor@mu.ac.in
2	P.A to Pro-Vice-Chancellor pvc@fort.mu.ac.in
3	P.A to Registrar, registrar@fort.mu.ac.in
4	P.A to all Deans of all Faculties
5	P.A to Finance & Account Officers, (F & A.O), camu@accounts.mu.ac.in

To,

1	The Chairman, Board of Deans pvc@fort.mu.ac.in
2	**Faculty of Humanities,** **Offg. Dean** 1. Prof.Anil Singh Dranilsingh129@gmail.com **Offg. Associate Dean** 2. Prof.Manisha Karne mkarne@economics.mu.ac.in 3. Dr.Suchitra Naik Naiksuchitra27@gmail.com
	Faculty of Commerce & Management, **Offg. Dean,** 1 Prin.Ravindra Bambardekar principal@model-college.edu.in **Offg. Associate Dean** 2. Dr.Kavita Laghate kavitalaghate@jbims.mu.ac.in 3. Dr.Ravikant Balkrishna Sangurde Ravikant.s.@somaiya.edu 4. Prin.Kishori Bhagat kishoribhagat@rediffmail.com

	Faculty of Science & Technology **Offg. Dean** 1. Prof. Shivram Garje ssgarje@chem.mu.ac.in **Offg. Associate Dean** 2. Dr. Madhav R. Rajwade Madhavr64@gmail.com 3. Prin. Deven Shah sir.deven@gmail.com
	Faculty of Inter-Disciplinary Studies, **Offg. Dean** 1.Dr. Anil K. Singh aksingh@trcl.org.in **Offg. Associate Dean** 2.Prin.Chadrashekhar Ashok Chakradeo cachakradeo@gmail.com 3. Dr. Kunal Ingle drkunalingle@gmail.com
3	Chairman, Board of Studies,
4	The Director, Board of Examinations and Evaluation, dboee@exam.mu.ac.in
5	The Director, Board of Students Development, dsd@mu.ac.in DSW direcotr@dsw.mu.ac.in
6	The Director, Department of Information & Communication Technology, director.dict@mu.ac.in

AC – 20/5/2025
Item No. – 6.26 (N)

As Per NEP 2020

University of Mumbai

<table>
<tr><td colspan="3">Syllabus for Major
Vertical – 1, 4, 5 & 6</td></tr>
<tr><td colspan="3">Name of the Programme – B.E. (Computer Science & Engineering - Artificial Intelligence and Machine Learning)</td></tr>
<tr><td colspan="3">Faculty of Engineering</td></tr>
<tr><td colspan="3">Board of Studies in (Computer Science & Engineering - Artificial Intelligence and Machine Learning)</td></tr>
<tr><td>U.G. Second Year Programme</td><td>Exit Degree</td><td>U.G. Diploma in Computer Science and Engineering - Artificial Intelligence and Machine Learning.</td></tr>
<tr><td colspan="2">Semester</td><td>III & IV</td></tr>
<tr><td colspan="2">From the Academic Year</td><td>2025-26</td></tr>
</table>

University of Mumbai

(As per NEP 2020)

Sr.No.	Heading	Particulars
1	Title of program O:____________________	B.E. (Computer Science & Engineering - Artificial Intelligence and Machine Learning)
2	Exit Degree	U.G. Diploma in Computer Science and Engineering - Artificial Intelligence and Machine Learning.
3	Scheme of Examination R: ____________	NEP 40% Internal 60% External, Semester End Examination Individual Passing in Internal and External Examination
4	Standards of Passing R: ______________	40%
5	Credit Structure R. TEU-535C R. TEU-535D	Attached herewith
6	Semesters	Sem. III & IV
7	Program Academic Level	5.00
8	Pattern	Semester
9	Status	New
10	To be implemented from Academic Year	2025-26

Sd/-
Dr. Subhash K. Shinde
BoS Chairman, Computer Engineering
Faculty of Science & Technology

Sd/-
Dr. Deven Shah
Associate Dean
Faculty of Science & Technology

Sd/-
Prof. Shivram S. Garje
Dean
Faculty of Science & Technology

Preamble

To meet the challenge of ensuring excellence and NEP 2020 policy in engineering education, the issue of quality needs to be addressed, debated, and taken forward systematically. Accreditation is the principal means of quality assurance in higher education. The major emphasis of the accreditation process is to measure the outcomes of the program that is being accredited. In line with this Faculty of Science and Technology (in particular Engineering) of the University of Mumbai has taken the lead in incorporating the philosophy of NEP 2020 education in the process of curriculum development.

The second-year engineering course is a core training program to impart scientific and logical thinking training to learners in general, with a choice of course selection from the program core course, multidisciplinary minor, and vocational skill-enhanced course. Simultaneously, the objectives of NEP 2020 demand nurturing the core program and skills required for the Computer Science and Engineering (Artificial Intelligence and Machine Learning) Branch of engineering in the learner. Keeping this in view, a pool of courses is offered in Core Courses covering fundamentals required to understand core and modern engineering practices and emerging trends in technology. Considering the shift in pedagogy and the convenience of a stress-free learning process, a choice-based subject pool is offered in the coursework under the heads of Computer Science and Engineering (Artificial Intelligence and Machine Learning) in Engineering for open electives and multidisciplinary minor courses in the third and fourth semesters. Essentially, to give a glimpse of trends in the industry under vocational and enhanced skill practices, the pool is offered to nurture and develop creative skills in contemporary industrial practices. Criteria met in the structure is the opportunity for learners to choose the course of their interest in all disciplines.

Program Core Course Cover Computer Science and Engineering (Artificial Intelligence and Machine Learning) core courses. Also, OE and MDM where a pool of subjects are given for selection. Considering the present scenario, diverse choices need to be made available to fulfill the expectation of a learner to aspire for a career in the field of current trends of Technology and interdisciplinary research. Ability enhancement can be achieved in Undergraduate training by giving an objective viewpoint to the learning process and transitioning a learner from a rote learner to a creative professional. for the purpose Design Thinking is introduced in the First Semester to orient a journey learner to become a skilled professional. Considering the NEP-2020 structure of award of Certificate & Diploma at multiple exit-point pools of Vocational skills is arranged for giving exposure to the current Industry practices.

The faculty resolved that course objectives and course outcomes are to be clearly defined for every course so that all faculty members in affiliated higher education institutes understand the depth and approach of the course to be taught, which will enhance the learner's learning process. NEP 2020 grading system enables a much-required shift in focus from teacher-centric to continuous-based learner-centric education since the workload estimated is based on the investment of time in learning and not in teaching. It also focuses on continuous evaluation, which will enhance the quality of education. Credit assignment for courses is based on a 15-week teaching-learning process for NEP 2020, however, the content of courses is to be taught in 12-13 weeks, and the remaining 2-3 weeks are to be utilized for revision, tutorial, guest lectures, coverage of content beyond the syllabus, etc.

There was a concern that in the present system, the second-year syllabus must not be heavily loaded to the learner and it is of utmost importance that the learner entering into the second year of an engineering course should feel at ease by lowering the burden of syllabus and credits. This is necessary for a learner to get accustomed to the new environment of a college and to create a bond between the teacher and the learner. The present curriculum will be implemented for the Second Year of Engineering from the academic year 2054-26. Subsequently, this system will be carried forward for Third Year and Final Year Engineering in the academic years 2026-27, and 2027-28, respectively.

Sd/-	Sd/-	Sd/-
Dr. Subhash K. Shinde	**Dr. Deven Shah**	**Prof. Shivram S. Garje**
BoS Chairman, Computer Engineering	**Associate Dean**	**Dean**
Faculty of Science & Technology	**Faculty of Science & Technology**	**Faculty of Science & Technology**

Under Graduate Diploma in Engineering- Computer Science and Engineering (Artificial Intelligence and Machine Learning).

Credit Structure (Sem. III & IV)

Level	Semester	Major		Minor	OE	VSC, SEC (VSEC)	AEC, VEC, IKS	OJT, FP, CEP, CC,RP	Cum. Cr. / Sem.	Degree/ Cum. Cr.
	R. TEU-535C									
		Mandatory	Electives							
5.0	III	PCC301:3 PCC302:3 PCC303:3 PCC304:3 PCL301: 1 PCL302:1	--	--	OE:2	--	VEC: 2 HSL: 2	CEP: 2	22	UG Diploma 45
	R. TEU-535D									
	IV	PCC401:3 PCC402:3 PCC403:3 PCL401:1 PCL402:1	--	MDM: 4	OE:2	VSEC:2	VEC: 2 EEM:2	--	23	
	Cum Cr.	25	--	4	4	2	2+2+2+2	2	45	

Exit option: Award of UG Diploma in Major and MDM with 90 credits and additional 4 credits core **one** theory subject with 3 credits and **one** lab with 1 credit from one third year from where they want to take Exit degree. Along with theory and practical course student must compulsory do internship for **one month or 160 hours** which internship is equal to 4 credits.

[Abbreviation - OE – Open Electives, VSC – Vocation Skill Course, SEC – Skill Enhancement Course, (VSEC), AEC – Ability Enhancement Course, VEC – Value Education Course, IKS – Indian Knowledge System, OJT – on Job Training, FP – Field Project, CEP – Continuing Education Program, CC – Co-Curricular, RP – Research Project]

Sem. - III

S.E.
Computer Science and Engineering (Artificial Intelligence and Machine Learning)
Scheme

Program Structure for Second Year of Computer Science and Engineering (Artificial Intelligence and Machine Learning)

UNIVERSITY OF MUMBAI (With Effect from 2025-2026)

SEMESTER III

Course Code	Course Description	Teaching Scheme (Contact Hours)			Credit Assigned			
		Theory	Practical	Tutorial	Theory	Tutorial	Practical	Total Credits
2163111	Mathematics for Computer Engineering	2	--	1	2	1	--	3
2163112	Discrete Structures and Graph Theory	3	–	--	3	–	--	3
2163113	Analysis of Algorithm	3	--	--	3	–	--	3
2163114	Computer organization & Architecture	3	--	--	3	–	--	3
2163311	Open Elective	2#	--	--	2	–	--	2
2163115	Analysis of Algorithm Lab	--	2	--	--	–	1	1
2163116	Computer Organization and Architecture Lab	--	2	--	--	–	1	1
2163611	Full Stack Java Programming	--	2*+2	--	--	–	2	2
2993511	Entrepreneurship Development	--	2*+2	---	--	–	2	2
2993512	Environmental Science for Engineers	--	2*+2	--	--	–	2	2
Total		**13**	**16**	**01**	**13**	**01**	**08**	**22**

* Two hours of practical class to be conducted for full class as demo/discussion.

Theory / Tutorial 1 credit for 1 hour and Practical 1 credit for 2 hours

Institute shall offer a course for Open Elective from Science/Commerce/Management stream bucket provided by the University of Mumbai.

#Institute shall offer a course for MDM from other Engineering Boards.

Program Structure for Second Year of Computer Science and Engineering (Artificial Intelligence and Machine Learning)

UNIVERSITY OF MUMBAI (With Effect from 2025-2026)

SEMESTER III

Course Code	Course Description	Examination scheme							
		Internal Assessment Test (IAT)			End Sem. Exam Marks	End Sem. Exam Duration (Hrs)	Term Work (Tw)	Oral & Pract.	Total
		IAT-I	IAT-II	Total (IAT-I) + IAT-II)					
2163111	Mathematics for Computer Engineering	20	20	40	60	2	25	--	15
2163112	Discrete Structures and Graph Theory	20	20	40	60	2	--	--	100
2163113	Analysis of Algorithm	20	20	40	60	2	--	--	100
2163114	Computer organization & Architecture	20	20	40	60	2	--	--	100
2163311	Open Elective	20	20	40	60	2	--	--	100
2163115	Analysis of Algorithm Lab	--	--	--	--	--	25	25	50
2163116	Computer Organization and Architecture Lab	--	--	--	--	--	25	25	50
2163611	Full Stack Java Programming	--	--	--	--	--	50	25	75
2993511	Entrepreneurship Development	--	--	--	--	--	50	--	50
2993512	Environmental Science for Engineers	--	--	--	--	--	50	--	50
Total		**100**	**100**	**200**	**300**	**10**	**225**	**75**	**800**

Program Structure for Second Year of Computer Science and Engineering (Artificial Intelligence and Machine Learning)

UNIVERSITY OF MUMBAI (With Effect from 2025-2026)

SEMESTER IV

Course Code	Course Description	Teaching Scheme (Contact Hours)			Credit Assigned			
		Theory	Practical	Tutorial	Theory	Tutorial	Practical	Total Credits
2164111	Computational Theory	3	--	--	3	--	--	3
2164112	Database Management System	3	–	--	3	--	--	3
2164113	Operating System	3	--	--	3	--	--	3
MDC401	Multidisciplinary minor	3	--	--	3	--	--	3
2164311	Open Elective	2#	--	--	2	--	--	2
2164114	Database Management System Lab	--	2	--	--	--	1	1
2164115	Operating System Lab	--	2	--	--	--	1	1
MDL401	Multidisciplinary minor	--	2	--	--	--	1	1
2164411	Mini Project	--	4	--	--	--	2	2
2994511	Business Model Development	--	2*+2	--	--	--	2	2
2994512	Design Thinking	--	2*+2	--	--	--	2	2
Total		**13**	**18**	**01**	**13**	**01**	**09**	**23**

* Two hours of practical class to be conducted for full class as demo/discussion.

Theory / Tutorial 1 credit for 1 hour and Practical 1 credit for 2 hours

Students must select course for Open Elective from Science/Commerce/Management stream bucket provided by the University of Mumbai.

#Institute shall offer a course for MDM from other Engineering Boards.

Program Structure for Second Year of Computer Science and Engineering (Artificial Intelligence and Machine Learning)

UNIVERSITY OF MUMBAI (With Effect from 2025-2026)

SEMESTER IV

Course Code	Course Description	Examination scheme							
		Internal Assessment Test (IAT)			End Sem. Exam Marks	End Sem. Exam Duration (Hrs)	Term Work (Tw)	Oral & Pract.	Total
		IAT-I	IAT-II	Total (IAT-I) + IAT-II)					
2164111	Computational Theory	20	20	40	60	2	--	--	100
2164112	Database Management System	20	20	40	60	2	--	--	100
2164113	Operating System	20	20	40	60	2	--	--	100
MDC401	Multidisciplinary minor	20	20	40	60	2	--	--	100
2164311	Open Elective	20	20	40	60	2	--	--	100
2164114	Database Management System Lab	--	--	--	--	--	25	25	50
2164115	Operating System Lab	--	--	--	--	--	25	25	50
MDL401	Multidisciplinary minor Lab	--	--	--	--	--	25	--	25
2164411	Mini Project-I	--	--	--	--	--	50	25	75
2994511	Business Model Development	--	--	--	--	--	50	--	50
2994512	Design Thinking	--	--	--	--	--	50	--	50
Total		**100**	**100**	**200**	**300**	**10**	**250**	**75**	**800**

Vertical – 1
Major

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2163111	Mathematics for Computer Engineering	2	-	1	2	-	1	3

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		Test 1	Test 2	Total					
2163111	Mathematics for Computer Engineering	20	20	40	60	2	25	--	125

Rationale :

The goal of this course is to achieve conceptual understanding and to retain the best applied mathematics for Computer Science and Engineering (Artificial Intelligence and Machine Learning) and technology. The syllabus is designed to provide the basic tools of mathematics mainly for the purpose of modelling the Computer Science and Engineering (Artificial Intelligence and Machine Learning) problems mathematically and obtaining solutions. This is engineering mathematics course which mainly deals with topics concern to Computer Science and Engineering (Artificial Intelligence and Machine Learning) and technology.

Course Objectives: Six Course Objectives

1. To introduce concepts and fundamentals Matrix algebra for engineering problems
2. To introduce concepts of Linear and Non-linear programming problems of optimization and its applications.
3. To introduce the concept of modular arithmetic.
4. To enhance the skills to expand Fourier series for periodic functions with various period.
5. To develop the proficiency in statistical techniques arising in engineering applications.
6. To familiarize with the concepts of probability distributions with its applications in engineering and science.

Course Outcomes: Six Course outcomes (Based on Blooms Taxonomy)
On successful completion, of course, learner/student will be able to:

1. Apply the concepts of eigenvalues and eigenvectors in engineering problems.
2. Solve Linear and Non-Linear Programming Problems for optimization of engineering problems.
3. Analyze modular arithmetic for security applications.
4. Expand the periodic function by using the Fourier series for real-life problems and complex engineering problems.
5. Apply the concept of Correlation and Regression to the engineering problems in data science, machine learning, and AI.
6. Apply the concept of probability distribution to engineering problems, mostly used in varied applications in engineering and science.

DETAILED SYLLABUS:

Sr. No	Name of Module	Detailed Content	Hours	CO Mapping
0	**Prerequisite**	Complex Numbers, Probability, Central tendencies and dispersion in Statistical techniques, Baye's theorem, Random variable, Discrete and Continuous random variables.		

I	**Linear Algebra (Theory of Matrices)**	1. Characteristic Equation, Eigenvalues and Eigenvectors, and properties (without proof) 2. Cayley-Hamilton Theorem (without proof), verification and reduction of higher degree polynomials 3. Similarity of matrices, diagonalizable and non-diagonalizable matrices **Self-learning Topics**: Derogatory and non-derogatory matrices, Functions of Square Matrix, Linear Transformations, Quadratic forms.	**5**	**CO1**
II	**Linear and Non-Linear Programming Problems**	1. Types of solutions, Standard and Canonical of LPP, Basic and Feasible solutions, slack variables, surplus variables, Simplex method. 2. NLPP with one and two equality constraint (two or three variables) using the method of Lagrange's multipliers **Self-learning Topics**: Sensitivity Analysis, Big-M method, Artificial variables, Kuhn-Tucker conditions	**5**	**CO2**
III	**Modular Arithmetic**	1. Introduction to Congruence, Linear congruence, reminder theorem, solving polynomials, system of linear congruence 2. Eluer's theorem, Fermat's little theorem, Application of congruence-RSA algorithm. **Self-learning Topics**: Divisiblility, GCD, properties of prime numbers, fundamental theorem of arithmetic.	**4**	**CO3**
IV	**Fourier Series**	1. Dirichlet's conditions, Fourier series of periodic function with period 2π and $2l$. 2. Fourier series of even and odd functions. **Self-learning Topics:** Orthogonal and orthonormal set of functions, Complex form of Fourier Series, Half range Sine and Cosine Series.	**4**	**CO4**
V	**Statistical Techniques**	1. Karl Pearson's coefficient of correlation (r). 2. Spearman's Rank correlation coefficient (R) (with repeated and non-repeated ranks). 3. Lines of regression, fitting of first-degree curves. **Self-learning Topics:** Covariance, Fitting of second-degree and exponential curve.	**4**	**CO5**
VI	**Probability**	1. Moment generating function, Raw moments. 2. Poisson Distribution, Normal Distribution **Self-learning Topics:** Skewness and Kurtosis of distribution (data), types of distribution and their application.	**4**	**CO6**

Books:

1. Higher Engineering Mathematics, Dr. B. S. Grewal, Khanna Publication.

2. Advanced Engineering Mathematics, Erwin Kreyszig, Wiley Eastern Limited.

3. Advanced Engineering Mathematics, R. K. Jain and S. R. K. Iyengar, Narosa publication

4. Probability, Statistics and Random Processes, T. Veerarajan, Mc. Graw Hill education.

5. Number theory, M. G. Nadkarni and J. S. Dani, Tata Mc. Graw Hill education.

Online References:

Sr. No.	Website Name
1.	https://nptel.ac.in/courses/111/104/111104085/
2.	https://nptel.ac.in/courses/111/106/111106139/
3.	https://www.youtube.com/watch?v=2CP3m3EgL1Q
4.	https://www.youtube.com/watch?v=Hw8KHNgRaOE
5.	https://nptel.ac.in/courses/111/105/111105041/

Assessment: Note: Tutorial shall be conducted batch wise

Term Work: General Instructions:

1. Students must be encouraged to write at least 6 class tutorials on entire syllabus.

The tutorials should be conducted batch wise.

2. A group of 4-6 students should be assigned a ***self-learning topic*** to prepare a presentation/problem solving of 10-15 minutes. This should be considered as mini project in Engineering Mathematics.

This project should be graded for 10 marks depending on the performance of the students.

3. The distribution of Term Work marks will be as follows –
    a. Attendance (Theory and Tutorial) : 05 marks
    b. Class Tutorials on entire syllabus : 10 marks
    c. Mini project : 10 marks

Internal Assessment (IA) for 40 marks:

- IA will consist of Two Compulsory Internal Assessment Tests. Approximately 40% to 50% of syllabus content must be covered in First IA Test and remaining 40% to 50% of syllabus content must be covered in Second IA Test.

End Semester Internal Examination for 40 marks:

Question paper format:

1. Question Paper will comprise of a total of **six questions each carrying 20 marksQ.1** will be **compulsory** and should **cover maximum contents of the syllabus**
2. **Remaining questions** will be **mixed in nature** (part (a) and part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module randomly selected from all the modules)
3. A total of **Three questions** needs to be answered.

Course Code	Course Name	Teaching Scheme (Contact Hours)	Credits Assigned

		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2163112	**Discrete Structures and Graph Theory**	**3**	**-**	**-**	**3**	**-**	**-**	**3**

Course Code	Course Name	Theory					Term work	Pract. / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		Test 1	Test 2	Total					
2163112	**Discrete Structures and Graph Theory**	20	20	40	60	2	--	--	100

Rationale:
Mathematics forms the foundation of computer science and engineering. The study of Discrete Structures and Graph Theory enables students to develop strong logical reasoning, combinatorial techniques, and mathematical structures that are essential in programming, algorithm design, networking, database design, artificial intelligence, and cryptography.

Course Objectives:

1) Cultivate clear thinking and creative problem solving.
2) Thoroughly train in the construction and understanding of mathematical proofs. Exercise common mathematical arguments and proof strategies.
3) To apply graph theory in solving practical problems.
4) Thoroughly prepare for the mathematical aspects of other Computer Science and Engineering (Artificial Intelligence and Machine Learning) courses.
5) Solve real-world problems using counting principles, recurrence relations.
6) Strengthen mathematical foundations for research and higher studies in Computer Science and Engineering (Artificial Intelligence and Machine Learning).

Course Outcomes:

1) Understand the notion of mathematical thinking, mathematical proofs and to apply them in problem solving using set theory and logic.
2) Apply properties of Relation sets in real-life problem-solving domains.
3) Apply properties of Function sets in real-life problem-solving domains
4) Apply counting principles, including the Pigeonhole Principle and Inclusion-Exclusion Principle, to solve combinatorial problems.
5) Apply algebraic structure for a given mathematical problem.
6) Apply graph theory in solving computing problems.

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
0	**Prerequisite**	Basic Set Theory, Logical Operators, Truth Tables, Cartesian product, Types of Functions. Basic Algebra and Number Theory, Fundamental Counting Principle, Permutations, Combinations. Graph Basics.	1	
I	**Crisp Set Theory and Logic**	**Set Theory:** Sets, Subsets, Universal and Empty Sets, Set Operations, Set Representation, Laws of Set theory. **Logic:** Propositional Logic, Predicate Logic, Quantifiers (Universal and Existential). **Types of Mathematical Proof:** Direct proof, Proof by contradiction, Proof by deduction, Proof by cases, Proof by exhaustion, Proof by counterexample, Mathematical induction. **Self-learning Topics:** PROLOG / LISP programming to create expert system using Propositional and Predicate Logic, Other types of logic and sets.	7	CO1
II	**Mathematical Relations**	**Relations:** Definition, Representation of Relations, Properties of Relations, Equivalence Relations, Equivalence Classes, Closures of Relations, Warshall's algorithm. **Posets and Lattice:** Partial Order Relations, Poset, Hasse Diagram, Chain and Anti chains, Lattice, Types of Lattices, Sub lattice. **Self-learning Topics:** Practical applications of relations in real life in the field of Database Management, Economics, Social Network, Sports, Medical Diagnosis, Weather, etc.	8	C02
III	**Functions**	**Functions:** Types: Injective, Surjective, and Bijective Functions. Composition, Inverse Functions. Real life applications of Functions. **Self-learning Topics:** Practical applications of function in Neural Network, Determining risk factors for insurance rates, Taxes and tax brackets, Vending machines, etc.	3	CO3
IV	**Counting**	Pigeonhole Principle, Inclusion-Exclusion Principle. Recurrence relations, Solving recurrence relations **Self-learning Topics:** Applications of Recurrence Relations – Analysis of recursive algorithms in computing. Combinatorial Problem Solving – Using counting techniques in probability and decision-making.	5	CO4
V	**Algebraic Structures**	Algebraic structures with one binary operation: Semi group, Monoid, Groups, Subgroups, Abelian Group, and Cyclic group. Algebraic structures with two binary operations: Ring. **Self-learning Topics:** Error Correcting codes.	7	CO5

VI	**Graph Theory**	Types of graphs, Graph Representation, Sub graphs, Operations on Graphs, Walk, Path, Circuit, Connected Graphs, Disconnected Graph, Components, Homomorphism and Isomorphism of Graphs, Euler and Hamiltonian Graphs, Planar Graph, Cut Set, Cut Vertex, Real life applications of Graph Theory. **Self-learning Topics**: Network Flow Problems – Understanding flow in networks and its optimization. Graph Coloring Applications in Scheduling – Use of graph coloring in timetabling and resource allocation. Optimization Techniques – Application of graphs in shortest path problems, spanning trees, and clustering.	8	CO6

Text Books:

1. Susanna S. Epp, "Discrete Mathematics with Applications", 5th Edition, Cengage Publications.
2. Ralph P. Grimaldi, "Discrete and Combinatorial Mathematics", 5th Edition, Pearson Publications.
3. Edgar Goodaire and Michael Parmenter, "Discrete Mathematics and Graph Theory", 3rd Edition, Pearson Publications.

Reference Books:

1. Kenneth A. Ross, "Discrete Mathematics", 5th Edition, Pearson Publications.
2. Swapan Kumar Sarkar, "Textbook of Discrete Mathematics", 9th Edition, S. Chand Publications.
3. Bernad Kolman, Robert Busby, Sharon Cutler Ross, Nadeem-ur-Rehman, "Discrete Mathematical Structures", 6th Edition, Pearson Education.
4. T. Veera Rajan, "Discrete mathematics with Graph Theory and Combinatorics", McGraw Hill Publications.
5. C. L. Liu "Elements of Discrete Mathematics", second edition 1985, McGraw-Hill Book Company. Reprinted 2000

Online References:

Sr. No.	Website Name
1.	https://nptel.ac.in/courses/106106094
2.	https://nptel.ac.in/courses/106108227
3.	https://nptel.ac.in/courses/106106183
4.	https://nptel.ac.in/courses/106103205
5.	https://nptel.ac.in/courses/111107058

Assessment:

- Internal Assessment Test (IAT) for 40 Marks:
  - IA will consist of Two Compulsory Internal Assessment Tests. Approximately 40% to 50% of the syllabus content must be covered in the IAT-I and the remaining 40% to 50% of the syllabus content must be covered in the IAT-II.
- End Semester Theory Examination for 60 Marks:

  Question paper format :
  - Question Paper will comprise a total of six questions each carrying 15 marks Q.1 will be compulsory and should cover the maximum contents of the syllabus.
  - Remaining questions will be mixed in nature (part (a) and part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module randomly selected from all the modules).
  - A total of four questions need to be answered.

Course Code	Course Name	Teaching Scheme (Contact Hours)	Credits Assigned

		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2163113	Analysis of Algorithm	2		-	2	-	-	2

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		Test 1	Test 2	IAT-I + IAT-II (Total)					
2163113	Analysis of Algorithm	20	20	40	60	2	--	--	100

Course Objectives:

1. To provide mathematical approaches for Analysis of Algorithms
2. To understand and solve problems using various algorithmic approaches
3. To analyze algorithms using various methods

Course Outcomes:

1. Evaluate the time and space complexity of algorithms.
2. Implement the Divide and Conquer strategy and assess its complexity.
3. Utilize the Greedy algorithm approach and determine its efficiency.
4. Develop solutions using Dynamic Programming and examine its complexity.
5. Employ Backtracking and Branch and Bound techniques.
6. Apply String Matching algorithms for pattern searching

Prerequisite: Data structure concepts

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
I	**Introduction**	Performance analysis- Master Method, space, and time complexity Growth of function, Big-Oh, Omega Theta notation Mathematical background for algorithm analysis. Analysis of selection sort, insertion sort. **Self-learning Topics:** Complexity class: Definition of P, NP, NP-Hard, NP-Complete	**4**	CO1
II	**Divide and Conquer Approach**	General method, Merge sort, Quick sort, Analysis of Binary search. **Self-learning Topics:** Finding minimum and maximum algorithms and their Analysis, Strassen's Algorithm, real life applications of all algorithms	**5**	CO2
III	**Greedy Method Approach**	General Method, Single source shortest path: Dijkstra AlgorithmFractional Knapsack problem, Minimum cost spanning trees: Kruskal and Prim's algorithms **Self-learning Topics:** Job sequencing with deadlines, real life applications of all algorithms	**5**	CO3
IV	**Dynamic Programming Approach**	General Method, Multistage graphs, All pair shortest path: Floyd Warshall Algorithm, 0/1	**8**	CO4

		knapsack Problem, Travelling Salesperson problem, Longest common subsequence. **Self-learning Topics:** Bellman Ford Algorithm, real life applications of all algorithms		
V	**Backtracking and Branch and bound**	General Method, Backtracking: N-queen problem, Sum of subsets,Graph coloring. Branch and Bound: Travelling Salesperson Problem, 15 Puzzle problem **Self-learning Topics:** Real life applications of all algorithms	**7**	CO1
VI	**String Matching Algorithms**	The Naïve string-matching algorithm, The Rabin Karp algorithm, The Knuth-Morris-Pratt algorithm **Self-learning Topics:** Real life applications of all algorithms	**3**	CO2

Text Books:

1. T. H. Cormen, C.E. Leiserson, R. L. Rivest, and C. Stein, "Introduction to algorithms", 2nd Edition, PHI Publication 2005

2. Ellis Horowitz, Sartaj Sahni, S. Rajsekaran. "Fundamentals of computer algorithms" University Press.

Reference Books:

1. Sanjoy Dasgupta, Christos Papadimitriou, Umesh Vazirani, "Algorithms", Tata McGraw- Hill Edition.
2. S. K. Basu, "Design Methods and Analysis of Algorithm", PHI

Online References:

Sr. No.	Website Name
1.	https://nptel.ac.in/courses/106/106/106106131/
2.	https://swayam.gov.in/nd1_noc19_cs47/preview
3.	https://www.coursera.org/specializations/algorithms
4.	https://www.mooc-list.com/tags/algorithms

Assessment:

- **Internal Assessment (IA) for 40 marks:**

  IA will consist of Two Compulsory Internal Assessment Tests. Approximately 40% to 50% of syllabus content must be covered in First IA Test and remaining 40% to 50% of syllabus content must be covered in Second IA Test

End Semester Examination for 60 Marks:

Question paper format:

- Question Paper will comprise of a total of **six questions each carrying 20 marks Q.1** will be **compulsory** and should **cover maximum contents of the syllabus**

- **Remaining questions** will be **mixed in nature** (part (a) and part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module randomly selected from all the modules)

- A total of **Three questions** needs to be answered.

Course	Course Name	Teaching Scheme	Credits Assigned

Code		(Contact Hours)						
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2163114	Computer Organization & Architecture	3	-	-	3	-	-	3

Course Code	Course Name	Evaluation Scheme (Theory)					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		Test 1	Test 2	Avg.					
2163114	Computer Organization & Architecture	20	20	40	60	2	-	-	100

Pre-requisite.	Fundamental of Mathematics
Course Objective: To study the fundamentals of number system and arithmetic operations. To equip students with the foundational knowledge of computer organization and architecture, fostering an understanding of how hardware and software components collaborate to execute tasks, and preparing them to design and optimize computing systems for real-world applications.	
Course Outcomes (CO): At the End of the course students will be able to	
CO.1	Conceptualize basic computer structure with its models.
CO.2	Design algorithms to solve ALU operations
CO.3	Comprehend processor organization with various control signal design methods of CPU with comparative analysis.
CO.4	Design memory systems with analysis of mapping techniques for cache memory.
CO.5	Explore different types of I/O buses, examine data transfer methods, and assess arbitration techniques for optimized system performance.
CO.6	Analyze different parallel organizations that includes pipelined and parallel processors

DETAIL SYLLABUS:

Sr. No.	Name of the Module	Topics	Hrs	CO Mapping
1	**Computer Fundamentals**	**Number Systems:** Binary. Octal and Hexadecimals. **Binary Number representation:** Sign Magnitude, 1's and 2's Compliment representation. **Logic Gates:** AND, OR, NOT, NAND, NOR, EX-OR. Basic Organization of Computer, Von Neumann model.	4	CO 1
2	**ALU Operations**	**ALU Operation:** Addition and Subtraction on Binary, Octal, Hexadecimal number. Booth's Algorithms, Restoring and Non restoring division algorithm. IEEE 754 Floating point representation and conversation.	8	CO 2
3	**Processor Organization and Control Unit Design**	**8086 Processor**: Architecture of 8086 processor, Register Organization, Instruction formats, instruction cycle, addressing modes. **Control Unit:** Instruction interpretation and sequencing, Micro-programmed and hardwired control unit design methods. Microinstruction sequencing and execution, Micro programs. **RISC and CISC:** Introduction to RISC and CISC architectures and design issues.	8	CO 3

4	**Memory Systems Organization**	Introduction to Memory and Memory parameters. Classifications of primary and secondary memories. Types of RAM and ROM, Memory hierarchy and characteristics, Virtual Memory: Segmentation and Paging Cache memory: Concept, hierarchy (L1, L2, L3), mapping techniques. Cache Coherency and technique to resolve it. Interleaved and Associative memory. **Self-Study** : Case study of Pentium Processor Cache Memory Model (MESI Protocol)	**7**	CO 4
5	**I/O Organization**	Buses: Types of Buses, Bus Arbitration, Bus standards and its comparative study I/O Interface, I/O channels, I/O modules and IO processor, Types of data transfer techniques: Programmed I/O, Interrupt driven I/O and DMA.	**4**	CO 5
6	**Parallel Processing**	Advanced Processor Models(80386DX): Real Model, Protected Model, Virtual Model Pipelined Architecture: Pipeline Stages, Superscalar architecture Pipeline Hazards, Mitigation of Hazards with branch prediction and data forwarding techniques, Amdahl's Law Introduction to parallel processing concepts, Flynn's classifications. **Self-Study:** Superscalar Architecture: Case study of Pentium processor and GPGPU architecture.	**8**	CO 6

Text Books:

Sr. No	**Title**	**Edition**	**Authors**	**Publisher**	**Year**
1	Modern Digital Electronics	4th	R P Jain	Tata McGraw-Hill	2009
2	Computer Organization	5th	Carl Hamacher, Zvonko Vranesic	Tata McGraw-Hill	2002
3	Computer Architecture and Organization	3rd	John P. Hayes	Tata McGraw-Hill	2012
4	Computer Organization and Architecture: Designing for Performance	8th	William Stallings	Pearson	2010
5	Microprocessors and Interfacing	3rd	Douglas V Hall	Tata McGraw-Hill	2017
6	The 80386, 80486, and Pentium Microprocessor: Hardware, Software, and Interfacing	3rd	Walter Triebel	Pearson	1997
7	Pentium Pro Processor System Architecture	3rd	Tom Shanely	Addison Wesley	1996

Reference Books

Sr. No	**Title**	**Edition**	**Authors**	**Publisher**	**Year**
1	Structured Computer Organization	6th	Andrew S. Tanenbaum	Pearson	2012
2	Computer Architecture and Organization: Design Principles and Applications	2nd	B. Govindarajulu	McGraw Hill	Paperback-2017
3	Advance Computer Architecture: Parallelism, Scalability, Programmability	3rd	Kai Hwang	Tata-McGraw Hill	2017

4	Microcomputer System The 8086/8088 family	2nd	Liu and Gibson	Pearson	2015
5	Programmer's reference Manual for IBM Personal Computers	1st	Steven Armburst	Tata-McGraw Hill	

Online References:

Sr. No.	Website Name
1.	https://www.classcentral.com/course/swayam-computer-organization-and-architecture-a-pedagogical-aspect-9824
2.	https://nptel.ac.in/courses/106/103/106103068/
3.	https://www.coursera.org/learn/comparch
4.	https://www.edx.org/learn/computer-architecture

Assessment:

- **Internal Assessment (IA) for 40 marks:**

  IA will consist of Two Compulsory Internal Assessment Tests. Approximately 40% to 50% of syllabus content must be covered in First IA Test and remaining 40% to 50% of syllabus content must be covered in Second IA Test

- **End Semester Examination for 60 Marks:**

  Question paper format:

  - Question Paper will comprise of a total of **six questions each carrying 20 marks Q.1** will be **compulsory** and should **cover maximum contents of the syllabus**
  - **Remaining questions** will be **mixed in nature** (part (a) and part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module randomly selected from all the modules)
  - A total of **Three questions** needs to be answered.

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2163115	Analysis of Algorithm Lab	--	2	-	--	2	-	2

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Practical/ Oral	Total
		Internal assessment			End Sem. Exam			
		Test1	Test 2	Avg. of 2 Tests				
2163115	Analysis of Algorithm Lab	--	--	--	--	25	25	50

Lab Objectives:

1. To introduce the methods of designing and analyzing algorithms
2. Design and implement efficient algorithms for a specified application
3. Strengthen the ability to identify and apply the suitable algorithm for the given real-world problem.
4. Analyze worst-case running time of algorithms and understand fundamental algorithmic problems

Lab Outcomes: At the end of the course, the students will be able to

1. Implement the algorithms using different approaches.
2. Analyze the complexities of various algorithms.
3. Compare the complexity of the algorithms for specific problem.

Prerequisite: Basic knowledge of programming and data structure

DETAILED SYLLABUS:

Sr. No.	Module	Detailed Content	Hours	LO Mapping
0	Prerequisite	Basic knowledge of programming and data structure		
I	Introduction	Performance analysis- Master Method, space, and time complexity Growth of function, Big-Oh, Omega Theta notation Mathematical background for algorithm analysis. Analysis of selection sort, insertion sort.	2	**LO1, LO2, LO3**
II	Divide and Conquer Approach	General method, Merge sort, Quick sort, Analysis of Binary search. **Self-learning Topics:** Finding minimum and maximum algorithms and their Analysis, Strassen's Algorithm	2	**LO1, LO2, LO3**
III	Greedy Method Approach	General Method, Single source shortest path: Dijkstra AlgorithmFractional Knapsack problem, Minimum cost spanning trees: Kruskal and Prim’s algorithms **Self-learning Topics:** Job sequencing with deadlines	2	**LO1, LO2, LO3**
IV	Dynamic Programming	General Method, Multistage graphs	2	**LO1, LO2,**

	Approach	All pair shortest path: Floyd Warshall Algorithm, 0/1 knapsack Problem, Travelling Salesperson problem, Longest common subsequence **Self-learning Topics:** Bellman Ford Algorithm		**LO3**
V	Backtracking and Branch and bound	General Method, Backtracking: N-queen problem, Sum of subsets,Graph coloring Branch and Bound: Travelling Salesperson Problem, 15 Puzzle problem	2	**LO1, LO2, LO3**
VI	String Matching Algorithms	The Naïve string-matching algorithm, The Rabin Karp algorithm, The Knuth-Morris-Pratt algorithm	2	**LO1, LO2, LO3**

Text Books:

1. T. H. Cormen, C.E. Leiserson, R. L. Rivest, and C. Stein, "Introduction to algorithms", 2nd Edition, PHI Publication 2005.
2. Ellis Horowitz, Sartaj Sahni, S. Rajsekaran. "Fundamentals of computer algorithms" University Press.

Reference Books:

1. Sanjoy Dasgupta, Christos Papadimitriou, Umesh Vazirani, "Algorithms", Tata McGraw- Hill Edition.
2. S. K. Basu, "Design Methods and Analysis of Algorithm", PHI

Online Resources:

Sr. No.	Website Name
1.	https://nptel.ac.in/courses/106/106/106106131/
2.	https://swayam.gov.in/nd1_noc19_cs47/preview
3.	https://www.coursera.org/specializations/algorithms
4.	https://www.mooc-list.com/tags/algorithms

Sr No	List of Assignments / Tutorials	Hrs
01	Assignment covers the topics from first three units limited to three Questions	2
02	Assignment covers the topics from Last three units limited to three Questions	2

Suggested list of Experiments.

Sr No	Title of Experiments	Hrs
01	Experiment based on common mathematical functions.(Selection sort, Insertion sort)	2
02	Experiment based on divide and conquers approach. (Merge sort, Quick sort, Binary search)	2
03	Experiment based on greedy approach.(Single source shortest path- Dijkstra Fractional Knapsack problem, Minimum cost spanning trees-Kruskal and Prim's algorithm)	2
04	Experiment using dynamic programming approach (All pair shortest path- Floyd Warshall, 0/1 knapsack)	2
05	Travelling salesperson problem Longest common subsequence	2
06	Experiment based on graph Algorithms (BFS, DFS , etc)	2
07	Experiment using Backtracking strategy. (N-queen problem, Sum of subsets, Graph coloring)	2
08	Experiment using branch and bound strategy.	2
09	Experiment based on string matching/amortized analysis (The Naïve string-matching Algorithms , The Rabin Karp algorithm, The Knuth-Morris-Pratt algorithm.	2
10	Implementation Min-Max Algorithm	2
11	Implementation of Job Sequencing with deadlines.	2
12	Implementation of Bellman Ford Algorithm using Dynamic programming	2

Assessment:

Term Work: Term Work shall consist of at least 10 to 12 practicals' based on the above list. Also, Term work Journal must include at least 2 assignments.

Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)

Practical& Oral Exam: An Oral & Practical exam will be held based on the above syllabus.

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2163116	Computer Organization & Architecture Lab	-	2	-	-	1	-	1

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Practical/ Oral	Total
		Internal assessment			End Sem. Exam			
		Test1	Test 2	Avg. of 2 Tests				
2163116	Computer Organization & Architecture Lab	--	--	--	--	25	25	50

Prerequisite: C/C++ Programming Language.	
Lab Objectives:	
1	To study and learn assembler and using its utilities.(MASM)
2	To write assembly language programs.
3	To perform various ALU operations using assembly language programs.
4	To enable and use graphical mode in assembly language programs.
5	To implement arithmetics operations using algorithms.
6	To implement cache memory mapping techniques.

Lab Outcomes: At the end of the course, student will be able to	
1	To install the MASM.
2	Write assembly language programs.
3	Utilised various utility of INT 21H interrupts.
4	Utilised various utility of INT 10H interrupts.
5	Simulate various algorithms.
6	Simulate varus cache memory mapping techniques.

Suggested List of Experiments:

Sr. No	Title of Experiments	LO
1	Installation and configure: DOS, MASM, Debug and X86 Mode	1
2	Implementation of various ALU operations (ADD, SUB, MUL, DIV, AND, OR, XOR, NOT) through assembly language programming for 8086 using MASM and Debug.	2
3	Implementation of number conversion (HEX to BCD, ASCII to BCD, BCD to ASCII) using MASM.	2
4	Implementation of two 8-bit BCD addition with accepting input from keyboard and displaying output on monitor using INT 21H interrupts.	3
5	Implement various String Operations in 8086 through the utilities provided by DOS and BIOS interrupts (MASM)	2
6	Block Transfer and Block Exchange using Index Registers.	2
7	Drawing basic shapes like rectangle, triangle, etc. using BIOS services [Use C/MASM]	4

8	Design Password Detection Application using BIOS and DOS interrupts along with 8086 instructions.	2
9	Implement file operations [DOS Interrupts in C/MASM]	2
10	Implement I/O interfacing using inbuilt speakers of IBM PC	2
11	Implementation of cursor activity like hiding cursor and changing it to box size using INT 10H interrupts.	4
12	Implement Booth's Multiplication Algorithm	5
13	Implement Division Algorithm (Non-Restoring and/or Restoring)	5
14	Implementation of Mapping techniques of Cache memory	6
15	Displaying 8086 processor's Flag register content on monitor.	2
16	Designing 4X4 memory using 1X1 memory chips. Use COA virtual lab by IIT Kharagpur.	

Text Books:

Sr. No	Title	Edition	Authors	Publisher	Year
1	Computer Organization	5th	Carl Hamacher, Zvonko Vranesic and Safwat Zaky	Tata McGraw-Hill	2002
2	Computer Architecture and Organization	3rd	John P. Hayes	Tata McGraw-Hill	2012
3	Computer Organization and Architecture: Designing for Performance	8th	William Stallings	Pearson	2010
4	Microprocessor and Interfacing: Programming & Hardware	3rd	Douglas V Hall	Tata-McGraw Hill	2017

Reference Books:

Sr. No	Title	Edition	Authors	Publisher	Year
1	Structured Computer Organization	6th	Andrew S. Tanenbaum	Pearson	2012
2	Microcomputer System The 8086/8088 family	2nd	Liu and Gibson	Pearson	2015
3	Computer Architecture and Organization: Design Principles and Applications	2nd	B. Govindarajulu	McGraw Hill	Paperback-2017
4	Advance Computer Architecture: Parallelism, Scalability, Programmability	3rd	Kai Hwang	Tata-McGraw Hill	2017
5	Programmer's reference Manual for IBM Personal Computers	1st	Steven Armburst	Tata-McGraw Hill	

Online References:

Sr. No.	Website Name
1.	https://www.classcentral.com/course/swayam-computer-organization-and-architecture-a-pedagogical-aspect-9824
2.	https://nptel.ac.in/courses/106/103/106103068/
3.	https://www.coursera.org/learn/comparch

4	https://www.edx.org/learn/computer-architecture
5	http://cse10-iitkgp.virtual-labs.ac.in/

Sr No	Suggested List of Assignments
1.	Number conversion from one base to another and addition and subtraction on converted numbers.
2.	Numerical on Booth's Algorithm and on Restoring and Non restoring algorithm. IEEE 754 conversion.
3.	Numerical on Cache memory mapping. Cache coherency and resolution methods.
4.	Different techniques for designing control unit of computer.
5.	Different data transfer techniques and bus arbitration.
6.	Pipeline and pipeline hazards.
7.	Flynn's classification scheme.
8.	Memory interleaving and associative memory.

Assessment:

Term Work: Term Work shall consist of at least 10 to 12 practical's based on the above list. Also, Term work Journal must include at least 2 assignments.

Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)

Practical& Oral Exam: An Oral & Practical exam will be held based on the theory and practical syllabus.

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2164111	**Computational Theory**	**3**	**-**	**-**	**3**	**-**	**-**	**3**

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		Test 1	Test 2	T1 +T2					
2164111	**Computational Theory**	**20**	**20**	**40**	**60**	**2**	**--**	**--**	**100**

Course Objectives:	1) To acquire conceptual knowledge of grammar and languages. 2) To understand the relation between Regular Language and Finite Automata. 3) To understand the language hierarchy, CFG and CFL. 4) To design a PDA equivalent to a given context-free grammar/language. 5) To learn the principles of computation by designing a Turing Machine 6) To infer the knowledge of undecidable and NP class problems.
Course Outcomes:	Upon completion of the course, the learners will be able to: 1) Use TCS theory to design regular expressions that represent regular languages. 2) Design, analyze, and optimize Finite Automata for language recognition. 3) Design Regular and Context Free Grammars and learn to simplify the CFG. 4) Design PDA for a given context-free grammar or language and enumerate its applications. 5) Design Turing machines as generators, deciders, and acceptors for various computational tasks. 6) Understand and utilize problem classification techniques for problem analysis.

Detailed Contents:

Sr. No.	Name of Module	Detailed Content	Hrs	CO Mapping
0	Prerequisite	Basic Mathematical Fundamentals: Sets, Logic, Relations, Functions, Discrete Structures.		
I	**Basics Concepts and Regular Languages**	Importance of TCS, Alphabets, Strings, Languages	**1**	**CO1**
		Regular operations, Regular Expression, Arden's theorem, RE Applications, Regular Language, Closure properties. Decision properties of RLs, Pumping lemma for RLs.	**5**	**CO1**
		Self-learning Topics: RE in text search and replace, Application of Regular Languages in Compiler Design, Text Processing, and Natural Language Processing (NLP).		
II	**Finite Automata**	Finite Automata (FA) & Finite State machine (FSM).	**1**	**CO2**
		Deterministic Finite Automata (DFA) and Nondeterministic Finite Automata (NFA): Definitions, transition diagrams and Language recognizers, Equivalence between NFA with and ϵ- transitions, NFA to DFA Conversion, Minimization of DFA, FSM with	**6**	**CO2**

		output: Moore and Mealy machines, Applications and limitations of FA.		
		Self-learning Topics: State Elimination Method for converting FA to RE, Minimization of DFA using Equivalence Theorem, Conversion of Moore to Mealy & Mealy to Moore machine.		
III	**Regular and Context Free Grammars**	Grammars and Chomsky Hierarchy	**1**	**CO3**
		Regular Grammar (RG), Equivalence of Left and Right linear grammar, Equivalence of RG and FA.	**2**	**CO3**
		Context Free Grammars (CFG) Definition, Sentential forms, Leftmost and Rightmost derivations, Parse tree, Ambiguity, **Simplification of CFG:** Eliminating unit productions, useless production, useless symbols, and Є-productions, **Normal Forms:** Chomsky Normal Form (CNF) and Greibach Normal Form (GNF), Context Free language (CFL) - Application: Parser, Markup languages; Pumping lemma, Closure properties.	**6**	**CO3**
		Self-learning Topics: Left Recursion and Its Elimination, Applications of CFGs in XML Parsing, and Natural Language Processing (NLP).		
IV	**Pushdown Automata (PDA)**	Definition, Language of PDA, PDA as generator, decider and acceptor of CFG, Deterministic PDA , Non-Deterministic PDA, Equivalence of PDA and CFG, Application of PDA.	**5**	**CO4**
		Self-learning Topics: Parsing & PDA: Top-Down Parsing, Bottom-up Parsing, Closure properties and Deterministic PDA.		
V	**Turing Machine (TM)**	Definition, Design of TM as generator, decider and acceptor, Variants of TM: Multitrack, Multitape, Universal TM, Applications, Power and Limitations of TMs.	**7**	**CO5**
		Self-learning Topics: Algorithms using Turing Machine, The Model of Linear Bounded Automata		
VI	**Decidability and Computability**	Decidability and Undecidability, Recursive and Recursively Enumerable Language, Halting Problem, Rice's Theorem, Post Correspondence Problem.	**5**	**CO6**
		Self-learning Topics: NP Completeness of the SAT Problem, A Restricted Satisfiability Problem		

Text Books:	1) John E. Hopcroft, Rajeev Motwani, Jeffery D. Ullman, Introduction to Automata Theory Language and Computation, 3rd Edition, Pearson Education, 2008. 2) Michael Sipser, Theory of Computation, 3rd Edition, Cengage learning. 2013. 3) Vivek Kulkarni, Theory of Computation, Illustrated Edition, Oxford University Press, (12 April 2013) India.
References Books :	1) J. C. Martin, Introduction to Languages and the Theory of Computation, 4th Edition, Tata McGraw Hill Publication, 2013. 2) Kavi Mahesh, Theory of Computation: A Problem-Solving Approach, Kindle Edition, Wiley-India, 2011.

Online References:	1) https://www.jflap.org/ 2) https://nptel.ac.in/courses/106104028 3) https://nptel.ac.in/courses/106104148
Internal Assessment (IA) :	Internal Assessment will consist of **Two** Compulsory IA Tests and shall be conducted for Total 40 Marks including 02 Tests of 20 marks each. Approximately 40% to 50% of syllabus content must be covered in First IA Test and remaining 40% to 50% of syllabus content must be covered in Second IA Test.
Question paper format:	• Question Paper will comprise of a total of **six questions each carrying 20 marks Q.1** will be **compulsory** and should **cover maximum contents of the syllabus** • **Remaining questions** will be **mixed in nature** (part (a) and part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module randomly selected from all the modules) • A total of **Three questions** needs to be answered

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2164112	**Database Management System**	**3**	**2**	**-**	**3**	**1**	**-**	**4**

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		Test 1	Test 2	Avg.					
2164112	**Database Management System**	20	20	40	60	2	--	--	100

Rationale:

Today's data-driven world, Database Management Systems (DBMS) are essential for efficiently storing, managing, and analyzing data. This course equips students with foundational concepts and practical skills to design and implement robust data-driven solutions across diverse domains.

Sr. No.	Course Objectives:
1	To Understand the fundamentals of a database systems
2	Develop entity relationship data model /EER and its mapping to relational model
3	Learn relational algebra and Formulate SQL queries.
4	Apply normalization techniques to normalize the database
5	Understand concept of transaction, concurrency control and recovery techniques
6	Explore and understand recent databases and their application

Sr No	Course Outcomes	BL
CO1	Understand concepts of DBMS and design ER/EER diagram for real world application.	L2, L3
CO2	Apply mapping rules to construct relational model from data model and formulate relational algebra queries.	L3
CO3	Apply SQL queries for database operations.	L3
CO4	Analyze and apply normalization techniques to relational database design.	L3, L4
CO5	Understand transaction, concurrency and recovery techniques to analyze conflicts in multiple transactions.	L2
CO6	Understand recent databases.	L2

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
0	Prerequisite	Basic knowledge of Data structure, Fundamentals of computer system		
I	**Introduction to Database and Data Modeling**	**Introduction:** Definitions and application, Characteristics of databases, DBMS architecture, ACID Properties **The Entity-Relationship (ER) Model**: Entity types: Weak and strong entity sets, Entity sets, Types of Attributes, Keys, Relationship	**08**	**CO1**

		constraints: Cardinality and Participation, Extended Entity-Relationship (EER) Model: Generalization, Specialization and Aggregation **Self-learning Topics:** Design an ER model for any real time case study.		
II	**Relational Model and Relational Algebra**	Introduction to the Relational Model, relational schema and concept of keys. Mapping the ER and EER Model to the Relational Model, Relational Algebra- operators (Selection(σ),Projection(π),Union(∪),Difference (−),CartesianProduct(×),Join(⋈), Intersection (∩),Rename (ρ)), Relational Algebra Queries **Self-learning Topics:** Practice writing queries to perform common database tasks (e.g., selecting data, joining tables	**05**	**CO2**
III	**Structured Query Language (SQL)**	Overview of SQL, Data Definition Commands, Integrity constraints: key constraints, Domain Constraints, Referential integrity, check constraints, Data Manipulation commands, Data Control commands, Transaction Control Commands. aggregate function-group by, having, order by, joins, Nested and complex queries, Views in SQL, Set and string operations, Triggers, Introduction to Pl/SQL Block Structure **Self-learning Topics:** LeetCode (SQL practice problems), HackerRank (SQL challenges)	**10**	**CO3**
IV	**Database Normalization**	Pitfalls in relational database designs, Concept of normalization, Function Dependencies, FD closure, First Normal Form, 2NF, 3NF, BCNF, 4NF. **Self-learning Topics:** Consider any real time application and normalization upto 3NF/BCNF	**5**	**CO4**
V	**Transaction Management and Concurrency Control**	Transaction concept, Transaction states, Concurrent Executions, Serializability-Conflict and View, Concurrency Control: Lock-based, Timestamp-based protocols, Recovery System: log-based recovery, Introduction to Deadlock handling **Self-learning Topics:** SQL challenges related to transactions and concurrency	**7**	**CO5**
VI	**Introduction to Modern databases**	Recent trends in the industry, Introduction of Cloud Database, Introduction of Distributed Database, Introduction to NOSQL Database and Object-Oriented Databases **Self-learning Topics:** Learn about emerging database technologies. Explore different NoSQL types. Learn how object-oriented programming concepts like objects and inheritance are applied to database management systems.	**4**	**CO6**

Text Books:

1. Elmasri and Navathe, Fundamentals of Database Systems, 7thEdition, Pearson Education
2. Korth, Slberchatz, Sudarshan, Database System Concepts, 7thEdition, McGraw Hill
3. Raghu Ramkrishnan and Johannes Gehrke, Database Management Systems, TMH
4. RajkumarBuyya, Christian Vecchiola, S ThamaraiSelvi, “Mastering Cloud Computing”, Tata McGraw-Hill Education

Reference Books:

1. Peter Rob and Carlos Coronel, Database Systems Design, Implementation and Management, Thomson Learning, 5thEdition.
2. Dr. P.S. Deshpande, SQL and PL/SQL for Oracle 10g, Black Book, Dreamtech Press.
3. G. K. Gupta, Database Management Systems, McGraw Hill, 2012

Online References:

Sr. No.	Website Name
1.	NPTEL Lecture Series: Database Management system By Prof. Partha Pratim Das, Prof. Samiran Chattopadhyay | IIT Kharagpur
2.	https://www.classcentral.com/course/swayam-database-management-system-9914
3.	https://www.mooc-list.com/tags/dbms
4.	W3Schools: SQL tutorials

Internal Assessment (IA) for 40 marks:

IA will consist of Two Compulsory Internal Assessment Tests. Approximately 40% to 50% of syllabus content must be covered in First IA Test and remaining 40% to 50% of syllabus content must be covered in Second IA Test.

End Semester Internal Examination for 40 marks:

Question paper format:

- Question Paper will comprise of a total of **six questions each carrying 20 marksQ.1** will be **compulsory** and should **cover maximum contents of the syllabus**
- **Remaining questions** will be **mixed in nature** (part (a) and part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module randomly selected from all the modules)
- A total of **Three questions** needs to be answered.

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical /Oral	Tutorial	Total
2164113	**Operating System**	03	--	--	03	--	--	03

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Pract. /Oral	Total
		Internal assessment			End Sem. Exam			
		Test1	Test 2	Total				
2164113	**Operating System**	20	20	40	60	--	--	100

Course Objectives:

Sr. No.	Course Objectives
The course aims:	
1	To understand the basic concepts of Operating System, its functions and services.
2	To introduce the concept of a process and its management like transition, scheduling, etc.
3	To understand basic concepts related to Inter-process Communication (IPC) like mutual exclusion, deadlock, etc. and role of an Operating System in IPC.
4	To understand the concepts and implementation of memory management policies and virtual memory.
5	To understand functions of Operating System for storage management and device management.
6	To study the need and fundamentals of special-purpose operating system with the advent of new emerging technologies.

Course Outcomes:

Sr. No.	Course Outcomes
1	Define the basic concepts of Operating System, its operations and services.
2	Explain the process management policies and describe the scheduling of processes by the Operating System.
3	Apply synchronization primitives to address process coordination and demonstrate the occurrence of deadlock conditions.
4	Analyze memory allocation and management functions of Operating System.
5	Evaluate the effectiveness of the services provided by the Operating System for File and I/O Management, considering their impact on overall system performance.
6	Design a framework to compare and optimize the functions of various special-purpose Operating Systems for specific application requirements.

DETAILED SYLLABUS:

Sr. No.	Module	Detailed Content	Hours	CO Mapping
I	Fundamentals of Operating System	**Introduction of Operating Systems:** System Boot, Objectives of Operating System Functions of Operating System, Operating System Structure and Operations, Operating System Services, Multiprogramming, Multitasking, Multithreading, Types of Operating System, Types of System Calls.	**03**	CO1

		Self-learning Topics: Study of various Operating System Architecture like IoT, Android.		
II	Process Management	Basic Concepts of Process: Process State Transition Model, Operations, Process Control Block, Context Switching; Introduction to Threads, Types of Threads, Thread Models, Basic Concepts of Scheduling, Types of Schedulers, Type of scheduling algorithms: Preemptive and non preemptive (FCFS, SJF, Priority and Round Robin) **Self-learning Topics:** Real-time Scheduling algorithms and applications.	**06**	CO2
III	Process Synchronization and Deadlock Management	Basic Concepts of Inter-process Communication and Synchronization, Race Condition, Critical Section Problem ,Peterson's Solution, Process Synchronization, Hardware and Semaphores, Producer Consumer Problem. Deadlocks Management: System Model, Deadlock Characterization, Deadlock Prevention, Deadlock Avoidance: Bankers algorithm, Deadlock Detection and Recovery. **Self-learning Topics:** Study a real time case study for Deadlock detection and recovery. Overview of security mechanism in OS.	**10**	CO3
IV	Memory Management	Basic Concepts of Memory Management: Swapping, Memory Allocation strategy, Paging, Structure of Page Table, Segmentation, TLB. Basic Concepts of Virtual Memory, Demand Paging, Copy-on Write, Page Replacement Algorithms, Thrashing. **Self-learning Topics:** Memory Management of IoT, Android Operating System.	**09**	CO4
V	File and IO Management	File Management: Basic Concepts of File System, File Access Methods, Directory Structure, File-System implementation, Allocation Methods, Overview of Mass-Storage Structure, I/O devices, Organization of the I/O Function, Disk Organization, I/O Management and Disk Scheduling: FCFS, SSTF, SCAN, C-SCAN, LOOK, C-LOOK. **Self-learning Topics:** File System for Linux and Windows, Features of I/O facility for different OS.	**07**	CO5
VI	Special-purpose Operating Systems	Open-source and Proprietary Operating System, Fundamentals of Distributed Operating System, Network Operating System, Architecture and functions: Cloud Operating System, Real-Time Operating System, Mobile Operating System. **Self-learning Topics:** Case Study on any one Special-purpose Operating Systems.	**04**	CO6

Text Books:

1. A. Silberschatz, P. Galvin, G. Gagne, Operating System Concepts, 10^{th} ed., Wiley, 2018.
2. W. Stallings, Operating Systems: Internal and Design Principles, 9^{th} ed., Pearson, 2018.
3. A. Tanenbaum, Modern Operating Systems, Pearson, 4^{th} ed., 2015.

Reference Books:

1. Achyut Godbole and Atul Kahate, Operating Systems, McGraw Hill Education, 3rdEdition
2. N. Chauhan, Principles of Operating Systems, 1^{st} ed., Oxford University Press, 2014.
3. A. Tanenbaum and A. Woodhull, Operating System Design and Implementation, 3^{rd} ed., Pearson.
4. R. Arpaci-Dusseau and A. Arpaci-Dusseau, Operating Systems: Three Easy Pieces, CreateSpace Independent Publishing Platform, 1^{st} ed., 2018.

Online References:

1. https://www.nptel.ac.in
2. https://swayam.gov.in
3. https://www.coursera.org/

Assessment:

Internal Assessment (IA) for 40 marks:

IA will consist of Two Compulsory Internal Assessment Tests. Approximately 40% to 50% of syllabus content must be covered in First IA Test and remaining 40% to 50% of syllabus content must be covered in Second IA Test.

End Semester Internal Examination for 40 marks:

Question paper format:

- Question Paper will comprise of a total of **six questions each carrying 20 marksQ.1** will be **compulsory** and should **cover maximum contents of the syllabus**
- **Remaining questions** will be **mixed in nature** (part (a) and part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module randomly selected from all the modules)
- A total of **Three questions** needs to be answered.

<table>
<tr><th rowspan="2">Course Code</th><th rowspan="2">Course Name</th><th colspan="3">Teaching Scheme (Contact Hours)</th><th colspan="4">Credits Assigned</th></tr>
<tr><th>Theory</th><th>Pract.</th><th>Tut.</th><th>Theory</th><th>Pract.</th><th>Tut.</th><th>Total</th></tr>
<tr><td>2164114</td><td>Database Management System Lab</td><td>2</td><td>-</td><td>-</td><td>2</td><td>-</td><td>-</td><td>2</td></tr>
</table>

<table>
<tr><th rowspan="4">Course Code</th><th rowspan="4">Course Name</th><th colspan="7">Examination Scheme</th></tr>
<tr><td colspan="4">Theory Marks</td><td rowspan="3">Term Work</td><td rowspan="3">Practical/ Oral</td><td rowspan="3">Total</td></tr>
<tr><td colspan="3">Internal assessment</td><td rowspan="2">End Sem. Exam</td></tr>
<tr><td>Test 1</td><td>Test 2</td><td>Avg. of 2 Tests</td></tr>
<tr><td>2164114</td><td>Database Management System Lab</td><td>--</td><td>--</td><td>--</td><td>--</td><td>25</td><td>25</td><td>50</td></tr>
</table>

Lab Objectives:
1. To explore database management system concepts and their application
2. To learn major components of DBMS (DDL, DML, DCL, TCL)
3. Understand the use of Structured Query Language (SQL) and learn SQL syntax.
4. To understand the different database constraints and their usage.
5. Understand the needs of database processing transaction handling
6. Learn techniques for controlling and managing concurrent data access

Lab Outcomes: On successful completion of course, learner will be able to:
1. Design ER and EER diagram for the real-life problem with software tool.
2. Create and update database and tables with different DDL and DML statements.
3. Apply /Add integrity constraints and able to provide security to data.
4. Implement and execute Complex queries.
5. Apply triggers and procedures for specific module/task
6. Apply concurrent transactions and implement through practical examples

Prerequisite:
- The below suggested experiments needs to be performed by a group of **2 students. (Mini 10 Experiments)**
- Suggestion: Select any database management system problem statement and try to execute all experiments based on the same topic

Module	Suggested List of experiments	Hours
1	Identify the case study and detail statement of problem. Design an Entity-Relationship (ER) / Extended Entity-Relationship (EER) Model.	02
2	Mapping ER/EER to Relational schema model.	02
3	Create a database using Data Definition Language (DDL) and apply integrity constraints for the specified System	02
4	Apply DML Commands for the specified system	02
5	Perform Simple queries, string manipulation operations and aggregate functions.	02
6	Implement various Join operations.	02

7	Perform Nested and Complex queries	04
8	Perform DCL and TCL commands	02
9	Implement procedure and functions	02
10	Implementation of Views and Triggers.	02
11	Implementation and demonstration of Transaction and Concurrency control techniques using locks.	02
12	Mini project (Design simple GUI and Backend Connectivity)	02

Assessment:

Term Work: Term Work shall consist of at **least 10 to 12 practical** based on the above list.

Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment with Attendance) + 5 Marks (**very basic Mini Proj- as mention in Exp. No 12)** + 5 Marks (Assignment)

Practical& Oral Exam: An Oral & Practical exam will be held based on the above syllabus**.**

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2164115	**Operating System Lab**	2	-	-	2	-	-	2

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Practical/ Oral	Total
		Internal assessment			End Sem. Exam			
		Test 1	Test 2	Avg. of 2 Tests				
2164115	**Operating System Lab**	--	--	--	--	25	25	50

Lab Objectives:

1. To gain practical experience with designing and implementing concepts of operating systems such as system calls, CPU scheduling, process management, memory management, file systems and deadlock handling using C language in Linux environment.
2. To familiarize students with the architecture of Linux OS.
3. To provide necessary skills for developing and debugging programs in Linux environment.
4. To learn programmatically to implement simple operation system mechanisms

Suggested List of Experiments.

Sr No	Suggested List of Experiments	Hrs
01	Explore usage of basic Linux Commands and system calls for file, directory and process management. For eg: (mkdir, chdir, cat, ls, chown, chmod, chgrp, ps etc. system calls: open, read, write, close, getpid, setpid, getuid, getgid, getegid, geteuid. sort, grep, awk, etc.)"	02
02	Write shell scripts to do the following: a. Display OS version, release number, kernel version b. Display top 10 processes in descending order c. Display processes with highest memory usage. d. Display current logged in user and log name. e. Display current shell, home directory, operating system type, current path setting, current working directory.	02
03	Implement any one basic commands of linux like ls, cp, mv and others using kernel APIs.	02
04	Create a child process in Linux using the fork system call. From the child process obtain the process ID of both child and parent by using getpid and getppid system call.	02
05	a. Write a program to demonstrate the concept of non-preemptive scheduling algorithms (any one).	02
06	Write a program to demonstrate the concept of preemptive scheduling algorithms (any one)	02
07	Write a C program to implement solution of Producer consumer problem through Semaphore	02
08	Write a program to demonstrate the concept of deadlock avoidance through Banker's Algorithm	02
09	Write a program to demonstrate the concept of MVT and MFT memory management techniques	02

10	Write a program to demonstrate the concept of dynamic partitioning placement algorithms i.e. Best Fit, First Fit, Worst-Fit etc.	02
11	Write a program to demonstrate the concept of demand paging for simulation of Virtual Memory implementation	02
12	Write a program in C demonstrate the concept of page replacement policies for handling page faults eg: FIFO, LRU etc.	02
13	Write a C program to simulate File allocation strategies typically sequential, indexed and linked files	02
14	Write a C program to simulate file organization of multi-level directory structure.	02
15	Write a program in C to do disk scheduling - FCFS, SCAN, C-SCAN	02
16	Understand the basics of distributed systems through simple file sharing. Set up a network of two or more computers (or virtual machines) on the same network. Configure a shared folder using Samba on Linux (or Windows shared folders) so both systems can access files. Transfer files between the machines and observe the performance of data sharing.	02
17	Get hands-on experience with mobile OS development. Develop a basic app using Android Studio (Java/Kotlin) or Xcode (Swift). Explore Android/iOS permissions by requesting basic access like camera or location. Deploy the app on an emulator or physical device.	02

Note: Any 3 questions from assignment 1 and assignment 2 but should cover all CO's

Sr No	Suggested List of Assignments / Tutorials	Co mapped
	Assignment 1	
01	System Boot Process and OS Initialization: Research and document the system boot process on two different platforms: Windows and Linux.	CO1
02	Exploring Operating System Services : Research and create a detailed report or presentation on the various services provided by an operating system.	CO1
03	Process State Transition Model and Process Control Block (PCB): Explore the structure and role of the Process Control Block (PCB) in modern operating systems. Research how the process state transition model works in various OS architectures (e.g., Unix, Linux, Windows).	CO2
04	Types of Threads and Thread Models: A Comparative Study of Thread Models and Their Applications in Multi-core Systems. Analyze different thread models (User-level, Kernel-level, Hybrid) and their performance in real-world applications.	CO2
05	Inter-process Communication and Synchronization: Explore different inter-process communication (IPC) mechanisms used in operating systems, such as message passing, shared memory, and pipes. Compare their performance, scalability, and use cases in modern OS environments.	CO3
06	Operating System Security: Investigate and prepare a report on common security vulnerabilities in modern operating systems (e.g., buffer overflow, privilege escalation) and propose measures to mitigate these vulnerabilities.	CO3
	Assignment 2	
01	Swapping: Compare and contrast how concept of swapping works in modern OS (e.g., Linux, Windows) versus older systems. Include the performance trade-offs involved in swapping and how it impacts system responsiveness and resource utilization.	CO4
02	Structure of Page Table :Explore the structure of page tables in modern operating systems, and compare different schemes such as hierarchical page tables, inverted page tables, and hashed page tables. Investigate the benefits and limitations of each.	CO4
03	Basic Concepts of File System: Focus on the role of the file system in managing files, directories, and metadata. Compare different types of file systems, such as FAT, NTFS, ext4, and APFS, and explain how each handles file organization, access, and storage.	CO5
04	Disk Organization : Study the physical and logical organization of disks, including tracks, sectors, cylinders, and the role of the disk controller. Explain	CO5

	how the OS maps logical block addresses (LBA) to physical addresses	
05	Open-source vs Proprietary Operating Systems : Compare and contrast open-source operating systems (e.g., Linux, FreeBSD) and proprietary operating systems (e.g., Windows, macOS).	CO6
06	Real-Time Operating System (RTOS): explain the key characteristics of a Real-Time Operating System (RTOS), focusing on aspects like deterministic behavior, task scheduling, and real-time deadlines.	CO6

Assessment:

Term Work: Term Work shall consist of at least 10 to 12 practicals' based on the above list. Also, Term work Journal must include at least 2 assignments.

Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)

Practical& Oral Exam: An Oral & Practical exam will be held based on the above syllabus.

Vertical – 4

Course Code	Course Name	Teaching Scheme (Contact Hours)	Credits Assigned

		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2164411	**Mini Project**	--	4	--	--	2	--	2

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Practical/ Oral	Total
		Internal assessment			End Sem. Exam			
		Test 1	Test 2	Avg. of 2 Tests				
2164411	**Mini Project**	--	--	--	--	50	25	75

Objectives	
1	To acquaint with the process of identifying the needs and converting it into the problem.
2	To familiarize the process of solving the problem in a group.
3	To acquaint with the process of applying basic engineering fundamentals to attempt solutions to the problems.
4	To inculcate the process of self-learning and research.

Outcome: Learner will be able to…	
1	Identify problems based on societal /research needs.
2	Apply Knowledge and skill to solve societal problems in a group.
3	Develop interpersonal skills to work as member of a group or leader.
4	Draw the proper inferences from available results through theoretical/ experimental/simulations.
5	Analyse the impact of solutions in societal and environmental context for sustainable development.
6	Use standard norms of engineering practices
7	Excel in written and oral communication.
8	Demonstrate capabilities of self-learning in a group, which leads to lifelong learning.
9	Demonstrate project management principles during project work.

Guidelines for Mini Project	
1	Students shall form a group of 3 to 4 students, while forming a group shall not be allowed less than three or more than four students, as it is a group activity.
2	Interdisciplinary mini project is also permitted.

3	Students should do survey and identify needs, which shall be converted into problem statement for mini project in consultation with faculty supervisor/head of department/internal committee of faculties.
4	Students shall submit implementation plan in the form of Gantt/PERT/CPM chart, which will cover weekly activity of mini project.
5	A logbook to be prepared by each group, wherein group can record weekly work progress, guide/supervisor can verify and record notes/comments.
6	Faculty supervisor may give inputs to students during mini project activity; however, focus shall be on self-learning.
7	Students in a group shall understand problem effectively, propose multiple solution and select best possible solution in consultation with guide/ supervisor.
8	Students shall convert the best solution into working model using various components of their domain areas and demonstrate.
9	The solution to be validated with proper justification and report to be compiled in standard format of University of Mumbai.
10	With the focus on the self-learning, innovation, addressing societal problems and entrepreneurship quality development within the students through the Mini Project.

Term Work		
The review/ progress monitoring committee shall be constituted by head of departments of each institute. The progress of mini project to be evaluated on continuous basis, minimum two reviews in the semester.		
In continuous assessment focus shall also be on each individual student, assessment based on individual's contribution in group activity, their understanding and response to questions.		
Distribution of Term work marks shall be as below:		**Marks**
1.	Marks awarded by guide/supervisor based on logbook	10
2.	Marks awarded by review committee (Average of Review 1 & Review 2)	10
	Mini-Project Review-1	
	a. Identification of Problem	2
	b. Requirement analysis and Feasibility of the proposed work	2
	c. Literature Review	2
	d. Objectives of the proposed work	2
	e. Methodology of the proposed work	2
	Total Marks	10
	Mini-Project Review-2	
	a. Planning of project work and team structure	2

	b. Design Methodology	2
	c. Conceptual and Technical Demonstration	2
	d. Presentation: Oral delivery, contact with audience, slides, timing	2
	e. Quality of answers	2
	Total Marks	10
3	**Quality of Project report**	**5**

Review / progress monitoring committee may consider following points for the assessment	
1	Students group shall complete project in all aspects including, · Identification of need/problem · Proposed final solution · Procurement of components/systems · Building prototype and testing
2	Two reviews will be conducted for continuous assessment, · First shall be for finalization of problem and proposed solution · Second shall be for implementation and testing of solution.
Assessment criteria of Mini Project.	
Mini Project shall be assessed based on following criteria;	
1	Quality of survey/ need identification
2	Clarity of Problem definition based on need.
3	Innovativeness in solutions
4	Feasibility of proposed problem solutions and selection of best solution
5	Cost effectiveness
6	Societal impact
7	Innovativeness
8	Cost effectiveness and Societal impact
9	Full functioning of working model as per stated requirements
10	Effective use of skill sets
11	Effective use of standard engineering norms
12	Contribution of an individual's as member or leader
13	Clarity in written and oral communication

Guidelines for Assessment of Mini Project Practical/Oral Examination:		
1	Report should be prepared as per the mentioned guidelines (Preferred in LaTex).	5
2	Mini-Project shall be assessed through a presentation and demonstration of the working model by the student project group to a panel of Internal and External Examiners preferably from industry or research organizations having experience of more than five years approved by the head of Institution. Project presentation and demonstration to be evaluated w.r.t following parameters.	
	a. Identification of Problem and Literature review	4
	b. Problem Statement and Objective of the Proposed work	4
	c. Design Methodology	4
	d. Implementation	4
	Total marks	**16**
3	Students shall be motivated to publish a paper based on the work in Conferences/ students competitions e.t.c	4
	Total marks	**25**

References to get Project ideas:

- https://www.guvi.in/blog/top-mini-project-ideas-for-college-students/
- https://www.geeksforgeeks.org/project-idea-college-network/?ref=ml_lbp
- https://www.simplilearn.com/tutorials/artificial-intelligence-tutorial/ai-project-ideas
- https://roadmap.sh/backend/project-ideas
- https://webflow.com/blog/website-ideas
- https://gist.github.com/MWins/41c6fec2122dd47fdfaca31924647499
- https://www.projectpro.io/article/artificial-intelligence-project-ideas/461
- https://github.com/The-Cool-Coders/Project-Ideas-And-Resources
- https://nevonprojects.com/project-ideas/software-project-ideas/
- https://roadmap.sh/projects

Vertical – 5

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2993511	**Entrepreneurship Development**	--	**2*+2**	-	-	**2*+2**	-	2

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Practical/ Oral	Total
		Internal assessment			End Sem. Exam			
		IAT-I	IAT-II	IAT-I + IAT-II				
2993511	**Entrepreneurship Development**	--	--	--	--	50	--	50

Note: * Two hours of practical class to be conducted for full class as demo/discussion/theory.

Lab Objectives:

1. To introduce students to entrepreneurship concepts and startup development.
2. To develop business idea generation, validation, and business model preparation.
3. To provide hands-on experience in market research, financial planning, and business pitching.
4. To enhance problem-solving and decision-making skills in entrepreneurial ventures.
5. To familiarize students with government schemes and support systems for entrepreneurs.
6. To develop communication and presentation skills required for business pitching.

Lab Outcomes:

Upon successful completion of this course, students will be able to:

1. Understand the fundamental concepts of entrepreneurship and business models.
2. Conduct market research and develop business plans.
3. Utilize financial planning and cost analysis for startups.
4. Apply entrepreneurial skills to identify and solve business challenges.
5. Develop prototypes using open-source software for business operations.
6. Pitch business ideas effectively with structured presentations.

DETAILED SYLLABUS

Sr. No.	Module	Detailed Content	Hours	LO Mapping
0	**Prerequisite**	Fundamentals of communication and leadership skills.	01	--
I	**Introduction to Entrepreneurship**	Definition, Characteristics, and Types of Entrepreneurs. Entrepreneurial Motivation and Traits. Start-up Ecosystem in India. Challenges in Entrepreneurship	02	LO1
II	**Business Idea Generation & Validation**	Ideation Techniques: Design Thinking, Brainstorming, Mind Mapping. Business Model Canvas (BMC). Market Research & Customer	04	LO2

		Validation. Minimum Viable Product (MVP) Concept.		
III	**Business Planning & Strategy**	Writing a Business Plan. SWOT Analysis and Competitive Analysis. Financial Planning and Budgeting. Risk Assessment and Management	04	LO3
IV	**Funding and Legal Framework**	Sources of Funding: Bootstrapping, Angel Investors, Venture Capital Government Schemes & Start-up India Initiatives. Business Registration & Legal Formalities. Intellectual Property Rights (IPR) & Patents	05	LO4
V	**Marketing & Digital Presence**	Branding and Digital Marketing. Social Media Marketing & SEO. Customer Relationship Management (CRM). E-commerce & Online Business Models	05	LO5
VI	**Business Pitching & Prototype Development**	Pitch Deck Preparation & Presentation Techniques. Prototyping with Open-source Tools. Elevator Pitch & Investor Pitch. Case Studies of Successful Start-ups	05	LO6

Text Books:

1. "Entrepreneurship Development and Small Business Enterprises" – Poornima M. Charantimath, Pearson, 3rd Edition, 2021.
2. "Innovation and Entrepreneurship" – Peter F. Drucker, Harper Business, Reprint Edition, 2019.
3. "Startup and Entrepreneurship: A Practical Guide" – Rajeev Roy, Oxford University Press, 2022.
4. "Essentials of Entrepreneurship and Small Business Management" – Norman Scarborough, Pearson, 9th Edition, 2021.
5. "The Lean Startup" – Eric Ries, Crown Publishing, 2018.

References:

1. "Disciplined Entrepreneurship: 24 Steps to a Successful Startup" – Bill Aulet, MIT Press, 2017.
2. "Zero to One: Notes on Startups, or How to Build the Future" – Peter Thiel, 2014.
3. "The $100 Startup" – Chris Guillebeau, Crown Business, 2019.
4. "Business Model Generation" – Alexander Osterwalder & Yves Pigneur, Wiley, 2020.
5. "Blue Ocean Strategy" – W. Chan Kim & Renée Mauborgne, Harvard Business Review Press, 2019.

Online Resources:

Website Name
1. **Startup India Portal** – https://www.startupindia.gov.in 2. **MIT OpenCourseWare – Entrepreneurship** – https://ocw.mit.edu/courses/sloan-school-of-management/ 3. **Coursera – Entrepreneurship Specialization** – https://www.coursera.org/specializations/entrepreneurship 4. **Harvard Business Review – Entrepreneurship Articles** – https://hbr.org/topic/entrepreneurship 5. **Udemy – Startup & Business Courses** – https://www.udemy.com/courses/business/entrepreneurship/

List of Experiments.

Sr No	List of Experiments	Hrs
01	Business Idea Generation using Mind Mapping.	02
02	Conducting Market Research & Customer Validation.	02
03	Preparing a Business Model Canvas for a Startup Idea.	02
04	Developing a Financial Plan & Break-even Analysis.	02
05	Creating a Website using WordPress/Wix.	02
06	Social Media Marketing Campaign using Open-source Tools.	02
07	Digital Prototyping using Figma/Inkscape.	02
08	Business Pitch Deck Preparation & Presentation.	02
09	Exploring Government Schemes for Startups.	02
10	Legal Compliance & IPR Basics (Case Study).	02

Sr No	List of Assignments / Tutorials	Hrs
01	a. Write a report on any successful entrepreneur and their startup journey. b. Conduct SWOT analysis for a real-life startup.	02
02	Develop a business idea and create a one-page business plan.	02
03	Conduct market research using surveys & present findings.	02
04	Design a simple logo and branding strategy for a startup.	02
05	Create a financial model and cost estimation for a startup.	02
06	Make a case study report on startup failure analysis.	02

List of Open-Source Software

1. Canva – Designing pitch decks, social media posts, and branding materials.
2. Trello / Asana – Project management for startups.
3. GIMP / Inkscape – Graphic design and logo creation.
4. WordPress / Wix – Website development for startups.
5. OpenCart / PrestaShop – E-commerce website setup.
6. Figma – UI/UX design and prototyping.
7. LibreOffice Calc – Financial planning and budgeting.
8. Google Suite (Docs, Sheets, Slides) – Documentation and presentations.
9. Python (Pandas, Flask, Django) – Data analytics and web application development.
10. MailChimp – Email marketing and customer engagement.

Assessment :

Term Work: Term Work shall consist of at least 08 to 10 practicals' based on the above list. Also, Term work Journal must include at least 6 assignments.

Term Work Marks: 50 Marks (Total marks) = 20 Marks (Experiment) + 15 Marks (Assignments) + 5 Marks (Attendance)+ 10 Marks (Report)

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2993512	Environmental Science for Engineers	--	2*+2	-	--	2*+2	-	2

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-I	IAT-II	IAT-I+IAT-II					
2993512	Environmental Science for Engineers	--	--	--	--	--	50	--	50

Note: * Two hours of practical class to be conducted for full class as demo/discussion/theory.

Rationale:

Most of the engineering branches are offspring of applied sciences, and their practices have a significant impact on the environment. Understanding environmental studies is essential for engineers to develop sustainable solutions, minimize ecological footprints, and promote responsible resource management. This course equips students with the knowledge of ecosystems, biodiversity, pollution control, and environmental laws, enabling them to integrate sustainability into engineering practices.

Lab Objectives:

1. To understand the scope, importance, and role of environmental studies in public awareness and health.
2. To study different natural resources, their issues, and sustainable conservation.
3. To understand ecosystem types, structures, and functions.
4. To explore biodiversity, its importance, threats, and conservation.
5. To learn about pollution types, causes, effects, and control measures.
6. To understand environmental challenges, sustainability, and ethics.

Lab Outcomes:

1. Explain the significance of environmental studies and the role of IT in environment and health.
2. Describe resource types, associated problems, and conservation methods.
3. Classify ecosystems and explain their role in ecological balance
4. Analyze biodiversity levels and conservation strategies, especially in India.
5. Explain pollution impacts and suggest preventive measures.
6. Discuss environmental issues and propose sustainable solutions.

DETAILED SYLLABUS:

Unit Name	Topic Name	Topic Description	Hours	LO Mapping
I	**The Multidisciplinary Nature of Environmental Studies**	Definition, scope and importance. Need for public awareness, Role of information technology in environment and human health. Human population and the environment: Population growth, variation among nations. Population Explosion- family welfare program. Environment and human health Women and child welfare	**03**	LO1

II	**Natural Resources**	Renewable and non-renewable resources. Natural resources & associated problems: a) Forest resources: b) Water resources: Natural resources & associated problems c) Mineral resources: d) Food resources: e) Energy resources: Role of an individual in conservation of natural resources: f) Equitable use of resources for sustainable lifestyles.	**04**	LO2
III	**Ecosystems**	Concepts of an ecosystem. Introduction, types, characteristic features, structure and function of the following ecosystem: a. Forest ecosystem b. Grassland ecosystem c. Desert ecosystem d. Aquatic ecosystem (ponds, streams, lakes, rivers, oceans, estuaries). Case study on various ecosystems in India.	**05**	LO3
IV	**Biodiversity and its Conservation**	Introduction, Definition, genetic species and ecosystem diversity. Bio-geographical classification of India Value of biodiversity, Consumptive use, productive use, social, ethical, aesthetic and option values, Bio-diversity at global, national, local levels India as a mega diversity nation, Case study on Bio diversity in India.	**05**	LO4
V	**Environmental Pollution Definition**	Causes, effects and control measures of: a) Air pollution b) Water pollution b) Soil pollution. Solid waste management: Causes, effect and control measures of urban and industrial wastes. Role of an individual in prevention of pollution, Case study on Pollution Disaster management: floods, earthquake, cyclone and landslides. Carbon Credits for pollution prevention.	**05**	LO5
VI	**Social Issues and Environment**	From unsustainable to sustainable development Urban problems related to energy, Water conservation, rain water harvesting, watershed management. Environmental ethics: issues and possible solution. Climate change, global warming, acid rain, ozone layer depletion, nuclear accidents and holocaust. Case studies. Consumerism and waste products. Environment protection act. Public awareness Case study on Environmental Ethics	**04**	LO6

Textbooks

1. Environmental Science: Towards a Sustainable Future, G. Tyler Miller and Scott Spoolman, 13th Edition, Cengage Learning 2021
2. Environmental Management: Text and Cases, Bala Krishnamoorthy, 3rd Edition, PHI Learning, Publication Year: 2016
3. Green IT: Concepts, Technologies, and Best Practices, Markus Allemann, Springer 2008

4. Sustainable IT: Slimming Down and Greening Up Your IT Infrastructure, David F. Linthicum, IBM Press 2009
5. Environmental Modelling: Finding Solutions to Environmental Problems, David L. Murray, Cambridge University Press 2016
6. Remote Sensing and Image Interpretation, Thomas M. Lillesand, Ralph W. Kiefer, and Jonathan W. Chipman, 9th Edition, John Wiley & Sons 2020
7. Business Ethics: Concepts and Cases, Manuel Velasquez, Pearson 2012

Reference Books

1. Environmental Law and Policy in India, Shyam Divan and Armin Rosencranz, 2nd Edition, Oxford University Press 2018
2. The International Handbook of Environmental Laws, David Freestone and Jonathon L. Rubin, Edward Elgar Publishing 2000
3. E-Waste Management: Challenges and Opportunities in Developing Countries, Ruediger Kuehr and Ram K. Jain, Springer 2008
4. The E-Waste Handbook: Managing Electronic Waste, Klaus Hieronymi, Ruediger Kuehr, and Ram K. Jain, Earthscan 2009
5. Environmental Ethics: An Introduction, J. Baird Callicott, University of Georgia Press1999

Online References:

Sr. No.	Website Name
1.	Centre for Science and Environment (CSE), Website: cseindia.org
2.	Ministry of Environment, Forest and Climate Change (MoEFCC), Government of India
3.	CSIR-National Environmental Engineering Research Institute (NEERI)

List of Experiments.

Sr No	List of Experiments	Hrs
01	Study of Environmental Components and Ecosystems.	2
02	Visit and Report on Solid Waste Management Plant.	2
03	Study of Renewable Energy Sources (Solar, Wind, Biogas).	2
04	Analysis of Air and Water Quality Parameters.	2
05	Study of Local Biodiversity and Conservation Methods.	2
06	Awareness Activity on Environmental Issues.	2
07	Rainwater Harvesting System Design	2
08	Case Study on Environmental Pollution & Control Measures.	2
09	Report on Climate Change Impact and Adaptation.	2
10	Study of Environmental Laws and Acts.	2
11	Study of Disaster Management Techniques.	2
12	Report on Role of IT in Environmental Protection.	2

Sr No	List of Assignments / Tutorials	Hrs
01	Prepare a report on Renewable and Non-Renewable Resources.	2
02	Write a case study on Ecosystem Types in India	2
03	Write a report on Biodiversity in India.	2
04	Prepare a report on Pollution Types and Control Measures.	2
05	Prepare a report on Environmental Ethics and Sustainability.	2
06	Prepare a case study report on Global Warming and Climate Change.	2
07	Report on Role of an Individual in Environmental Protection.	2
08	Write a report on Disaster Management Techniques.	2

09	Prepare a report on Environmental Laws and Acts in India.	2
10	Case Study on E-waste Management and Recycling Techniques.	2

Assessment :

Term Work: Term Work shall consist of at least 10 to 12 practical's based on the above list. Also, Term work Journal must include at least 8 to 10 assignments.

Term Work Marks: 50 Marks (Total marks) = 20 Marks (Experiment) + 15 Marks (Assignments) + 5 Marks (Attendance)+ 10 Marks (Report)

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2994511	Business Model Development	--	2*+2	-	--	2*+2	-	2

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-I	IAT-II	IAT-I+IAT-II					
2994511	Business Model Development	--	--	--	--	--	50	--	50

Note: * Two hours of practical class to be conducted for full class as demo/discussion/theory.

Lab Objectives:

1. To introduce a learner to entrepreneurship and its role in economic development.
2. To familiarize a learner with the start-up ecosystem and government initiatives in India.
3. To explain the process of starting a business.
4. To familiarize a learner with the building blocks of a business.
5. To teach a learner to plan their own business with the help of Business Model Canvas.
6. To teach a learner to have financial plan for a business model.

Lab Outcomes:

The learner will be able to:

1. Discuss the role of entrepreneurship in the economic development of a nation and describe the process of starting a business.
2. Describe start-up ecosystems in Indian and global context.
3. Identify different types of business models.
4. Identify customer segments, channels and customer relationship components for a particular business.
5. Identify key activities, key partners and key resources for a particular business.
6. Develop a financial plan for a business with the help of cost structure and revenue model.

DETAILED SYLLABUS:

Sr. No.	Module	Detailed Content	Hours	LO Mapping
0	**Prerequisite**	Basic Design Thinking principles	**01**	**--**
I	**Introduction to Entrepreneurship**	**Introduction to Entrepreneurship**: Definition, the role of entrepreneurship in the economic development, the entrepreneurial process, Women entrepreneurs, Corporate entrepreneurship, Entrepreneurial mindset **Self-learning Topics:** Case studies on Henry Ford: https://www.thehenryford.org/docs/default-source/default-document-library/default-document-library/henryfordandinnovation.pdf?sfvrsn=0 The Tatas: How a Family Built a Business and a Nation by Girish Kuber, April 2019, Harper Business	**04**	**L1, L2**
II	**Entrepreneurship Development**	**Entrepreneurship Development**: Types of business ownerships: Proprietorship, Public and Private Companies, Co-operative businesses, Micro, Small and Medium Enterprises (MSME): Definition and role of MSMEs in economic development	**05**	**L2, L3, L4**

III	**Start-up financing**	**Start-up financing:** Cost and revenue models, Sources of start-up fundings: Angel investors, Venture capitalists, Crowd funding, Government schemes for start-up funding **Self-learning Topics:** Successful business pitching	**04**	**L2, L3, L4, L5**
IV	**Intellectual Property Rights (IPR)**	**Intellectual Property Rights (IPR):** Types of IPR: Patents, trademarks and copyrights, Patent search and analysis, Strategies for IPR protection, Ethics in technology and innovation	**04**	**L2, L3, L4**
V	**Business Model Development**	**Business Model Development:** Types of business models, Value proposition, Customer segments, Customer relationships, Channels, Key partners, Key activities, Key resources, Prototyping and MVP **Self-learning Topics:** The Art of the Start 2.0: The Time-Tested, Battle-Hardened Guide for Anyone Starting Anything by Guy Kawasaki	**04**	**L3, L4, L5, L6**
VI	**Digital Business Management**	**Digital Business Management:** Digital Business models (Subscription, Freemium *etc*), Digital marketing: Search Engine Optimization (SEO), Search Engine Marketing (SEM), Social media and influencer marketing, Disruption and innovation in digital business **Self-learning Topics:** Case study: Airbnb https://www.prismetric.com/airbnb-business-m	**04**	**L2, L3**

Textbooks:

1. Entrepreneurship: David A. Kirby, McGraw Hill, 2002
2. Harvard Business Review: Entrepreneurs Handbook, HBR Press, 2018
3. Business Model Generation; Alexander Ostlewalder and Yves Pigneur, Strategyzer, 2010
4. E- Business & E– Commerce Management: Strategy, Implementation, Practice – Dave Chaffey, Pearson Education

Reference books:

1. Entrepreneurship: New venture creation by David Holt, Prentice Hall of India Pvt. Ltd.
2. E- Business & E– Commerce Management: Strategy, Implementation, Practice – Dave Chaffey, Pearson Education

Online Resources:

Sr. No.	Website Name
1.	Entrepreneurship by Prof. C Bhaktavatsala Rao https://onlinecourses.nptel.ac.in/noc20_mg35/preview
2.	Innovation, Business Models and Entrepreneurship by Prof. Rajat Agrawal, Prof. Vinay Sharma https://onlinecourses.nptel.ac.in/noc21_mg63/preview
3.	Sarasvathy's principles for effectuation https://innovationenglish.sites.ku.dk/model/sarasvathy-effectuation/

List of Experiments.

The lab activities are to be conducted in a group. One group can be formed with 4-5 students. A group has to develop a Business Model Canvas and a digital prototype (Web App/ mobile app). Weekly activities are to be conducted as follows:

Sr No	Lab activities	Hrs
01	Problem identification (Pain points, Market survey)	2
02	Design a digital solution for the problem (Ideation techniques)	2
03	Preparing a business model canvas: Value proposition, Key partners, Key resources, Key activities	2
04	Preparing a business model canvas: Customer segment, Customer relationships and channels	2
05	Preparing a business model canvas: Cost and Revenue structure	2
06	Prototype development: Low fidelity	2
07	Prototype development: Customer feedback	2
08	Prototype development: High fidelity	2
09	Presentation of high-fidelity prototype	2

Sr No	List of Assignments / Tutorials	Hrs
01	Presentation on case study of a failed business model	2
02	Presentation on case study of a woman entrepreneur	2

Assessment:

Term Work: Term Work shall consist of 09 lab activities based on the above list. Also, Term work journal must include any 2 assignments from the above list.

Term Work Marks: 50 Marks (Total marks) = 25 Marks (Experiment) + 10 Marks (Assignments) + 5 Marks (Attendance)+10 Marks (Report).

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2994512	**Design Thinking**	**--**	**2*+2**	**-**	**--**	**2*+2**	**-**	**2**

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-I	IAT-II	IAT-I+IAT-II					
2994512	**Design Thinking**	**--**	**--**	**--**	**--**	**--**	**50**	**--**	**50**

Note: * Two hours of practical class to be conducted for full class as demo/discussion/theory.

Lab Objectives:

1. To introduce a learner to the principles of Design Thinking.
2. To familiarize a learner with the process (stages) of Design Thinking.
3. To introduce various design thinking tools.
4. Study of the techniques for generation of solutions for a problem.
5. To expose a learner to various case studies of Design Thinking.
6. Create and test a prototype.

Lab Outcomes:

Students will be able to …

1. Compare traditional approach to problem solving with the Design Thinking approach and discuss the principles of Design Thinking
2. Define a user persona using empathy techniques
3. Frame a problem statement using various Design Thinking tools
4. Use ideation techniques to generate a pool of solutions for a problem
5. Create prototypes using different techniques
6. Test the prototypes and gather feedback for refining the prototype

DETAILED SYLLABUS:

Sr. No.	Module	Detailed Content	Hours	LO Mapping
0	Prerequisite	No perquisites	-	-
I	**Introduction to Design Thinking**	**Introduction to Design Thinking:** Definition, Comparison of Design Thinking and traditional problem-solving approach, Need for Design Thinking approach, Key tenets of Design Thinking, 5 stages of Design Thinking (Empathize, Define, Ideate, Prototype, Test) **Self-learning Topics:** Design thinking case studies from various domains https://www.design-thinking-association.org/explore-design-thinking-topics/external-links/design-thinking-case-study-index	**05**	**L1, L2**
II	**Empathy**	**Empathy:** Foundation of empathy, Purpose of empathy, Observation for empathy, User observation technique, Creation of empathy map **Self-learning Topics:** Creation of empathy maps https://www.interactiondesign.org/literature/topics/empathy-mapping	**05**	**L2, L3**

III	**Define**	**Define:** Significance of defining a problem, Rules of prioritizing problem solving, Conditions for robust problem framing, Problem statement and POV **Self-learning Topics:** Creating a Persona – A step-by-step guide with tips and examples https://uxpressia.com/blog/how-to-create-persona-guide-examples	**05**	**L2, L3**
IV	**Ideate**	**Ideate:** What is ideation? Need for ideation, Ideation techniques, Guidelines for ideation: Multi-disciplinary approach, Imitating with grace, Breaking patterns, Challenging assumptions, Looking across value chain, Looking beyond recommendation, Techniques for ideation: Brainstorming, Mind mapping **Self-learning Topics:** How To Run an Effective Ideation Workshop: A Step-By-Step Guide https://uxplanet.org/how-to-run-an-effective-ideation-workshop-a-step-by-step-guide-d520e41b1b96	**05**	**L3**
V	**Prototype**	**Prototype:** Low and high-fidelity prototypes, Paper prototype, Story board prototype, Scenario prototype	**03**	**L6**
VI	**Test**	**Test:** 5 guidelines of conducting test, The end goals of test: Desirability, Feasibility and Viability, Usability testing	**03**	**L4, L5**

Textbooks:

1. Design Your Thinking: The Mindsets, Toolsets, and Skill Sets for Creative Problem-solving, Pavan Soni, Penguin Random House India Private Limited
2. Design Thinking: Methodology Book, Emrah Yayichi, 2016
3. Handbook of Design Thinking: Christian Mueller-Roterberg, 2018

Reference books:

1. Design Thinking for Strategic Innovation: What They Can't Teach You at Business or Design School, Idris Mootee, Wiley, 2013
2. Change by Design, Tim Brown, Harper Business, 2009

Online Resources:

Sr. No.	Website Name
1.	Design Thinking and Innovation by Ravi Poovaiah https://onlinecourses.swayam2.ac.in/aic23_ge17/preview
2.	Introduction to Design Thinking by Dr. Rajeshwari Patil, Dr. Manisha Shukla, Dr. Deepali Raheja, Dr. Mansi Kapoor https://onlinecourses.swayam2.ac.in/imb24_mg37/preview
3.	Usability Testing https://www.interaction-design.org/literature/topics/usability-testing

List of Experiments:

The experiments are to be performed in groups. A practical batch may be divided into groups of 4-5 students.

Sr No	List of Experiments	Hrs
01	Customer Journey Mapping: Visualize the steps users take to interact with a product or service. Map out the customer journey from discovering a product to making a purchase and using the product. Identify pain points and opportunities for improvement.	2
02	Stakeholder mapping: Identify all relevant stakeholders in a project. Create a stakeholder map, categorizing stakeholders based on their influence and interest. Include management of relationships with key stakeholders.	2
03	"How Might We" Problem Framing: Transform user insights into actionable problem statements. After empathizing with users, turn challenges into "How Might We" statements that define the problem without prescribing a solution.	2
04	Brainstorming Session: Generate a pool of ideas in a creative, non-judgmental environment. Using ideation techniques like mind mapping and brainwriting, students brainstorm as many solutions as possible to their "How Might We" problem statements.	2
05	Affinity Diagramming: Organize group ideas to find patterns and insights. After brainstorming, students will categorize their ideas into themes by placing sticky notes on a wall and moving them into groups based on similarities.	2
06	Rapid Prototyping: Create quick, low-fidelity versions of solutions. Use materials like paper, cardboard, and markers to build a prototype of their solution within 30 minutes. The focus is on speed and functionality, not aesthetics.	2
07	Wireframing: Create a visual guide for digital interfaces for mobile app / web app for the problems identified in earlier lab sessions. Students will sketch wireframes of the user interface for their product or service. Use tools like Balsamiq or paper and pen for low-fidelity wireframes.	2
08	Role-Playing: Walk through a prototype from the user's perspective. Students act as both users and designers, role-playing scenarios where they interact with their prototype (Developed in earlier lab sessions). Gather feedback from participants on how to improve the experience.	2
09	Usability Testing: Evaluation of the effectiveness and user-friendliness of a prototype (developed in earlier lab sessions). Students will have peers or target users test their prototypes, observe how they interact with it, and collect feedback on any issues or improvements needed.	2
10	Feedback Loop and Iteration: Refine solutions based on user feedback. After usability testing, students will refine their prototypes. Document changes made based on feedback and discuss how continuous iteration improves the design.	2

Sr No	List of Assignments (Any two)	Hrs
01	Create an empathy map for a target user group. Break them into four sections: *Says, Thinks, Feels, and Does*. Interview users or research their experiences to fill in the map.	3
02	Based on research, students will create user personas including demographic details, motivations, pain points, and goals. Each group will present their persona to the class.	3
03	Consider 3 examples of real-life products which have good design and bad	3

	design. Write down reasons why do you think they are good or bad designs. May take user survey to support your work.	
04	Study any open-source design thinking tool and write a brief report about it.	3

Assessment:

Term Work: Term Work shall consist of 08 to 10 lab activities based on the above list. Also, Term work journal must include any 2 to 4 assignments from the above list.
Term Work Marks: 50 Marks (Total marks) = 25 Marks (Experiment) + 10 Marks (Assignments) + 5 Marks (Attendance)+ 10 Marks (Report).

Vertical – 6

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2163611	**Full Stack Java Programming**	**-**	**2*+2**	**-**	**-**	**2**	**-**	**2**

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Practical/ Oral	Total
		Internal assessment			End Sem. Exam			
		Test1	Test 2	Avg. of 2 Tests				
2163611	**Full Stack Java Programming**	--	--	--	--	50	25	75

Lab Objectives: This subject seeks to give students an understanding of full stack development in Java. The main aim of this course is to:

1. Familiarize with Basic OOP concepts in Java,
2. Understand the concepts of inheritance and exceptions in java,
3. Design and implement programs involving Client and Server Side Programming,
4. Describe and utilize the functioning of DOM and Java script,
5. Study different design patterns in web programming and understand the working of react framework,
6. To describe the Spring Framework and implement the related case studies.

Lab Outcomes: At the end of the course, the students should be able to:

1. Understand and apply the fundamentals of Java Programming and Object-Oriented Programming,
2. Analyze and Illustrate Inheritance and Exception Handling Mechanisms,
3. Elaborate and design applications using Client and Server Side Programming,
4. Understand the concepts in JavaScript for interactive Web Development,
5. Implement the real-world application development in web programming using React,
6. Design and Develop Enterprise-Level Applications Using the Spring Framework.

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
0	Prerequisite	Basic Programming constructs in C & Python.		
I	**Introduction to OOP in Java**	**OOP concepts:** Objects, class, Encapsulation, Abstraction, Inheritance, Polymorphism, message passing. Branching and looping. Class, object, data members, member functions Constructors, types, static members and functions Method overloading Input and output functions in Java, Buffered reader class, scanner class, Packages in java, types, user defined packages. **Self-learning Topics:** Array and Vectors in Java	4	LO 1
II	**Inheritance & Exception Handling**	**Inheritance:** Types of inheritance, Method overriding, super, abstract class and abstract method, final, Multiple inheritances using interface, extends keyword. **Exception Handling:** try, catch, finally, throw and throws, Multiple try and catch blocks, user	3	LO 2

		defined exception. **Self-learning Topics:** Multithreading in Java		
III	**Client and Server Side Programming**	**Java Database Connectivity (JDBC):** JDBC architecture and drivers Connecting to databases (MySQL, Oracle, etc.) Executing SQL queries using Java Statements. **Client Side Scripting: HTML:** Elements, Attributes, Head, Body, Hyperlink, Formatting, Images, Tables, List, Frames, Forms. **CSS3:**Syntax, Inclusion, Color, Background, Fonts, Tables, lists,CSS3 selectors. **Server side programming in Java:** Introduction of Servlet, Servlet lifecycle, Servlet Request, Servlet Response, Servlet Context, HTTP Sessions, Handling forms and user inputs, Session management. Introduction to Java Server Pages, JSP architecture and page directives, Components of a JSP, Scripting elements and Standard actions, Method Definitions, JSTL. **Self-learning Topics:** Database Connectivity in Servlets and Implement JSP with JDBC to fetch data from a database	5	LO 3
IV	**Fundamentals of Java Script**	**Java Script:** Introduction to JavaScript: Conditionals Statements, Loops, Functions, Arrays, Objects, Control Flow, Math Function, Browser Object Model, Document Object Model. **DOM Manipulation:** Introduction to the DOM, Defining the DOM, Defining DOM, Dom Tree, Language-Specific DOMs, Accessing relative nodes, Checking the node type, Dealing with attributes, Creating and manipulating nodes, DOM HTML Features, Attributes as properties, Table methods, DOM Traversal, NodeIterator, TreeWalker, Selector methods, Detecting DOM Conformance, DOM style methods, Custom tooltips, Collapsible sections, Accessing style sheets Events, Fetch & Callbacks: Event Flow, Event Handlers/Listeners, The Event Object, Types of Events, Cross-Browser Events, HTTP Responses, Working with JSON data. **Self-learning Topics:** AJAX	5	LO 4
V	**Web Programming using React**	**Design Pattern:** Understanding MVC architecture Implementing MVC with servlets and JSP Developing a complete web application Solving company's use cases. **React Framework:** Introduction to React JS, Components and Elements of React, Rendering Components, React State and Props, Events, Hooks, Routing Conditional Rendering, Lists and Keys, Forms, create a single page application using React. **Self-learning Topics:** Flux and Redux	5	LO 5
VI	**Applications of Spring Framework**	**Spring Framework:** Introduction to Microservices, Basics Dependency injection and inversion of control (IoC), Spring annotations, Database integration and Aspect-oriented programming (AOP) with spring, creating spring boot applications, Building RESTful APIs with spring boot.	4	LO 6

		Self-learning Topics: Real-time Applications on Spring Framework		

Text Books:

1. Herbert Schildt, "Java The Complete Reference" Ninth Edition, Oracle Press
2. Christopher Schmitt and Kyle Simpson, "HTML5 Cookbook",O'Really Press
3.Nicholas C. Zakas,"Professional JavaScript™ for Web Developers", Wiley Publishing
4. Amuthan G., "Spring MVC, Beginners Guide" Pakt Publication
5. Chris Minnick,"BEGINNING ReactJS Foundations Building User Interfaces with ReactJS", Wrox publication
6.Iuliana Cosmina, Rob Harrop, "Pro Spring 5 An In-Depth Guide to the Spring Framework and Its Tools",Fifth Edition,APress

Reference Books:

1. Laura Lemay, Charles L. Perkins","Teach Yourself JAVA in 21 Days",Sams.net Publishing
2. Eureka, Ribbon, Zuul and Cucumber Moises Macero,"Learn Microservices with Spring Boot A Practical Approach to RESTful Services using RabbitMQ", APress
3.Alex Banks & Eve Porcello," React FUNCTIONAL WEB DEVELOPMENT WITH REACT AND REDUX", O'Really Press

Online Resources:

Sr. No.	Website Name
1.	https://www.javatpoint.com/html5-tutorial
2.	https://www.w3schools.com/js/
3.	https://www.tutorialspoint.com/spring_boot/index.htm
4.	https://www.w3schools.com/REACT/DEFAULT.ASP

Suggested list of Experiments		
Sr No	**Title of Experiments**	**Hrs**
01	Programs on classes and objects	2
02	Programs on method and constructor overloading.	2
03	Programs on various types of inheritance and Exception handling	2
04	Program on Implementing Generic and HTTP servlet.	2
05	Design a login webpage in JSP that makes validation through Database using JDBC and call the servlet for various operations	2
06	Program on Implicit and Explicit objects in JSP	2
07	Program to create a website using HTML CSS and JavaScript	2
08	Program using Java Script to validate the email address entered by the user (check the presence of “@” & “.” character. If this character is missing, the script should display an alert box reporting the error and ask the user to re-enter it again).	2
09	Program based on Document Object Model to change the background color of the web page automatically after every 5 seconds.	2
10	Program for making use of React Hooks that displays four buttons namely, “Red”, “Blue”, “Green”, “Yellow”. On clicking any of these buttons, the code displays the message that you have selected that particular color.	2
11	Creating a Single Page website using the concepts in React like Hooks, Router, Props and States.	2
12	Program to create a Monolithic Application using SpringBoot	2
13	Program for Building RESTful APIs with spring boot	2

Sr No	Suggested List of Assignments / Tutorials	Hrs
1.	Theory Assignment based on Introduction to OOP in Java, Inheritance, Exception Handling and Client/Server Side Programming (Chapter 1 to 3)	4
2.	Theory Assignment based on Fundamentals of Java Script, Web Programming using React and Applications of Spring Framework (Chapter 4 to 6)	4

Assessment:

Term Work: Term Work shall consist of at least 10 to 12 practicals' based on the above list. Also, Term work Journal must include at least 2 assignments. Mini Project based on the content of the syllabus (Group of 2-3 students), The final certification and acceptance of term work ensures that satisfactory performance of laboratory work and minimum passing marks in term work.

Term Work Marks: Total 50-Marks (Experiments: 15-marks, Attendance: 05-marks, Assignments: 05-marks, Mini Project: 20-marks, MCQ as a part of lab assignments: 5-marks)

Practical& Oral Exam: An Oral & Practical exam will be held based on the above syllabus.

Letter Grades and Grade Points:

Semester GPA/ Programme CGPA Semester/ Programme	**% of Marks**	**Alpha-Sign/ Letter Grade Result**	**Grading Point**
9.00 - 10.00	90.0 – 100	O (Outstanding)	10
8.00 - < 9.00	80.0 - < 90.0	A+ (Excellent)	9
7.00 - < 8.00	70.0 - < 80.0	A (Very Good)	8
6.00 - < 7.00	60.0 - < 70.0	B+ (Good)	7
5.50 - < 6.00	55.0 - < 60.0	B (Above Average)	6
5.00 - < 5.50	50.0 - < 55.0	C (Average)	5
4.00 - < 5.00	40.0 - < 50.0	P (Pass)	4
Below 4.00	Below 40.0	F (Fail)	0
Ab (Absent)	**----**	Ab (Absent)	0

Sd/-

Dr. Subhash K. Shinde
BoS Chairman, Computer Engineering
Faculty of Science & Technology

Sd/-

Dr. Deven Shah
Associate Dean
Faculty of Science & Technology

Sd/-

Prof. Shivram S. Garje
Dean
Faculty of Science & Technology

AC –

Item No. –

As Per NEP 2020

University of Mumbai

Syllabus for Major Vertical – 1, 4 & 5		
Name of the Programme – B.E. Computer Science and Engineering (Artificial Intelligence & Machine Learning)		
Faculty of Engineering		
Board of Studies in Computer Engineering		
U.G. Third Year Programme	Exit Degree	B.Sc. Tech Engineering- Computer Science and Engineering (Artificial Intelligence & Machine Learning)
Semester		V & VI
From the Academic Year		2026-27

University of Mumbai

(As per NEP 2020)

Sr. No.	Heading	Particulars
1	Title of program O:____________________	**B.E. Computer Science and Engineering (Artificial Intelligence & Machine Learning)**
2	Exit Degree	**B.Sc. Tech Engineering- Computer Science and Engineering (Artificial Intelligence & Machine Learning)**
3	Scheme of Examination R: ____________	NEP 40% Internal 60% External, Semester End Examination Individual Passing in Internal and External Examination
4	Standards of Passing R: ______________	**40%**
5	Credit Structure R: ______________	**Attached herewith**
6	Semesters	**Sem. V & VI**
7	Program Academic Level	**5.5**
8	Pattern	**Semester**
9	Status	**New**
10	To be implemented from Academic Year	**2026-27**

Sd/-
Dr. Subhash K. Shinde
BoS-Chairman-Computer Engineering
Faculty of Technology

Sd/-
Dr. Deven Shah
Associate Dean
Faculty of Science & Technology

Sd/-
Prof. Shivram S. Garje
Dean
Faculty of Science & Technology

Preamble

To meet the challenge of ensuring excellence and NEP 2020 policy in engineering education, the issue of quality needs to be addressed, debated, and taken forward systematically. Accreditation is the principal means of quality assurance in higher education. The major emphasis of the accreditation process is to measure the outcomes of the program that is being accredited. In line with this Faculty of Science and Technology (in particular Engineering) of the University of Mumbai has taken the lead in incorporating the philosophy of NEP 2020 education in the process of curriculum development.

The third-year engineering course consists of a core and elective training program to impart scientific and logical thinking training to learners in general, with a choice of course selection from the program core course, program elective course, multidisciplinary minor, IKS, PCE and vocational skill-enhanced course. Simultaneously, the objectives of NEP 2020 demand nurturing the core program, elective program, communication skills and skills required for the Computer Engineering Branch of engineering in the learner. Keeping this in view, a pool of courses is offered in Core Courses covering fundamentals required to understand core and modern engineering practices and emerging trends. Elective courses are offered to the learners to understand the concepts of emerging technology. This will help with the holistic development of learners through interdisciplinary skills development and mobility among other departments as real-time solutions. Considering the shift in pedagogy and the convenience of a stress-free learning process, a choice-based subject pool is offered in the coursework under the heads of Computer Engineering in Engineering for open electives and multidisciplinary minor courses in sem V and VI. Essentially, to give a glimpse of trends in the industry under vocational and enhanced skill practices, the pool is offered to nurture and develop creative skills in contemporary industrial practices. Criteria met in the structure is the opportunity for learners to choose the course of their interest in all disciplines.

Program Core Course and Program Elective Courses Cover Computer Engineering core and elective courses. Also, MDM is where a pool of subjects is given for selection. In addition to this learner will have IKS and PCE courses. Considering the present scenario, diverse choices need to be made available to fulfil the expectation of a learner to aspire for a career in the field of current trends of Technology and interdisciplinary research. The IKS course will be program-specific, which can be achieved in Undergraduate training by giving an objective viewpoint to the learning technology, from ascent time and how it was learned and the process and transitioning a learner from a rote learner to a creative professional. For this purpose, PCE is introduced in the First Semester to orient a communication skills to become skilled professionals for placement and higher education entrance exams. Considering the NEP-2020 structure of award of Certificate after U.G. First Year, U. G. Diploma after the Second Year and B.Sc. Tech Engineering after the Third Year at multiple exit-point pools of Vocational skills, is arranged to give exposure to the current Industry practices.

The faculty resolved that course objectives and course outcomes are to be clearly defined for every core and elective course so that all faculty members in affiliated higher education institutes understand the depth and approach of the course to be taught to the learners, which will enhance the learners' learning process. NEP 2020 grading system enables a much-needed shift in focus from teacher-centric to continuous-based learner-centric education since the workload estimated is based on the investment of time in learning and not in teaching. It also focuses on the holistic development of the learners through mobility and continuous evaluation, which will enhance the quality of education. Credit assignment for courses is based on a 15-week teaching-learning process for NEP 2020; however, the content of courses is to be taught in 12-13 weeks, and the remaining 2-3 weeks are to be utilized for revision, self learning, tutorial, guest lectures, coverage of content beyond the syllabus, etc.

There was a concern that in the present system, the third-year syllabus must not be heavily loaded for the learner and it is of utmost importance that the learner entering into the third year of an engineering course should feel at ease by lowering the burden of syllabus and credits. This is necessary for a learner to get accustomed to the new environment of a college and to create a bond between the teacher and the learner. The present curriculum will be implemented for the Third Year of Engineering from the academic year 2026-27. Subsequently, this system will be carried forward for Final Year Engineering in the academic years 2027-28, respectively.

Sd/-
Dr. Subhash K. Shinde
BoS-Chairman-Computer Engineering
Faculty of Technology

Sd/-
Dr. Deven Shah
Associate Dean
Faculty of Science & Technology

Sd/-
Prof. Shivram S. Garje
Dean
Faculty of Science & Technology

B.Sc. Tech Engineering- Computer Science and Engineering (Artificial Intelligence & Machine Learning)

Level	Semester	Major		Minor	OE	VSC, SEC (VSEC)	AEC, VEC, IKS	OJT, FP, CEP, CC,RP	Cum. Cr. / Sem.	Degree/ Cum. Cr.
	R: _______________C									
		Mandatory	Electives							
5.5	V	PCC501:3 PCC502:3 PCC503:3 PCL501: 1 PCL502:1 PCL503:1	PEC501:3 PEL501:1	MDC:3 MDL:1	OE:2	--	IKS: 2	--	24	B.Sc. Tech Engineering 48
	R:_______________D									
	VI	PCC601:3 PCC602:3 PCL601:1 PCL602:1	PEC601:3 PEC602:3 PEL601:1 PEL602:1	MDC: 3 MDL:1	--	VSEC:2	AEC: 2	--	24	
	Cum Cr.	20	12	8	2	2	2+2	--	48	

Exit option: Award of UG Diploma in Major and MDM with 90 credits and additional 4 credits core **one** theory subject with 3 credits and **one** lab with 1 credit from one third year from where they want to take Exit degree. Along with theory and practical course student must compulsory do internship for **one month or 160 hours** which internship is equal to 4 credits.

[Abbreviation - OE – Open Electives, VSC – Vocation Skill Course, SEC – Skill Enhancement Course, (VSEC), AEC – Ability Enhancement Course, VEC – Value Education Course, IKS – Indian Knowledge System, OJT – on Job Training, FP – Field Project, CEP – Continuing Education Program, CC – Co-Curricular, RP – Research Project]

T.E.
Scheme

Program Structure for Third Year of Computer Science and Engineering (Artificial Intelligence & Machine Learning)
UNIVERSITY OF MUMBAI (With Effect from 2026-2027)

SEMESTER V

Course Code	Course Description	Teaching Scheme (Contact Hours)			Credit Assigned			
		Theory	Practical	Tutorial	Theory	Tutorial	Practical	Total Credits
2165111	Statistics for Machine Learning and Data Science	3	--	--	3	--	--	3
2165112	Artificial Intelligence and Soft Computing	3	–	--	3	--	--	3
2165113	Agile Software Development and DevOps	3	--	--	3	--	--	3
216511#	Program Electives -I	3	--	--	3	--	--	3
2165211	Multi-disciplinary Minor	3	--	--	3	--	--	3
2165117	Statistics for Machine Learning and Data Science Lab	--	2	--	--	--	1	1
2165118	Artificial Intelligence and Soft Computing Lab	--	2	--	--	--	1	1
2165119	Agile Software Development and DevOps Lab	--	2	--	--	--	1	1
216512#	Program Electives -I Lab	--	2	--	--	--	1	1
2165212	Multi-disciplinary Minor Lab	--	2@	--	--	--	1	1
2165311	To be taken from the bucket provided by the University from other Faculty	2#	--	---	2	--	--	2
2165511	Indian Knowledge System (Syllabus common to all Branch).	--	2*+2	--	--	--	2	2
Total		**17**	**14**	**00**	**17**	**00**	**07**	**24**

* Two hours of practical class to be conducted for full class as demo/discussion.

Theory / Tutorial 1 credit for 1 hour and Practical 1 credit for 2 hours

Institute shall offer a course for Open Elective from Science/Commerce/Management stream bucket provided by the University of Mumbai.

@ Institute shall offer a course for MDM from other Engineering Boards Group.

Program Electives:	
Course Code	Course Description
2175114	Web and Mobile Application Development
2175115	User Experience Design with VR
2175116	Computer Network
Lab Code	Lab Description
2175120	Web and Mobile Application Development Lab
2175121	User Experience Design with VR Lab
2175122	Computer Network Lab

Program Structure for Third Year of Computer Science and Engineering (Artificial Intelligence & Machine Learning)
UNIVERSITY OF MUMBAI (With Effect from 2026-2027)

SEMESTER V

Course Code	Course Description	Examination Scheme							
		Internal Assessment Test (IAT)			End Sem. Exam Marks	End Sem. Exam Duration (Hrs)	Term Work (Tw)	Oral & Pract.	Total
		IAT-I	IAT-II	Total (IAT-I) + IAT-II)					
2165111	Statistics for Machine Learning and Data Science	20	20	40	60	2	--	--	100
2165112	Artificial Intelligence and Soft Computing	20	20	40	60	2	--	--	100
2165113	Agile Software Development and DevOps	20	20	40	60	2	--	--	100
216511#	Program Electives -I	20	20	40	60	2	--	--	100
2165211	Multi-disciplinary Minor	20	20	40	60	2	--	--	100
2165117	Statistics for Machine Learning and Data Science Lab	--	--	--	--	--	25	25	50
2165118	Artificial Intelligence and Soft Computing Lab	--	--	--	--	--	25	25	50
2165119	Agile Software Development and DevOps Lab	--	--	--	--	--	25	25	50
216512#	Program Electives -I Lab	--	--	--	--	--	25	-	25
2165212	Multi-disciplinary Minor Lab	--	--	--	--	--	25	25	50
2165311	To be taken from the bucket provided by the University from other Faculty	--	--	--	--	--	50	--	50
2165511	Indian Knowledge System (Syllabus common to all Branch).	--	--	--	--	--	50	--	50
Total		**100**	**100**	**200**	**300**	**10**	**225**	**100**	**825**

Program Electives:	
Course Code	Course Description
2165114	Web and Mobile Application Development
2165115	User Experience Design with VR
2165116	Computer Network
Lab Code	Lab Description
2165120	Web and Mobile Application Development Lab
2165121	User Experience Design with VR Lab
2165122	Computer Network Lab

Program Structure for Third Year of Computer Science and Engineering (Artificial Intelligence & Machine Learning)
UNIVERSITY OF MUMBAI (With Effect from 2026-2027)

SEMESTER VI

Course Code	Course Description	Teaching Scheme (Contact Hours)			Credit Assigned			
		Theory	Practical	Tutorial	Theory	Tutorial	Practical	Total Credits
2166111	Machine Learning and Swarm Intelligence	3	--	--	3	--	--	3
2166112	Data Analytics and Visualization	3	--	--	3	--	--	3
216611#	Program Electives -II	3	--	--	3	--	--	3
216611#	Program Electives -III	3	--	--	3	--	--	3
2166211	Multi-disciplinary Minor	3	--	--	3	--	--	3
2166411	Mini Project-II	--	4	--	--	--	2	2
2166511	Professional Skill	--	2*+2	--	--	--	2	2
2166119	Machine Learning and Swarm Intelligence Lab	--	2	--	--	--	1	1
2166120	Data Analytics and Visualization Lab	--	2	--	--	--	1	1
216612#	Program Electives -II Lab	--	2	--	--	--	1	1
216612#	Program Electives -III Lab	--	2	--	--	--	1	1
2166212	Multi-disciplinary Minor Lab	--	2@	--	--	--	1	1
Total		15	18	00	15	00	09	24

* Two hours of practical class to be conducted for full class as demo/discussion.
Theory / Tutorial 1 credit for 1 hour and Practical 1 credit for 2 hours
@ Institute shall offer a course for MDM from other Engineering Boards.

	Course Code	Course Description	Lab Code	Lab Description
Program Electives -II	2166113	Internet of Things	2166121	Internet of Things Lab
	2166114	Cyber Security	2166122	Cyber Security Lab
	2166115	Nature-Inspired Optimization Algorithms	2166123	Nature-Inspired Optimization Algorithms Lab
Program Electives -III	2166116	Edge Computing	2166124	Edge Computing Lab
	2166117	Digital Imaging Techniques and Analysis	2166125	Digital Imaging Techniques and Analysis Lab
	2166118	Blockchain Technology	2166126	Blockchain Technology Lab

Program Structure for Third Year of Computer Science and Engineering (Artificial Intelligence & Machine Learning)
UNIVERSITY OF MUMBAI (With Effect from 2026-2027)

SEMESTER VI

Course Code	Course Description	Examination Scheme							
		Internal Assessment Test (IAT)			End Sem. Exam Marks	End Sem. Exam Duration (Hrs)	Term Work (Tw)	Oral & Pract.	Total
		IAT-I	IAT-II	Total (IAT-I) + IAT-II)					
2166111	Machine Learning and Swarm Intelligence	20	20	40	60	2	--	--	100
2166112	Data Analytics and Visualization	20	20	40	60	2	--	--	100
216611#	Program Electives -II	20	20	40	60	2	--	--	100
216611#	Program Electives -III	20	20	40	60	2	--	--	100
2166211	Multi-disciplinary Minor	20	20	40	60	2	--	--	100
2166411	Mini Project-II	--	--	--	--	--	50	--	50
2166511	Professional Skill	--	--	--	--	--	25	--	25
2166119	Machine Learning and Swarm Intelligence Lab	--	--	--	--	--	25	25	50
2166120	Data Analytics and Visualization Lab	--	--	--	--	--	25	25	50
216612#	Program Electives -II Lab	--	--	--	--	--	25	25	50
216612#	Program Electives -III Lab	--	--	--	--	--	25	25	50
2166212	Multi-disciplinary Minor Lab	--	--	--	--	--	25	25	50
Total		**100**	**100**	**200**	**300**	**10**	**200**	**125**	**825**

	Course Code	Course Description	Lab Code	Lab Description
Program Electives -II	2166113	Internet of Things	2166121	Internet of Things Lab
	2166114	Cyber Security	2166122	Cyber Security Lab
	2166115	Nature-Inspired Optimization Algorithms	2166123	Nature-Inspired Optimization Algorithms Lab
Program Electives -III	2166116	Edge Computing	2166124	Edge Computing Lab
	2166117	Digital Imaging Techniques and Analysis	2166125	Digital Imaging Techniques and Analysis Lab
	2166118	Blockchain Technology	2166126	Blockchain Technology Lab

Vertical – 1
Major

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2165111	Statistics for Machine Learning and Data Science	3	-	-	3	-	-	3

Course Code	Course Name	Theory					Term Work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT -I	IAT-II	IAT-I + IAT-II					
2165111	Statistics for Machine Learning and Data Science	20	20	40	60	2	--	--	100

Rationale: In modern data-driven systems, statistical understanding forms the backbone of Machine Learning and Data Science. This course integrates foundational statistical concepts with practical statistical learning methods such as regression, classification, resampling, unsupervised learning, Bayesian reasoning, and ethical AI practices. Students will learn to perform sound data analysis, build interpretable ML models, evaluate them rigorously, and understand the implications of data-driven decisions.

Course Objectives: By the end of the course, the learner will be able to:

1. Understand fundamental concepts of probability, descriptive statistics, and sampling techniques for data analysis.
2. Apply estimation and hypothesis testing methods, including classical and resampling approaches.
3. Build and interpret regression models (linear, nonlinear, and regularized) for predictive analytics.
4. Implement supervised and unsupervised statistical learning algorithms to model real-world data.
5. Evaluate models using cross-validation, MLE, and statistical performance metrics.
6. Analyze ethical implications, interpret model decisions, and apply Bayesian reasoning for explainable ML.

Course Outcomes: After successful completion of the course, students will be able to:

1. Summarize and interpret datasets using descriptive measures, sampling and distributional analysis.
2. Perform statistical inference using estimation, hypothesis testing, and resampling approaches.
3. Build, evaluate, and interpret regression and regularized models.
4. Apply supervised and unsupervised learning algorithms effectively.
5. Evaluate models using appropriate metrics, Bayesian reasoning, and cross-validation.
6. Analyze fairness, model interpretability, and ethical implications in ML workflows.

Prerequisite: Basic Probability, Linear Algebra, Python Programming

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
0	**Prerequisite**	Basic Probability, Linear Algebra, Python Programming	00	
I	**Statistical Foundations, Covariance, and Sampling**	Types of data: categorical, numerical, ordinal. Descriptive statistics: central tendency, dispersion, skewness, kurtosis. Data visualization: histograms, boxplots, scatter plots, pair plots. Probability distributions: Normal, Binomial, Poisson, Exponential, Uniform, Beta, Gamma. Covariance, correlation, confounding, correlation matrices. Distance measures: Euclidean, Manhattan, Cosine. Sampling techniques: simple random, stratified, cluster. Central Limit Theorem (CLT) and Law of Large Numbers. Data preprocessing: handling missing values, outlier treatment, encoding, normalization. **Self-learning Topics:** Explore "An Introduction to Statistical Learning (ISLR2)" YouTube videos and Kaggle datasets for descriptive analytics and sampling distribution exercises. https://onlinecourses.nptel.ac.in/noc24_ma03/preview https://onlinecourses.nptel.ac.in/noc24_ma64/preview?	07	CO1, CO6
II	**Statistical Inference, Estimation & Resampling**	Point estimates, properties of estimators: bias, variance, MSE. Maximum Likelihood Estimation (MLE) and Method of Moments. Confidence intervals (mean, proportion, variance). Hypothesis testing: z-test, t-test, chi-square, ANOVA. Nonparametric tests: Mann–Whitney U, Kruskal–Wallis. Resampling methods: bootstrap CI, permutation tests. Likelihood Ratio Tests (LRT). Multiple testing correction: Bonferroni & FDR. Introduction to Monte Carlo simulation for sampling distribution approximation. **Self-learning Topics:** Explore bootstrapping via Python tutorials. Read "Practical Statistics for Data Scientists", Ch. 5 on inference. https://onlinecourses.nptel.ac.in/noc25_ma55/preview?	06	CO2, CO5
III	**Regression Models, Regularization & Nonlinear Models**	Simple & multiple linear regression. Model assumptions, residual analysis, multicollinearity (VIF), leverage, Cook's distance. Polynomial regression and interaction effects. Nonlinear regression models. Regularization: Ridge, Lasso, **Elastic Net.** Spline regression, LOESS (conceptual). Generalized Linear Models (GLM) – logistic (binary), Poisson (count data). Model evaluation metrics: MAE, MSE, RMSE, R^2, Adjusted R^2. Subset selection: forward, backward	08	CO3, CO6

		(overview). Model calibration and interpretability basics. **Self-learning Topics:** Practice polynomial/spline regression using scikit-learn. Use open datasets to implement regularization. https://onlinecourses.nptel.ac.in/noc24_cs20/preview		
IV	**Classification & Supervised Learning**	Classification metrics: accuracy, precision, recall, F1, ROC–AUC, PR curves. Logistic regression, odds ratio, decision boundary. Naïve Bayes (Frequentist and Bayesian views). k-NN classifier. Decision Trees & pruning. Ensemble methods: Bagging, Random Forest, Gradient Boosting (overview). Support Vector Machines (SVM): linear & kernel trick– (conceptual). Handling imbalanced data: SMOTE, class weights, stratified splits. Confusion matrix analysis & error trade-offs. Model calibration: Platt scaling & isotonic regression (overview). **Self-learning Topics:** Implement logistic and k-NN in Python. Explore metric trade-offs using real datasets.	07	CO4, CO6
V	**Unsupervised Learning & Dimensionality Reduction**	PCA: covariance matrix, Eigen decomposition, explained variance. LDA (supervised dimensionality reduction). Kernel PCA & ICA (conceptual overview). Clustering: K-means, hierarchical, DBSCAN. Cluster evaluation: silhouette score, Davies–Bouldin index. Dimensionality reduction for visualization: t-SNE, UMAP (conceptual). Outlier detection methods: Z-score, IQR, Isolation Forest (intro), Local Outlier Factor (LOF) – conceptual overview. **Self-learning Topics:** Visualize PCA/t-SNE clusters. Compare clustering methods with real-world datasets.	05	CO4, CO6
VI	**Model Evaluation, Bayesian Inference & Ethics**	Model selection: AIC, BIC.Bias–variance trade-off. Cross-validation: k-fold, LOOCV, repeated CV. Calibration curves and Brier score (conceptual). Bayesian fundamentals: prior, likelihood, posterior. MAP vs MLE, conjugate priors (Beta–Bernoulli, Normal–Normal). Naïve Bayes from Bayesian perspective. Interpretability tools (conceptual): SHAP, LIME, PDP, ICE. Fairness in ML: equalized odds, demographic parity. Ethical considerations: model bias, transparency, data leakage, Data Privacy Overview: anonymization, differential privacy. Reproducible ML: dataset versioning, MLflow / Weights & Biases (W&B conceptual). **Self-learning Topics:** Read beginner-friendly blogs on Bayesian stats. Interpretability tools: SHAP, LIME (high-level); permutation importance, PDPs, ICE plots.	06	CO5, CO6

Text Books:

1. Aggarwal, Charu C. *Probability and Statistics for Machine Learning*. Springer, 2024.
2. James, Gareth; Witten, Daniela; Hastie, Trevor; Tibshirani, Robert. *An Introduction to Statistical Learning: with Applications in Python* (2nd ed.). Springer, 2023.
3. Murphy, Kevin P. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.
4. *Practical Statistics for Data Scientists — Peter Bruce et al., 2020*

References:

1. Hastie, Trevor; Tibshirani, Robert; Friedman, Jerome. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2nd ed.). Springer, 2009.
2. DasGupta, Anirban. *Probability for Statistics and Machine Learning: Fundamentals and Advanced Topics*. Springer, 2011.
3. Wasserman, Larry. *All of Statistics: A Concise Course in Statistical Inference*. Springer, 2004.
4. McElreath, Richard. Statistical Rethinking: *A Bayesian Course with Examples in R and Stan*. CRC Press, 2020 (2nd ed.).

5. *Before Machine Learning, Volume 3: Probability & Statistics for AI*. Packt Publishing,

Online References: Students are encouraged to use the following verified online learning platforms for additional learning and self-study.

Sr. No.	Website Name
1.	https://www.statlearning.com/
2.	https://scikit-learn.org/stable/
3.	https://developers.google.com/machine-learning/crash-course
4.	https://distill.pub/2016/misread-tsne/
5.	https://christophm.github.io/interpretable-ml-book/
6	https://onlinecourses.nptel.ac.in/noc25_ma20/preview?
7.	https://onlinecourses.nptel.ac.in/noc25_ma21/preview?

Assessment:

Internal Assessment (IA) for 20 marks:

- IA will consist of Two Compulsory Internal Assessment Tests. Each test will be 20 marks. Approximately 40% to 50% of syllabus content must be covered in First IA Test and remaining 40% to 50% of syllabus content must be covered in Second IA Test.

Question paper format:

- Question Paper will comprise of a total of **six questions each carrying 15 marks. Q.1** will be **compulsory** and should **cover maximum contents of the syllabus.**
- **Remaining questions** will be **mixed in nature** (part (a) and part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module randomly selected from all the modules).
- A total of Four questions need to be answered.

University of Mumbai

Course Code: _______________and Course Name:

Date of Exam: Duration: 2 Hours Max. Marks: 60

NB: (1) Question 1 is compulsory. (2) Solve any three questions out of five.		
Q.1	**Solve any three questions out of five: (05 marks each)**	**15**
a)		05
b)		05
c)		05
d)		05
e)		05
Q.2,3,4,5,6		**15**
a)	Part (a) and Part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module.	07or 08
b)		08 or 07

Note for QP Setters:

1. All COs should be mapped on the basis of the syllabus.
2. Module weightage as per contact hours mentioned in the syllabus.
3. Question 1 is compulsory, and it should include sub-questions covering all modules, with each sub-question carrying five marks.
4. The question paper should contain six questions, each of 15 marks.

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2165112	Artificial Intelligence & Soft Computing	3	-	-	3	-	-	3

Course Code	Course Name	Theory					Term Work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-I	IAT-II	IAT-I + IAT-II					
2165112	Artificial Intelligence & Soft Computing	20	20	40	60	2	--	--	100

Course Objectives: The course aims to:

1. To introduce the fundamental concepts, evolution, and significance of Artificial Intelligence and Soft Computing, highlighting their roles in solving real-world problems.
2. To enable students to understand and apply classical AI search strategies such as uninformed, informed, and local search for goal-driven problem solving.
3. To develop the ability to represent and reason with knowledge using logical and probabilistic frameworks for handling uncertainty in intelligent systems.
4. To familiarize learners with fuzzy set theory, fuzzy rules, and inference mechanisms for approximate reasoning and intelligent decision-making.
5. To impart knowledge of neural network architectures and learning algorithms for modelling cognitive and pattern-recognition processes.
6. To integrate fuzzy logic and neural networks into hybrid intelligent systems and analyze their comparative performance across applications.

Course Outcomes: After successful completion of the course, the learner will be able to:

1. Describe the core concepts, constituents, and applications of Artificial Intelligence and Soft Computing.
2. Apply various uninformed, informed, and local search techniques to solve real-world and optimization problems.
3. Represent and reason with knowledge using propositional and predicate logic, and manage uncertainty using probabilistic reasoning.
4. Develop and analyze fuzzy inference systems for decision-making and control applications.
5. Design, train, and evaluate neural network models using supervised and unsupervised learning algorithms.
6. Integrate fuzzy and neural models into hybrid intelligent systems such as ANFIS and evaluate their performance.

DETAILED SYLLABUS:

Sr. No	Name of Module	Detailed Content	Hours	CO Mapping
0	**Prerequisite**	Mathematical Foundations: Linear Algebra, Programming Skills: Python (preferred), Basic understanding of data structures, Discrete Mathematics	--	--
I	**Introduction to Artificial Intelligence & Soft Computing**	Introduction to Artificial Intelligence: Artificial Intelligence (AI), AI Perspectives, Applications of AI, The present state of AI Introduction to Soft Computing: Soft computing Constituents, Characteristics of Neuro Computing and Soft Computing, Difference between Hard Computing and Soft Computing, Concepts of Learning and Adaptation. **Self-learning Topics:** Recent advances in Generative AI, ChatGPT-like models, ethical challenges in AI.	05	CO1
II	**Solving Problems by Searching**	Definition, State space representation, Problem as a state space search, Problem formulation, Uninformed Search: Depth First Search, Breadth First Search, Informed Search: Greedy Best First Search, A* Search. Local Search: Hill Climbing Search, Simulated Annealing Search, **Self-learning Topics:** Game Playing, Adversarial Search Techniques	07	CO2
III	**Knowledge and Reasoning**	Knowledge Representation Systems, Propositional Logic (PL): Syntax, Semantics, Formal logic-connectives, Predicate Logic: FOPL, Syntax, Semantics, Quantification, Inference rules in FOPL, Forward Chaining, Backward Chaining and Resolution in FOPL. Handling Uncertain Knowledge, Random Variables, Prior and Posterior Probability **Self-learning Topics:** Ontologies and Semantic Web reasoning	07	CO3
IV	**Fuzzy Set Theory & Fuzzy Rules, Reasoning and Inference System**	Fuzzy Set Theory: Fuzzy Sets, Fuzzy relations, Fuzzification and Defuzzification. Features of the membership Functions Fuzzy Rules, Reasoning and Inference System: Fuzzy Rules: Fuzzy If-Then Rules, Fuzzy Reasoning Fuzzy Inference System (FIS): Mamdani FIS, Sugeno FIS, Comparison between, Mamdani and Sugeno FIS **Self-learning Topics:** Adaptive fuzzy systems and membership optimization	07	CO4
V	**Neural Networks**	Basics of Neural Networks: Introduction to Neural Networks, Biological Neural Networks, McCulloch Pitt model Supervised Learning algorithms: Perceptron (Single Layer, Multi-layer), Linear separability, Delta learning rule, Back Propagation algorithm Un-Supervised Learning algorithms: Hebbian Learning, Winner take all, Self Organizing Maps, Learning Vector Quantization. **Self-learning Topics:** CNNs and RNNs — introductory overview and comparison	08	CO5
VI	**Hybrid System**	Introduction to Hybrid Systems, Key applications of hybrid systems (industrial control, robotics, forecasting), Adaptive Neuro-Fuzzy Inference System	05	CO6

		(ANFIS) , Comparative evaluation of Fuzzy, Neural, and Neuro-Fuzzy models. **Self-learning Topics:** Sustainability and governance in hybrid AI systems.		

Text Books:

1. Stuart Russell and Peter Norvig, *Artificial Intelligence – A Modern Approach*, 4th Edition, Pearson Education, 2021.
2. S. Rajaraman and G. Vijayalakshmi Pai, *Neural Networks, Fuzzy Logic, and Genetic Algorithms: Synthesis and Applications,* PHI Learning, 2017.
3. S. N. Sivanandam and S. N. Deepa, *Principles of Soft Computing, 3rd Edition*, Wiley India, 2018.

References:

1. Elaine Rich, Kevin Knight, and Shivashankar B. Nair, *Artificial Intelligence,* 3rd Edition, McGraw-Hill Education, 2017
2. Simon Haykin, *Neural Networks and Learning Machines,* 3rd Edition, Pearson, 2017.
3. Bart Kosko, *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence,* Prentice Hall, 1992.

Online References:

Sr. No.	Website Name
3.	https://onlinecourses.nptel.ac.in/noc25_cs48/preview?
4.	https://nptel.ac.in/courses/106105077
3.	https://stanford-cs221.github.io/spring2024/

Assessment:
Internal Assessment (IA) for 20 marks:

- IA will consist of Two Compulsory Internal Assessment Tests. Each test will be 20 marks. Approximately 40% to 50% of syllabus content must be covered in First IA Test and remaining 40% to 50% of syllabus content must be covered in Second IA Test.

➢ **Question paper format:**

- Question Paper will comprise of a total of **six questions each carrying 15 marks. Q.1** will be **compulsory** and should **cover maximum contents of the syllabus.**
- **Remaining questions** will be **mixed in nature** (part (a) and part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module randomly selected from all the modules).
- A total of Four questions need to be answered.

University of Mumbai

Course Code: ________________and Course Name:

Date of Exam: Duration: 2 Hours Max. Marks: 60

NB: (1) Question 1 is compulsory. (2) Solve any three questions out of five.		
Q.1	**Solve any three questions out of five: (05 marks each)**	**15**
a)		05
b)		05
c)		05
d)		05
e)		05
Q.2,3, 4,5,6		**15**
a)	Part (a) and Part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module.	07or 08
b)		08 or 07

Note for QP Setters:

1. All COs should be mapped on the basis of the syllabus.
2. Module weightage as per contact hours mentioned in the syllabus.
3. Question 1 is compulsory, and it should include sub-questions covering all modules, with each sub-question carrying five marks.
4. The question paper should contain six questions, each of 15 marks.

....

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2165113	Agile Software Development & DevOps	3	-	-	3	-	-	3

Course Code	*Course Name*	*Theory*					Term Work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-I	IAT-II	IAT-I + IAT-II					
2165113	Agile Software Development & DevOps	20	20	40	60	2	--	--	100

Course Objectives:

1. Understand software engineering frameworks, SDLC models, and the need for Agile software development.
2. Recognize different Agile development models, their methods, values, roles, artifacts, and business benefits.
3. Understand Agile requirements elicitation, modeling, prioritization, and Agile design principles.
4. Learn Agile project planning, estimation techniques, forecasting, and risk management.
5. Understand Agile approaches to software quality, testing principles, automation, and continuous testing.
6. Understand DevOps principles, tools, lifecycle, and integration with Agile for business agility.

Course Outcomes: After learning this course, students will be able to

1. Describe SDLC process models, differentiate traditional vs. Agile models, and summarize Agile principles and stakeholder roles.
2. Compare Agile development models (Scrum, XP, Kanban, FDD, Lean, Crystal, ASD), and apply Agile values and tools to practical scenarios.
3. Develop and prioritize Agile requirements using product and sprint backlogs, user stories, story cards, and apply Agile design principles.
4. Estimate, plan, and track Agile projects using techniques such as Planning Poker, story points, velocity, burndown charts, and forecasting methods.
5. Implement Agile testing strategies including TDD, BDD, acceptance and exploratory testing, and analyze quality metrics using automation tools.
6. Describe DevOps principles and lifecycle, implement CI/CD pipelines using DevOps tools, and evaluate the impact on software delivery and team collaboration.

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
0	**Prerequisite**	Basics of Programming and concepts in Software development.		
I	**Introduction to Agile Software Development**	Software Engineering-process framework, Software Development Life Cycle (SDLC) Process Models: The Linear Sequential Model, The Prototyping Model, The RAD Model,	6	CO 1

		Evolutionary Process Models, Agile Process Model, Component-Based Development. Agile Process model, Need of Agile software development, Agile Manifesto and Principles, Stakeholders and Challenges, History of Agile Software Development, Traditional Model vs. Agile Model, Classification of Agile Methods, Agile Manifesto and Principles, Agile Teams, Team Meetings and Ceremonies, Agile Drivers, Capabilities and Values. **Self-learning Topics:** explore advanced Agile concepts such as Agile scaling frameworks (SAFe, LeSS), Agile estimation techniques, and metrics like velocity and burndown charts.		
II	**Agile Processes**	Agile Development Models: Scrum, Extreme Programming, Feature Driven Development, Crystal, Kanban, Adaptive Software Development, Test Driven Development, and Lean Software Development, Methods, Values, Roles, Artifacts, Stakeholders, and challenges. Business benefits of software agility. **Self-learning Topics:** To Study Agile metrics like cycle time, lead time, and cumulative flow diagrams along with tools such as Jira, Azure DevOps, and Trello for practical implementation.	8	CO 2
III	**Agile Requirements Engineering & Design**	Impact of Agile Processes in RE, Product Backlog, Sprint Backlogs, Current Agile Practices, Variance, Managing Unstable Requirements, Requirements Elicitation, Agile Requirements Prioritization, Agile Requirements Modeling and Generation, Concurrency in Agile Requirements Generation, User Stories – Specification and Formats, Story-Cards. Agile Design: Symptoms of poor design, Agile design principles **Self-learning Topics:** Explore advanced backlog management techniques such as user story mapping, impact mapping, and MoSCoW prioritization to deepen their understanding of Agile requirements.	7	CO 3
IV	**Agile Planning & Estimation**	Planning levels; User stories and estimation; Planning Poker; Velocity and Burndown Charts; Agile risk management. Agile forecasting and project Management, velocity, progress tracking, Track Done pattern, Project forecasting, Estimation: Level of Effort, Ideal Time, Hours, Story Points, Team Participation, Sprint Planning and Estimation. **Self-learning Topics:** Advanced Agile forecasting techniques such as Monte Carlo simulations, probabilistic planning, and	6	CO 4

		throughput-based forecasting for more accurate predictions.		
V	**Agile Quality Assurance & Testing**	Agile Metrics, Agile Approach to Quality Assurance. Testing: Functionality Testing, UI Testing, Performance Testing, Security Testing, Agile Testing: Principles of Agile Testers; The Agile Testing Quadrants, Test Driven Development, Acceptance Tests, Refactoring, Agile Automation, Test Automation Pyramid. Case Study – Selenium/ Jira **Self-learning Topics:** Continuous Testing, BDD, exploratory testing, and modern automation tools such as Selenium, Cypress, Postman, and JMeter.	6	CO 5
VI	**DevOps**	DevOps: Introduction, DevOps - Importance and Benefits, DevOps Principles and Practices, 7-C's of DevOps Lifecycle for Business Agility, DevOps and Continuous Testing, Choosing the right DevOps tools and challenges in DevOps Implementation. Mapping App to DevOps-Assessment. Study of various Open Source Tools in DevOps. **Self-learning Topics:** CI/CD pipelines, cloud-based DevOps tools, and real-world case studies on DevOps adoption and automation practices.	6	CO 6

Text Books:

1. Roger Pressman, Software Engineering: A Practitioner's Approach, 6th edition, McGraw Hill International Publication.
2. Robert C. Martin, "Agile Software Development, Principles, Patterns and Practices", First International Edition, Prentice Hall.
3. Introduction to Agile Methods, Sondra Ashmore
4. Deepak Gaikwad, Viral Thakkar. DevOps Tools from Practitioner's Viewpoint.
5. Mike Cohn – Agile Estimating and Planning (Pearson)

Reference Books:

1. David J. Anderson and Eli Schragenheim, ―Agile Management for Software Engineering: Applying the Theory of Constraints for Business Results‖, Prentice Hall, 2003.
2. Craig Larman, Agile and Iterative Development: A Manager's Guide, Addison-Wesley, 2004.
3. Sricharan Vadapalli , DevOps: Continuous Delivery, Integration, and Deployment with DevOps: Dive, , Packt, 2018.

Online References:

Sr. No.	Website Name
1.	https://www.edx.org/course/agile-software-development
2.	https://www.coursera.org/learn/agile-software-development
3.	https://devops.com/most-popular-open-source-devops-tools/
4.	https://github.com/jidibinlin/Free-DevOps-Books-1/blob/master/book/Practical%20DevOps.pdf

Assessment:

Internal Assessment (IA) for 20 marks:

- IA will consist of Two Compulsory Internal Assessment Tests. Each test will be 20 marks. Approximately 40% to 50% of syllabus content must be covered in First IA Test and remaining 40% to 50% of syllabus content must be covered in Second IA Test.

- **Question paper format:**
- Question Paper will comprise of a total of **six questions each carrying 15 marks. Q.1** will be **compulsory** and should **cover maximum contents of the syllabus.**
- **Remaining questions** will be **mixed in nature** (part (a) and part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module randomly selected from all the modules).
- A total of Four questions need to be answered.

University of Mumbai

Course Code: ________________and Course Name:

Date of Exam: Duration: 2 Hours Max. Marks: 60

NB: (1)Question 1 is compulsory. (2) Solve any three questions out of five.		
Q.1	**Solve any three questions out of five: (05 marks each)**	**15**
a)		05
b)		05
c)		05
d)		05
e)		05
Q.2,3, 4,5,6		**15**
a)	Part (a) and Part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module.	07or 08
b)		08 or 07

Note for QP Setters:

1. All COs should be mapped on the basis of the syllabus.
2. Module weightage as per contact hours mentioned in the syllabus.
3. Question 1 is compulsory, and it should include sub-questions covering all modules, with each sub-question carrying five marks.
4. The question paper should contain six questions, each of 15 marks.

....

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical & Oral	Tutorial	Total
2165114	Web and Mobile Application Development	03	–	--	03	–	--	03

Course Code	Course Name	Examination Scheme							
		Theory Marks					Term Work	Practical/ Oral	Total
		Internal assessment			End Sem. Exam	Time Duration (in Hrs.)			
		IA-1	IA-2	IA-1 + IA-2					
2165114	Web and Mobile Application Development	20	20	40	60	02	–	–	100

Rational

This course provides a practical foundation in modern web and mobile development using React, Kotlin, and Flutter. It equips students to design, build, and deploy scalable, cross-platform applications while encouraging exploration of emerging technologies like PWAs, AI/ML APIs, and cloud-based apps.

Course Objectives

1. Understand the core concepts of modern web and mobile application architectures, including REST APIs and development stacks.
2. Develop interactive and efficient front-end interfaces using React and state management tools.
3. Explore Kotlin programming fundamentals and apply object-oriented and functional programming concepts for robust app logic.
4. Build scalable and secure backend services using Kotlin with frameworks
5. Learn the foundations of Flutter and Dart to create cross-platform mobile applications with effective state management.
6. Integrate advanced UI designs, APIs, and Firebase services into Flutter applications for real-world functionality.

Course Outcomes

1. Explain the architectural patterns, design principles, and technology stacks used in modern web and mobile development.
2. Implement front-end applications using React components, routing, hooks, and data management techniques.
3. Apply Kotlin programming constructs, collections, and OOP principles to develop efficient application logic.
4. Design and deploy backend systems in Kotlin using micro services, authentication, and database integration.
5. Create cross-platform mobile apps using Flutter and Dart, incorporating layouts, gestures, and state management.
6. Develop advanced mobile solutions by integrating APIs, Firebase authentication, and real-time database.

Detailed Syllabus:

Sr. No..	Module Title	Contents of the module	Hours	CO
0	**Pre-Requisite Topics**	Basic Concepts: HTML5, CSS3; JavaScript, Client-Server paradigm, Front-end and Back-end development.		–
I	**Core Web and Mobile Development Concepts**	API architectures such as REST APIs. Architectural principles and styles, Design and Architectural patterns (MVC, MVVM). Modern development stacks for Web and mobile (MERN, MEAN, PERN). **Self-Learning:** Overview of Progressive Web Apps (PWAs),	3	CO1
II	**Front-End Development with React**	Foundations of React, JSX, all about Components, React Devtools and Data Flow, Events, Form, Refs, Styling with React, Hooks, rouring, error boundaries, deploying React, initialing projects, Fetching and Caching Data, react portals and accessibility, Redux. **Self-learning Toics:** Performance profiling using React DevTools, Custom Hooks creation	7	CO2
III	**Kotlin Programming Core Concepts**	Basics of kotlin, variables, data types, conditional statements, Functions: structure, calling single-expression functions, Unit, Anonymous functions, standard functions. Null Safety and Exception handling. Collections: List, Set, Map, etc., Collection operations: map, filter, forEach, reduce, fold. Classes: Initialization, Inheritance, Object keyword, nested classes, data classes, enumerated classes, operator overloading, interfaces and abstract classes. Generics and Extensions, Java Interoperability. **Self-learning Topics:** Jamstack architecture, Webassembly, Low-code/No-code development	8	CO3
IV	**Kotlin Backend Development and Advanced Concepts**	Building scalable services with Spring boot, Database integration and ORM, Authentication and Authorization: OAuth2, JWT, session-based authentication, role-based access. Building microservices with Kotlin, Deploying Kotlin backend applications, Testing Monitoring and Scaling. Annotations and Reflection, Domain Specific Languages, Concurrency: Coroutines, coroutine control-flow, concurrency communication. Testing with Kotlin: Assertions, Fixtures and configuration., REST APIs with Ktor, Introduction to Jetpack Composer and Kotlin Multiplatform Mobile. **Self-learning Topics:** API Gateway concept, Dockerizing Kotlin backend applications	8	CO4
V	**Flutter Foundations and Dart Programming**	Introduction to Flutter: Features, Advantages and Disadvantages, Installation (Windows & Mac OS), Creating Simple Application in Android Studio, Architecture of Flutter Applications Flutter Basics: Widgets, Gestures, Introduction to Dart Programming, Variables and Data Types,	7	CO5

		Decision Making, Loops, and Functions, OOP Concepts, Widget Build Visualization Dart Programming Essentials: Basic Programming in Dart, Exception Handling, Debugging, Asynchronous Programming (Futures, Async, Await, Streams) Layouts and State Management: Types of Layout Widgets, Single Child and Multiple Child Widgets, Advanced Layout Applications, Introduction to Gestures, State Management in Flutter, Ephemeral and Application State, Scoped Model, Navigation, and Routing. **Self-learning Topics:** Exploring the internals of **Flutter** framework, Mixins and advanced OOP.		
VI	**Advanced UI, API Integration, and Firebase: Flutter**	1. UI Development: Styles, Assets, and Fonts, Model, API, and Media Query, Working with Lists and Grids, Implementing Animations, Customizing UI and Drawing/Painting, Using Flutter Packages to Enhance UI 2. API and Package Development: Introduction to API and Packages, Types of Packages and Using Dart Packages, developing a Flutter Plugin Package, Accessing REST API, Accessing Product Service API 3. Flutter and Firebase: Introduction to Firebase and Its Features, Integrating Firebase with Flutter Apps, Implementing User Authentication with Firebase, Using Firestore for Real-time Database **Self-learning Topics:** AI/ML API integration, Cloud-based mobile apps, Mobile E-commerce	6	CO6

Text Books

1. Learning React: Modern Patterns for Developing React Apps – Alex Banks & Eve Porcello, O'Reilly (2023).
2. Beginning ReactJS Foundations: Building user interfaces with ReactJS - Chris Minnick, Wrox publication.
3. Kotlin Programming: The Big Nerd Ranch Guide by Josh Skeen and David Greenhalgh.
4. Kotlin for Backend development: Build scalable REST APIs, Microservices and Web Applications with Ktor, Spring Boot and Coroutines, Stokes J. Harrett.
5. Kotlin In-Depth: A Guide to a Multipurpose Programming Language for Server-Side, Front-End, Android, and Multiplatform Mobile - Aleksei Sedunov, BPB Publications.
6. Simplifying Application Development with Kotlin Multiplatform Mobile - Robert Nagy Packt publication
7. Beginning App Development with Flutter Create Cross-Platform Mobile Apps — Rap Payne, Apress
8. Flutter Cookbook, by Simone Alessandria, Brian Kayfitz, O'Reilly

Reference Books

1. Full-Stack Web Development with Node, Express, and React – Adam Freeman, Apress (2022)
2. Head First Kotlin - Dawn Griffiths & David Griffiths, O'Reilly.
3. Kotlin Development Complete Guide create 45 android applications including Jetpack Composer – Henry Codwell.
4. Adam Bretz and Colin J. Ihrig, "Full Stack JavaScript Development with MEAN", SitePoint Pty. Ltd., 2015 Edition.
5. Android UI development with Jetpack Compose - Thomas Kunneth, Packt Publishing.
6. Scalable Android applications in Kotlin.

7. E-Commerce, Pondicherry university

Online Resources:

1. https://www.udacity.com/course/developing-android-apps-with-kotlin--ud9012
2. https://www.coursera.org/learn/advanced-programming-in-kotlin
3. https://www.coursera.org/learn/advance-kotlin-for-multi-platform-development
4. https://www.pluralsight.com/paths/android-development-with-kotlin-fundamentals
5. https://github.com/androiddevnotes/awesome-android-learning-resources
6. https://www.udemy.com/course/android-app-development-with-kotlin-beginner-to-advanced
7. https://www.codecademy.com/learn/learn-kotlin
8. https://www.codingblocks.com/android-development-using-kotlin.html
9. https://www.geeksforgeeks.org/flutter/flutter-tutorial/
10. https://www.youtube.com/playlist?list=PL4cUxeGkcC9jLYyp2Aoh6hcWuxFDX6PBJ

Assessment:

Internal Assessment (IA) for 20 marks:

- IA will consist of Two Compulsory Internal Assessment Tests. Each test will be 20 marks. Approximately 40% to 50% of syllabus content must be covered in First IA Test and remaining 40% to 50% of syllabus content must be covered in Second IA Test.

➢ **Question paper format:**

NB: (1) Question 1 is compulsory. (2) Solve any three questions out of five.		
Q.1	**Solve any three questions out of five: (05 marks each)**	**15**
a)		05
b)		05
c)		05
d)		05
e)		05
Q.2,3, 4,5,6		**15**
a)	Part (a) and Part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module.	07or 08
b)		08 or 07

Note for QP Setters:

1. All COs should be mapped on the basis of the syllabus.
2. Module weightage as per contact hours mentioned in the syllabus.
3. Question 1 is compulsory, and it should include sub-questions covering all modules, with each sub-question carrying five marks.
4. The question paper should contain six questions, each of 15 marks.

****

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2165115	User Experience Design with VR	3	-		3	-		3

Course Code	Course Name	Theory					Term Work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-I	IAT-II	IAT-I + IAT-II					
2165115	User Experience Design with VR	20	20	40	60	2	-	--	100

Rationale:
The User Experience Design with VR, AI, and DS syllabus provides a forward-looking framework for developing creative, analytical, and ethical designers who can shape the future of human-technology interaction. By merging immersive technologies with data-driven intelligence, the course empowers learners to design experiences that are adaptive, meaningful, and human-centered.

Course Objectives:

1. To understand the fundamentals of interface design and core elements of user experience.
2. To learn the stages of the UX design life cycle and their importance in building user-centered systems.
3. To explore the UX design process, including information architecture, prototyping, and testing for web and mobile applications.
4. To study methods for evaluating and improving user experience using usability and heuristic techniques.
5. To understand the basic principles, components, and working of Virtual Reality systems.
6. To explore various applications of Virtual Reality and analyze the human factors involved in immersive environments.

Course Outcomes On successful completion, of course, learner/student will be able to:

1. Explain the basic concepts of interface design and elements of user experience.
2. Describe the complete UX design life cycle and its relevance in real-world projects.
3. Apply the UX design process to create and test interactive prototypes for web and mobile systems.
4. Evaluate user experience using different usability and heuristic evaluation techniques.
5. Illustrate the core principles and components of Virtual Reality systems.
6. Analyze various Virtual Reality applications and interpret human performance aspects in VR environments.

DETAILED SYLLABUS:

Sr. No	Name of Module	Detailed Content	Hours	CO Mapping
0	**Prerequisite**	**Computer Graphics**	-	-
I	**Introduction**	Introduction to interface design, Understanding and conceptualizing Interface, understanding user's conceptual cognition, Core Elements of User Experience, Working of UX elements. **Self-learning Topics:** Norman's principles of design by Don Norman, Human perception and	04	**CO1**

		visual processing		
II	**The UX Design - Life Cycle**	What is UX, Ubiquitous interaction, Emerging desire for usability, From usability to user experience, Emotional impact as part of the user experience, User experience needs a business case, Roots of usability. Introduction, A UX process lifecycle template, Choosing a process instance for your project, The system complexity space, Meet the user interface team, Scope of UX presence within the team, More about UX life cycle. **Self-learning Topics**: Historical evolution of UX, UX in IoT and smart device.	07	**CO2**
III	**The UX Design Process**	Introduction, The system concept statement, User work activity gathering, Look for emotional aspects of work practice, Abridged contextual inquiry process, Data-driven vs. model driven inquiry, Contextual Analysis, Extracting Interaction Design Requirements, Constructing Design Information Models. Information ,Architecture and Interaction Design and Prototyping Introduction, Design paradigms, Design thinking, Design perspectives, User personas, Ideation, Sketching, More about phenomenology, Mental Models and Conceptual Design, Wireframe, Prototyping Web UX Design Principles, Mobile UX Design Principles, Prototyping and Testing for Web/Mobile. **Self-learning Topics:** Translating user work activity into design requirements, Use case modeling for interaction design.	10	**CO3**
IV	**The UX Evolution and Improvement**	UX Evaluation and Improve UX Goals, Metrics and Targets, UX Evaluation Techniques.- Formative vs summative ,types of formative and informal summative evaluation methods, types of evaluation data, some data collection techniques Rapid Evaluation Methods: Design walkthroughs and reviews UX Inspection, Heuristic evaluation, a UX inspection method, practical approach to UX Inspection, Do UX Evaluation rite, Quasi-empirical UX evaluation Questionnaires, Specialized rapid UX evaluation methods. **Self-learning Topics:** Iterative UX improvement lifecycle, Role of evaluation in Agile & design thinking, UX maturity models.	06	**CO4**
V	**Introduction to Virtual Reality**	Defining Virtual Reality, The three I's of Virtual Reality, History, The five classic components of a VR system, Input Devices: Trackers, Navigation and Gesture Interfaces, Output Devices: Graphics, Three dimensional sound and Haptic displays. **Self-learning Topics:** Interaction design in immersive environments, Imagination and cognitive engagement in VR, Gesture recognition	06	**CO5**

		concepts.		
VI	**Applications Virtual Reality**	Modeling: Geometric modeling, Kinematics modeling, Physical Modeling, Behavior modeling, Model management. Human factor in VR: Methodology and terminology, User performance studies. Traditional VR Applications, Emerging Applications of VR. **Self-learning Topics:** Collision detection algorithms, Real-time rendering optimization, Forward and inverse kinematics concepts.	06	**CO6**

Textbooks:

1. The UX Book by Rex Hartson and PardhaPyla, MK Publication Smashing UX Design by Jesmond Allen and James Chudley, John Wiley & Sons
2. LaValle, S. M. (2017). Virtual Reality. Cambridge University Press.
3. Alan Dix, Janet Finlay, Gregory Abowd, Russel Beale, “Human–Computer Interaction”, Pearson, 2009.
4. Rogers Sharp Preece, “Interaction Design: Beyond Human Computer Interaction”, 5th Edition, Wiley, 2019.
5. Kalbande Dhananjay R, Kanade P., Iyer S., Galitz’s, “Human Machine Interaction”, Wiley Publications, 2015.

Reference Books:

1. Jeff Johnson, “Designing with the mind in mind”, 2nd Edition, Morgan Kaufmann Publication, 2014.
2. Brian Fling, “Mobile Design and Development”, 1 st Edition, O’Reilly Media Inc., 2009.
3. Wilbert O. Galitz, “The Essential Guide to User Interface Design”, Wiley publication, 2002.

Online References:

Sr. No.	Website Name
1.	https://archive.nptel.ac.in/courses/124/107/124107008/
2.	https://nptel.ac.in/courses/106106138
3.	https://www.coursera.org/specializations/virtual-reality
4.	https://msl.cs.uiuc.edu/vr/vrbook.pdf

Assessment:

Internal Assessment (IA) for 20 marks:

- IA will consist of Two Compulsory Internal Assessment Tests. Each test will be 20 marks. Approximately 40% to 50% of syllabus content must be covered in First IA Test and remaining 40% to 50% of syllabus content must be covered in Second IA Test.

➢ **Question paper format:**

- Question Paper will comprise of a total of **six questions each carrying 15 marks. Q.1** will be **compulsory** and should **cover maximum contents of the syllabus.**
- **Remaining questions** will be **mixed in nature** (part (a) and part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module randomly selected from all the modules).
- A total of Four questions need to be answered.

University of Mumbai

Course Code: ________________and Course Name:

Date of Exam: Duration: 2 Hours Max. Marks: 60

NB: (1) Question 1 is compulsory. (2) Solve any three questions out of five.		
Q.1	**Solve any three questions out of five: (05 marks each)**	**15**
a)		05
b)		05
c)		05
d)		05
e)		05
Q.2,3, 4,5,6		**15**
a)	Part (a) and Part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module.	07or 08
b)		08 or 07

Note for QP Setters:

1. All COs should be mapped on the basis of the syllabus.
2. Module weightage as per contact hours mentioned in the syllabus.
3. Question 1 is compulsory, and it should include sub-questions covering all modules, with each sub-question carrying five marks.
4. The question paper should contain six questions, each of 15 marks.

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2165116	Computer Network	3	-	-	3	-	-	3

Course Code	Course Name	Theory					Term Work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-I	IAT-II	IAT-I + IAT-II					
2165116	Computer Network	20	20	40	60	2	-	--	100

Rationale:
This course provides foundational knowledge of computer networks, covering architectures, topologies, transmission media, and devices. It develops skills in error control, IP addressing, routing, and socket programming while introducing advanced concepts like VPN, SDN, NFV, and data center networks to prepare students for modern networking applications.

Course Objectives:
1. Understand the fundamentals of computer networks, their types, architectures, topologies, and communication models.
2. Explain the design issues, error control, and media access mechanisms used in the Data Link Layer.
3. Analyze addressing, routing techniques, and protocols implemented in the Network Layer for data delivery.
4. Describe transport layer mechanisms including connection establishment, flow control, and congestion management.
5. Illustrate the functionality of common application layer protocols and their role in network communication.
6. Explore emerging networking technologies.

Course Outcomes: On successful completion of course, learner will be able to
1. Explain the basic concepts of computer networks, their architectures, topologies, transmission media, and network devices.
2. Demonstrate understanding of Data Link Layer design issues and implement error detection, correction, and flow control mechanisms.
3. Apply IPv4 and IPv6 addressing, subnetting, and routing algorithms for efficient packet delivery across networks.
4. Describe the working of transport layer protocols and implement basic socket programming for process-to-process communication.
5. Analyze the structure and operation of major application layer protocols such as HTTP, DNS, FTP, and SMTP.
6. Evaluate and discuss the concepts of VPN, SDN, NFV, and Data Center Networks as part of advanced networking technologies.

Prerequisite: Operating System Fundamentals

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
I	**Introduction To Computer Networks**	Definition, **Types of Networks**: Local area networks (LAN), Metropolitan area networks (MAN), Wide area networks (WAN), Wireless networks, Networks Software, Protocol, Design issues for the Network layers. **Network Models**: The OSI Reference Model, TCP/IP Model, Network Topologies, Types of Transmission Medium. **Network Architectures**: Client-Server, Peer to Peer, Hybrid. **Network Devices**: Bridge, Switch, Router, Gateway, Access Point. **Self-learning Topics:** Study of Campus wide networking	**6**	CO1
II	**Data Link Layer**	Introduction, Design Issues (Services to Network Layer, Framing. Error control, Flow control), Error Detection and correction, Parity Bits, Checksum, Hamming Codes and CRC. Flow Control Protocols: Unrestricted Simplex, Stop and Wait, Sliding Window Protocols. Elementary Data Link protocols MAC Sub layer: Multiple Access Protocols: Pure and Slotted ALOHA, CSMA, CSMA/CD, CSMA/CA, WDMA, Introduction to IEEE standards (IEEE 802.3, IEEE 802.11). Self-learning Topics: Demonstration of DLL protocols on Simulator	**8**	CO2
III	**Network Layer**	Network Layer design issues, **Switching Techniques**: Circuit switching, Message Switching, Packet Switching. **IP Protocol**: Classes of IP (Network addressing), IPv4 , IPv6, Network Address Translation, Sub-netting, Supernetting, CIDR. **Network layer Protocols**: ARP, RARP, ICMP, IGMP. **Network Routing and Algorithms**: Static Routing, Dynamic Routing, Distance Vector Routing, Link State Routing. **Routing Protocols**: RIP, OSPF, BGP **Self-learning Topics:** Count-to-Infinity Problem, AODV AND DSR routing protocol	**9**	CO 3
IV	**Transport Layer**	Process to Process Delivery, Services, Socket Programming. **Elements of Transport Layer Protocols**: Addressing, Connection establishment, Connection release, CP timers and TCP state transition diagram, Flow control and buffering, Multiplexing, Congestion Control: slow start protocol. **Transport Layer Protocols**: TCP and UDP, Quality of Service (QoS). **Self-learning Topics:** Demonstration of Transport layer protocols on Simulator.	**07**	CO4

V	**Application Layer**	Introduction, Web and HTTP, Web Caching, DNS, Email: SMTP, POP3, Webmail, FTP, TELNET, DHCP, SNMP. **Self-learning Topics:** GOOGLE DNS	**04**	CO5
VI	**Emerging & Advanced Topics**	VPN (Virtual Private Networks) - Types and basic principle. **Software-Defined Networking (SDN):** Concepts, Control and Data Planes, OpenFlow protocols and controllers. **Network Function Virtualization (NFV):** Concepts and Benefits. **Data Center Networks (DCNs):** Topologies (Fat Tree - Concept) **Self-learning Topics:** SD-WAN	**05**	CO6

Text Books:

1. Fourauzan B., "Data Communications and Networking", 5th Edition, Tata McGraw-Hill, Publications, ISBN:0–07 – 058408 – 7
2. Andrew S. Tanenbaum, Computer Networks, 5th Edition, Pearson India, 2012.
3. A. S. Godbole and A. Kahate, Data Communications & Networks, 2nd ed. New Delhi: McGraw Hill Education, 2011.

References:

1. Kurose, Ross, "Computer Networking a Top Down Approach Featuring the Internet", Pearson, ISBN-10: 0132856204
2. L. Peterson and B. Davie, "Computer Networks: A Systems Approach", 5th Edition, Morgan-Kaufmann, 2012.
3. Douglas E. Comer & M.S Narayanan, "Computer Network & Internet", Pearson Education

Online References:

Sr. No.	Website Name
1.	NPTEL Swayam Course: https://onlinecourses.nptel.ac.in/noc22_cs19/
2.	CCNA: Introduction to Networks https://www.netacad.com/courses/ccna-introduction-networks?courseLang=en-US

Assessment:

Internal Assessment (IA) for 20 marks:

- IA will consist of Two Compulsory Internal Assessment Tests. Each test will be 20 marks. Approximately 40% to 50% of syllabus content must be covered in First IA Test and remaining 40% to 50% of syllabus content must be covered in Second IA Test.

➢ **Question paper format:**

- Question Paper will comprise of a total of **six questions each carrying 15 marks. Q.1** will be **compulsory** and should **cover maximum contents of the syllabus.**
- **Remaining questions** will be **mixed in nature** (part (a) and part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module randomly selected from all the modules).
- A total of Four questions need to be answered.

University of Mumbai

Course Code: _______________and Course Name:

Date of Exam: Duration: 2 Hours Max. Marks: 60

NB: (1) Question 1 is compulsory. (2) Solve any three questions out of five.		
Q.1	**Solve any three questions out of five: (05 marks each)**	**15**
a)		05
b)		05
c)		05
d)		05
e)		05
Q.2,3,4,5,6		**15**
a)	Part (a) and Part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module.	07or 08
b)		08 or 07

Note for QP Setters:

1. All COs should be mapped on the basis of the syllabus.
2. Module weightage as per contact hours mentioned in the syllabus.
3. Question 1 is compulsory, and it should include sub-questions covering all modules, with each sub-question carrying five marks.
4. The question paper should contain six questions, each of 15 marks.

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2165117	**Statistics for Machine Learning and Data Science Lab**	**-**	**2**	**-**	**-**	**1**	**-**	**1**

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Practical/ Oral	Total
		Internal assessment			End Sem. Exam			
		IAT-I	IAT-II	IAT-I + IAT-II				
2165117	**Statistics for Machine Learning and Data Science Lab**	--	--	--	--	25	25	50

Rationale- The laboratory component reinforces theoretical understanding through hands-on data experiments using real-world datasets and tools such as Python (NumPy, Pandas, Scikit-learn). Students gain skills in exploratory analysis, sampling, hypothesis testing, regression, clustering, and model evaluation. It encourages experiential learning, data storytelling, and critical interpretation, building confidence to apply statistical learning methods for decision-making and AI development.

Lab Objectives: After completing the lab, students will be able to:

1. Apply descriptive statistics and sampling to analyze and visualize data.
2. Perform estimation, hypothesis testing, and resampling-based inference.
3. Implement regression and classification models and interpret diagnostic results
4. Apply unsupervised learning methods including PCA and clustering.
5. Evaluate ML models using appropriate metrics and cross-validation.
6. Interpret models with transparency, fairness, and ethical considerations.

Lab Outcomes: Upon successful completion, the student will be able to:

1. Perform EDA and sampling-based data exploration.
2. Conduct hypothesis testing and resampling analysis.
3. Implement regression and classification models.
4. Apply clustering and PCA for unsupervised analysis.
5. Evaluate statistical learning models using performance metrics.
6. Ensure interpretability and ethical awareness in statistical modeling.

Prerequisite: Python

Suggested List of Experiments (10-12 Experiments must be conducted)

Sr No	List of Experiments	Hrs
01	Exploratory Data Analysis: descriptive stats, histograms, correlation matrix	02
02	Data preprocessing: missing values, encoding, scaling, outlier detection.	02
03	Sampling methods: SRS, stratified, cluster sampling + dataset demonstration	02
04	Demonstration of Central Limit Theorem using simulations.	02
05	Hypothesis testing (t-test, chi-square, ANOVA) on real dataset.	02
06	Bootstrap confidence intervals & permutation tests.	02
07	MLE estimation for Normal & Bernoulli distributions.	02
08	Linear regression & diagnostics (VIF, residual plots, Cook's distance)	02
09	Ridge, Lasso, Elastic Net with hyperparameter tuning.	02
10	Classification models: Logistic regression, KNN, Naïve Bayes, ROC analysis.	02
11	Decision Trees & Random Forest; feature importance.	02
12	PCA + clustering (K-means, hierarchical)	02
13	Interpretability: SHAP/LIME/PDP/ICE (conceptual + coding)	02
14	Fairness & bias detection metrics (equalized odds, parity)	02

Sr No	List of Assignments / Tutorials	Hrs
01	Mini project on complete ML pipeline (EDA → model → evaluation)	02
02	Comparison of resampling methods (bootstrap vs. permutation)	02
03	Regularization comparison (Ridge / Lasso / ElasticNet) on multiple datasets	02
04	Bias–variance tradeoff study	02
05	Ethical ML case study (fairness, interpretability, transparency)	02

Students must use at least 3 real-world datasets from Kaggle/UCI/OpenML
Common recommended datasets:

- Iris
- Titanic
- Breast Cancer Wisconsin
- California Housing
- Mall Customers
- Bank Marketing

Tools Requirement- Python libraries: NumPy, Pandas, Matplotlib/Seaborn, Scikit-Learn, SciPy, SHAP, LIME
All implementations must follow responsible and ethical data practices.

Text Books (For Lab Use):

1. Wes McKinney – *Python for Data Analysis*, O'Reilly.
2. Aurélien Géron – *Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow*, O'Reilly.
3. Peter Bruce & Andrew Bruce – *Practical Statistics for Data Scientists*.
4. James et al. – Introduction to Statistical Learning (ISLR2).

References:

1. Charu Aggarwal – *Probability & Statistics for ML*.
2. Scikit-learn documentation (https://scikit-learn.org/stable/).

Assessment :

Term Work: Term Work shall consist of at least 10 to 12 practical based on the above list. Also, Term work Journal must include at least 2 assignments.

Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)

Practical & Oral Exam: An Oral & Practical exam will be held based on the above syllabus**.**

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2165118	Artificial Intelligence & Soft Computing Lab		2	-	-	1	-	1

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Practical/ Oral	Total
		Internal assessment			End Sem. Exam			
		IAT-I	IAT-II	IAT-I + IAT-II				
2165118	Artificial Intelligence & Soft Computing Lab	--	--	--	--	25	25	50

Lab Objectives: Six Lab Objectives

1. To introduce practical applications of AI fundamentals including intelligent agents, PEAS-based problem formulation, and basic PROLOG programming for knowledge representation and reasoning.
2. To implement classical search algorithms such as BFS, DFS, and heuristic-based A* for solving real-world planning and navigation problems.
3. To simulate foundational Neural Network models such as perceptron classification and explore their role in advanced neural architectures including Transformer-based systems.
4. To explore Generative AI applications including LLM-based text generation along with RAG and CAG enhancements.
5. To design and analyze Soft Computing models including fuzzy sets, membership functions, and fuzzy logic controllers for approximate reasoning and adaptive decision-making.
6. To develop and evaluate Hybrid Neuro-Fuzzy systems integrating neural learning with fuzzy inference mechanisms for intelligent automated solutions.

Lab Outcomes: Six Lab Outcomes (Based on Blooms Taxonomy)

1. Explain AI fundamentals and formulate problems using the PEAS model and PROLOG.
2. Apply search algorithms to achieve goal-oriented problem solving.
3. Implement basic neural network learning using perceptron models and interpret Transformer evolution conceptually.
4. Construct LLM-based text generation and enhance it using RAG & CAG techniques.
5. Design and simulate Fuzzy Logic and Neural Network models for adaptive decision making.
6. Create and evaluate a Hybrid Neuro-Fuzzy system for intelligent learning.

DETAILED SYLLABUS: Syllabus related Lab experiment must be considered and mapped with Blooms Taxonomy.

Total six modules for each subject lab (13 weeks) to be distributed among six modules.

Sr. No.	Module	Detailed Content	LO Mapping
0	**Prerequisite**	**Engineering Mathematics:** Matrix Algebra, **Discrete Structures:** Set Theory and Logic	--

I	**Introduction to Artificial Intelligence and Problem Formulation**	Introduction to AI — applications in engineering; Intelligent agents and PEAS model; Problem formulation in AI systems. Introduction to Data Centers and Cloud Services for AI deployment. Basics of PROLOG — facts, rules, queries; Ethical and Explainable AI. **Self-Learning Topics:** Applications of AI in autonomous systems, PEAS modeling.	**LO1**
II	**Search Strategies and Classical AI Problem Solving**	Search-based problem-solving techniques: uninformed (BFS, DFS) and informed (Best-First, A*) algorithms. Problem representation and use of heuristics in goal-driven tasks. Adversarial search: Mini-max Search. **Self-Learning Topics:** Bidirectional & iterative deepening search.	**LO2**
III	**Neural Networks Fundamentals**	Biological vs. Artificial neuron; Activation functions; Single-layer Perceptron and forward propagation; Introduction to advanced neural architectures with the basic concept of Transformer models as an evolution of neural network design. **Self-Learning Topics:** Gradient Descent intuition.	**LO3**
IV	**Generative AI and Advanced Architectures (LLM + RAG & CAG)**	Introduction to Generative AI workflow. Transformer architecture — Attention mechanism. Language Model Classification: Foundation Models (FM), Large Language Models (LLM), Small Language Models (SLM), Micro Language Models (μLM). Retrieval-Augmented Generation (RAG) using vector databases (FAISS/ChromaDB). Cache-Augmented Generation (CAG) for performance improvement. Text generation/summarization using GPT-2/Mistral/LLaMA-2via HuggingFace/Ollama. **Self-Learning Topics:** Prompt engineering and memory-optimized GenAI pipelines.	**LO4**
V	**Soft Computing Techniques — Fuzzy Logic**	Soft computing concepts, uncertainty and approximate reasoning. Fuzzy sets and membership functions; Design of Fuzzy Logic Controller (Mamdani/Sugeno). Implementation of a basic McCulloch–Pitts Neural Network for logic operations (AND/OR). **Self-Learning Topics:** Adaptive fuzzy systems and rule optimization.	**LO5**
VI	**Learning Rules and Hybrid AI Systems**	Basic learning rules: Perceptron, Delta, Hebbian. Concept of Neuro-Fuzzy System- architecture and benefits (Hybrid Neuro-Fuzzy System integrating fuzzy logic with neural learning). Comparative evaluation with standalone models. **Self-Learning Topics:** Applications of hybrid systems in decision support.	**LO6**

Text Books:

1. Stuart J. Russell and Peter Norvig, "Artificial Intelligence A Modern Approach —Second Edition" Pearson Education.
2. Elaine Rich and Kevin Knight —Artificial Intelligence‖ Third Edition, Tata McGraw-Hill Education Pvt. Ltd., 2008.
3. S. Rajasekaran & G. A. Vijayalakshmi Pai — *Neural Networks, Fuzzy Logic, and Genetic Algorithms: Synthesis and Applications*, PHI, Second Edition
4. J. S. R. Jang, C. T. Sun & E. Mizutani — *Neuro-Fuzzy and Soft Computing,* Pearson Education.

References:

1. Ivan Bratko —PROLOG Programming for Artificial Intelligence‖, Pearson Education, Third Edition.
2. S. N. Sivanandam & S. N. Deepa — *Principles of Soft Computing*, Wiley India.
3. Sebastian Raschka — *Build a Large Language Model (From Scratch) (*MIT Press Draft).
4. OpenAI / Hugging Face — *Introduction to Transformers and LLMs (Free Online).*
5. Jay Alammar — *The Illustrated Transformer* (jalammar.github.io).

Online Resources:

Sr. No.	Website Name
1.	https://www.coursera.org/learn/neural-networks-deep-learning
2.	https://www.coursera.org/courses?query=generative+ai
3.	https://www.coursera.org/learn/introduction-to-ai

List of Experiments.

Sr No	List of Experiments	Hrs
01	Select a problem statement relevant to Artificial Intelligence: Identify the problem, describe it using PEAS (Performance, Environment, Actuators, Sensors), and perform problem formulation	02
02	Introduction to AI Programming Language: Write and execute basic programs in PROLOG.	02
03	Implement uninformed search algorithms (BFS, DFS) to reach a goal state.	02
04	Implement any one of the Informed search techniques E.g. A* algorithm for 8 puzzle problem	02
05	Implement adversarial search using min-max algorithm.	02
06	Creating/building a neural network using basic language (C Programming/Java Programming)	02
07	Train a perceptron to learn AND, OR, XOR gates.	02
08	To implement Basic Supervised / Unsupervised Neural Network learning rules for a problem	02
09	Exploration and Implementation of Generative AI Techniques using Open-Source Tools and Frameworks	02
10	Developing Language Models (Small Language Models (SLM), Micro Language Models (μLM).)	02
11	Design and Implementation of a Retrieval-Augmented Generation (RAG) Framework for Contextual Question Answering	02
12	Development of a Cache-Augmented Generation (CAG) System for Efficient AI Responses	02
13	Implement a simple Fuzzy Logic Controller for real-world control problems.	02
14	Implement Mamdani/Sugeno Fuzzy Inference Systems (FIS) using MATLAB / Python.	02
15	Developing a Hybrid Neuro-Fuzzy System	02

Sr No	List of Assignments / Tutorials	Hrs
01	Applications and Impact of Artificial Intelligence in Modern Engineering	02
02	Role of Heuristics in Enhancing Search Efficiency in AI Systems	02
03	Open-Source Frameworks and Tools for Generative AI Development	02
04	Retrieval-Augmented Generation (RAG): Architecture and Applications	02
05	Cache-Augmented Generation (CAG): Concept and Performance Optimization in AI Systems	02
06	Fuzzy Logic Controllers and Neural Network Approaches for Adaptive Decision-Making	02
07	Design and Applications of Hybrid Neuro-Fuzzy Systems in Intelligent Automation	02
08	Comparative Study of Learning Approaches in Neural Networks and Fuzzy Logic Systems	02

Assessment:
Term Work: Term Work shall consist of at least 10 to 12 practical's based on the above list. Also, Term work Journal must include at least 2 assignments.
Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)
Practical & Oral Exam: An Oral & Practical exam will be held based on the above syllabus.

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2165119	Agile Software Development & DevOps Lab	-	2	-	-	1	-	1

Course Code	Course Name	Examination Scheme							
		Theory Marks				Term Work	Practical/ Oral	Total
		Internal assessment			End Sem. Exam			
		IAT-I	IAT-II	IAT-I + IAT-II				
2165119	Agile Software Development & DevOps Lab	--	--	--	--	25	25	50

Lab Objectives: The student will be able:

1. To understand and implement the Agile Software Development Lifecycle using Jira and integrate it with DevOps tools for continuous project management.
2. To perform version control operations using Git and apply collaboration workflows for software development.
3. To design and configure Continuous Integration / Continuous Deployment pipelines using Jenkins, GitHub Actions, and containerization tools.
4. To automate provisioning, configuration, container orchestration, and cloud resource management using Ansible, Docker, Kubernetes, and Terraform.
5. To study and implement MLOps workflows including model tracking, deployment, and data pipeline automation using MLflow and Airflow.
6. To integrate and implement a complete end-to-end DevOps pipeline through a mini project covering planning, building, testing, deployment, and monitoring.

Lab Outcomes: Upon completion, the student will be able to:

1. Apply Agile methodologies and use Jira for sprint planning, task tracking, and integrating with DevOps workflows.
2. Use Git effectively for version control, source code management, and collaborative team development.
3. Build and manage automated CI/CD pipelines for application development, testing, packaging, and deployment.
4. Containerize, orchestrate, and deploy applications using Docker, Docker Compose, Kubernetes, and cloud infrastructure tools.
5. Implement ML lifecycle management and automated workflows using MLflow, Airflow, and monitoring tools like Prometheus/Grafana/Nagios.
6. Develop and demonstrate an end-to-end DevOps pipeline mini-project integrating all stages of DevOps from code commit to deployment and monitoring.

DETAILED SYLLABUS:

Sr. No.	Module	Detailed Content	LO Mapping
0	**Prerequisite**	Knowledge of Programming, Computer Networks, Web Programming.	

I	**Agile Project Management & DevOps Integration**	Fundamentals of Agile and Scrum, Jira for Agile project lifecycle management, Creating epics, user stories, tasks, sprints, Planning, assigning, tracking sprint progress, Integrating Jira with CI/CD pipelines Agile reporting: Burndown chart, velocity chart. **Self-learning Topics:** Kanban vs Scrum, Agile ceremonies: Sprint review, retrospective	LO 1
II	Version Control & Collaboration using Git	Introduction to Git, GitHub, Basic operations: clone, commit, push, pull, Branching strategies and merging, Git workflows: Feature branching, Git Flow, Handling merge conflicts, GitHub Actions basics **Self-learning Topics:** GitHub Issues & Project Boards Webhooks in GitHub	LO 2
III	CI/CD Automation using Jenkins & Containerization using Docker	Understanding CI/CD and DevOps pipelines, Jenkins installation and job creation, GitHub webhook integration, CI pipeline: build → test → package, Docker fundamentals: images, containers, Creating Dockerfiles, Multi-service deployment using Docker Compose **Self-learning Topics:** Jenkinsfile (Pipeline as Code), Publishing images to Docker Hub/ECR	LO 3
IV	Configuration Automation, Orchestration & Cloud Provisioning	Ansible automation: inventory, playbooks, roles, Provisioning and configuring applications, Kubernetes concepts: Pods, Deployment, Service, Ingress, Managing containerized apps in Kubernetes, Terraform basics: providers, resources, state files, Automating cloud infrastructure creation. **Self-learning Topics:** Helm Charts, Terraform Cloud / Remote state management	LO 4
V	Monitoring, Serverless Computing & MLOps Fundamentals	Monitoring tools: Prometheus, Grafana, Nagios, Exporters, dashboards, alerting, AWS Lambda fundamentals and deployment, MLOps lifecycle overview, MLflow for tracking experiments, versioning & deployment, Airflow basics: DAGs, tasks, operators, scheduling **Self-learning Topics:** CloudWatch metrics for serverless apps, Airflow for ML/ETL automation	LO 5
VI	End-to-End DevOps Pipeline	Designing a full DevOps automation pipeline, Integrating Agile → Git → CI → CD → Containers → Kubernetes → Monitoring, Cloud deployment using AWS/GCP/Azure, Pipeline security basics (DevSecOps), Documentation, demonstration & evaluation. **Self-learning Topics:** Industry-grade DevOps architecture, Cost-efficient cloud deployment strategies	LO 6

Text Books:

1. The DevOps handbook : how to create world-class agility, reliability, and security in technology organizations by Kim, Gene.
2. Robert C. Martin, "Agile Software Development, Principles, Patterns and Practices", First International Edition, Prentice Hall.
3. Deepak Gaikwad, Viral Thakkar. DevOps Tools from Practitioner's Viewpoint.

References

1. DevOps: Continuous Delivery, Integration, and Deployment with DevOps: Dive . By Sricharan Vadapalli, Packt, 2018

Online Resources:

Sr. No.	Website Name
1.	https://devops.com/most-popular-open-source-devops-tools/
2.	https://github.com/bregman-arie/devops-resources?
3.	https://seblum.github.io/mlops-engineering-book/?

List of Experiments.

Sr No	List of Experiments	Hrs
01	To study and Implement Agile Software Project lifecycle using Jira with DevOps Pipeline Integration.	2
02	To perform version control operations using Git.	2
03	To set up a Continuous Integration Pipeline using Jenkins/GitHub Actions.	2
04	To automate software provisioning and configuration using Ansible	2
05	To containerize applications using Docker.	2
06	To deploy multi-service applications using Docker Compose.	2
07	To deploy and manage containers using Kubernetes.	2
08	To automate application deployment using Github/ Jenkins.	2
09	To provision cloud infrastructure using Terraform.	2
10	To deploy a serverless function using AWS Lambda.	2
11	To implement monitoring using Prometheus / Grafana / Nagios.	2
12	To study and Implement machine learning lifecycle with model development & deployment using MLFlow.	2
13	To study and Implement designing, scheduling, and monitoring workflows and data pipelines using AirFlow.	2
14	To design and implement an end-to-end DevOps pipeline through a mini project.	2

List of Assignments (Based on Theory Syllabus):

Sr No	List of Assignments / Tutorials	Hrs
	Assignment 1	
01	Choose any one application domain (E-commerce / Healthcare / College ERP / Banking / Food Delivery) and Create Agile Requirements using User Stories & Acceptance Criteria.	2
02	For the user stories created in Assignment 1 do Sprint Planning, Story Point Estimation & Release Planning	
03	Test Case Design using Agile QA Techniques	
	Assignment 2	
01	DevOps Pipeline – CI/CD Setup using GitHub Actions / Jenkins.	2
02	Simulate a Full Agile Process Implementation Using Jira.	
03	Integrate Agile Planning with DevOps Automation.	

Assessment :

Term Work: Term Work shall consist of at least 10 to 12 practical based on the above list. Also, Term work Journal must include at least 2 assignments.

Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)

Practical & Oral Exam: An Oral & Practical exam will be held based on the above syllabus.

Subject Code	Subject Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical & Oral	Tutorial	Total
2165120	Web and Mobile Application Development Lab	--	2	--	--	1	--	1

Subject Code	Subject Name	Examination Scheme						
		Theory Marks				Term Work	Practical / Oral	Total
		Internal assessment			End Sem. Exam			
		IAT-I	IAT-II	IAT-I + IAT-II				
2165120	Web and Mobile Application Development Lab	--	--	--	--	25	--	25

Lab Objectives

1. Understand and implement basic programming principles using different languages such as Kotlin, React, and Flutter.
2. Develop web and mobile applications using modern frameworks (React for web and Flutter for mobile).
3. Implement backend services and RESTful APIs using Kotlin with database connectivity.
4. Integrate frontend, backend, and mobile interfaces into a unified full-stack system.
5. Apply basic machine learning concepts to enhance application functionality using self-developed logic or models.
6. Deploy and evaluate the complete project for performance, usability, and scalability.

Lab Outcomes

1. Demonstrate understanding of modern web and mobile development environments and tools.
2. Develop responsive front-end applications using React components and state management.
3. Implement backend services in Kotlin with database integration and secure authentication.
4. Apply microservices, REST APIs, and deployment techniques for scalable applications.
5. Design cross-platform mobile apps using Flutter and integrate them with backend services.
6. Integrate advanced functionalities such as Firebase, API connectivity, and animations to enhance user experience

Guidelines:

- Mentors / Lab instructors should get the labs performed as an internal project-based lab assigned to a group of 2 - 3 students.
- The project development should focus on covering all Web and Mobile development concepts consisting of React Native, Kotlin and flutter topics as mentioned in the syllabus.
- All the following experiments/tasks, which together make up the mini-project.
- Students are expected to implement the projects integrally and sequentially, demonstrating both frontend and backend development, API integration, authentication, and deployment practices.
- Appropriate project management practices shall be utilized during the entire project life-cycle Requirement gathering and designing then following experiments to be implemented

Suggested List of experiments:

Sr. No.	Module Title	Experiment	LO
1	**Core Web & Mobile Development Concepts**	Study and demonstrate REST API architecture with sample CRUD operations using Postman.	LO1
2	**Front-End Development with React**	Create a simple React app with components and JSX; display static product/user data.	LO2
3	**Front-End Development with React**	Implement state management using Hooks and Context API; add form handling and validation.	LO 2
4	**Kotlin Programming Core Concepts**	Write Kotlin programs to demonstrate data types, collections, and OOP concepts (classes, inheritance).	LO 3
5	**Kotlin Programming Core Concepts**	Implement exception handling and use collection operations (map, filter, reduce).	LO 3
6	**Kotlin Backend Development**	Create RESTful APIs using Spring Boot for basic CRUD operations (users/products).	LO 4
7	**Kotlin Backend Development**	Integrate database using ORM (Hibernate/JPA) and perform CRUD operations.	LO 4
8	**Kotlin Backend Development**	Implement user authentication using JWT/OAuth2 and test with Postman.	LO 4
9	**Kotlin Backend Development**	Build microservice architecture (two services communicating via REST).	LO 4
10	**Kotlin Backend Development**	Deploy Spring Boot app to a cloud platform (Render/Heroku) or local server.	LO 4
11	**Flutter Foundations & Dart Programming**	Create a Flutter "Hello App" and explore basic Widgets and Layouts.	LO 5
12	**Flutter Foundations & Dart Programming**	Implement State Management (Provider/ScopedModel) for dynamic data display.	LO 5
13	**Advanced UI & Firebase Integration**	Integrate REST API from Kotlin backend into Flutter app using HTTP package.	LO 6
14	**Advanced UI & Firebase Integration**	Implement Firebase Authentication (Signup/Login) and Firestore database connectivity.	LO 6
15	**Advanced UI & Firebase Integration**	Develop and deploy a complete Flutter app with backend API and Firebase storage (Final Project Demo).	LO 6

One sample project with experiment mapping

Phase	Technology
Front-End (React)	Admin dashboard for managing students and courses (Expt 2–3)
Backend (Kotlin + Spring Boot)	APIs for CRUD operations on students, attendance, and grades (Expt 6–9)
Database Integration	Persistent storage of academic records (Expt 7)
Authentication	Role-based access (admin, faculty, student) (Expt 8)
Mobile (Flutter)	Student app for viewing attendance, grades, and notices (Expt 11–15)
Final Demo	Fully functional academic management system across platforms (Expt 15)

Note: Implement any 10-12 experiments from the given list

Text Books

1. Learning React: Modern Patterns for Developing React Apps – Alex Banks & Eve Porcello, O'Reilly (2023).
2. Node.js Design Patterns – Mario Casciaro, Packt (2023).
3. Kotlin Programming: The Big Nerd Ranch Guide by Josh Skeen and David Greenhalgh.
4. Kotlin for Backend development: Build scalable REST APIs, Microservices and Web Applications with Ktor, Spring Boot and Coroutines, Stokes J. Harrett.
5. Kotlin In-Depth: A Guide to a Multipurpose Programming Language for Server-Side, Front-End, Android, and Multiplatform Mobile - Aleksei Sedunov, BPB Publications.
6. Simplifying Application Development with Kotlin Multiplatform Mobile - Robert Nagy Packt publication
7. Beginning App Development with Flutter Create Cross-Platform Mobile Apps — Rap Payne, Apress
8. Flutter Cookbook, by Simone Alessandria, Brian Kayfitz, O'Reilly

Reference Books

1. Full-Stack Web Development with Node, Express, and React – Adam Freeman, Apress (2022)
2. Head First Kotlin - Dawn Griffiths & David Griffiths, O'Reilly.
3. Kotlin Development Complete Guide create 45 android applications including Jetpack Composer – Henry Codwell.
4. Adam Bretz and Colin J. Ihrig, "Full Stack JavaScript Development with MEAN", SitePoint Pty. Ltd., 2015 Edition.
5. Android UI development with Jetpack Compose - Thomas Kunneth, Packt Publishing.
6. Scalable Android applications in Kotlin.
7. E-Commerce, Pondicherry university

Online Resources:

1. https://www.udacity.com/course/developing-android-apps-with-kotlin--ud9012
2. https://www.coursera.org/learn/advanced-programming-in-kotlin
3. https://www.coursera.org/learn/advance-kotlin-for-multi-platform-development
4. https://www.pluralsight.com/paths/android-development-with-kotlin-fundamentals
5. https://github.com/androiddevnotes/awesome-android-learning-resources
6. https://www.udemy.com/course/android-app-development-with-kotlin-beginner-to-advanced
7. https://www.codecademy.com/learn/learn-kotlin
8. https://www.codingblocks.com/android-development-using-kotlin.html

Assignment 1: MAD (Any one)

1. To Study basics of ReactJS and React Native language and design/develop basic Web App
2. To include Files and JSON data in App
3. To build interactive App by including Flutter Gestures and Animations

Assignment 2: Kotlin Advanced Concepts (Any one)

1. To study the requirement for progressive web application for Ecommerce using the concept of service worker, Webapp Manifest and framework tools
2. To Design a wireframe for simple PWA for E-commerce website
3. Case study for successful real life implementation of PWA.

Assessment :

Term Work: Term Work shall consist of at least 10 to 12 practical based on the above list. Also, Term work Journal must include at least 2 assignments.

Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2165121	**User Experience Design with VR Lab**	-	2	-	-	1	-	1

Course Code	Course Name	Examination Scheme							
		Theory Marks				Term Work	Practical/ Oral	Total
		Internal assessment			End Sem. Exam			
		IAT-I	IAT-II	IAT-I + IAT-II				
2165121	**User Experience Design with VR Lab**	--	--	--	--	25	--	25

Lab Objectives

1. Understand how to design user interfaces that are effective, easy to use, and meet user needs.
2. Learn different methods for creating prototypes and testing user experience.
3. Apply the principles of good UI and UX design in real projects.
4. Install and set up Unity software and learn how VR devices work.
5. Create and explore simple VR scenes using Unity.
6. Perform usability and experience evaluation of UI and VR prototypes

Lab Outcomes

Learners will be able to :

1. Use suitable tools and methods to design and present user interface models.
2. Build a high-fidelity prototype that represents a complete design solution.
3. Evaluate user experience using standard testing techniques.
4. Set up a VR development environment and use VR devices like HTC Vive, Google Cardboard, or Samsung Gear VR.
5. Develop a basic VR scene and add interactive objects.
6. Conduct structured usability evaluations of UI and VR applications using various methods.

Suggested List of Experiments

Sr No	Suggested List of Experiments	Hrs
01	Project Proposal and Requirement Gathering (Choose the project). Briefly state the problem(s) that the project will seek to solve. Take the user's point of view. Consider what the user's goals are, and what obstacles lie in the way.	02
02	Create 2 Personas on the project topic selected.	02
03	Creation of Scenario: Write a scenario that involves all of the tasks identified for the chosen project.	02
04	Creating a Wireframe on selected problem statements.	02
05	Low Fidelity prototype using Figma tool.	02

06	High Fidelity prototype using Figma tool.	02
07	Usability Evaluation of the Design Testing of User Interface from Third Party(Test scripts)	02
08	Perform Rapid Evaluation on prototype design.	02
09	Perform Heuristic evaluation technique on the prototype design.	02
10	Emotional Design Evaluation of a user interface design by analyzing user reactions and emotional responses at various interaction points, in order to identify design elements that evoke positive, negative, or neutral feelings.	02
11	Installation of Unity and Visual Studio, setting up Unity for VR Development, understanding documentation of the same.	02
12	Develop a scene in Unity that includes: i. a cube, plane and sphere, apply transformations on the 3 game objects. ii. add a video and audio source	02
13	Understand basic VR environment design using Unity's *Play Mode* (desktop simulation).	02
14	Prototype VR Interaction using Mouse and Keyboard	02
15	Evaluate VR Usability using Video/Screen-based Heuristic Analysis	02

Useful Links	
1	https://www.coursera.org/professional-certificates/google-ux-design
2	https://nptel.ac.in/courses/124107008
3	https://www.coursera.org/learn/develop-augmented-virtual-mixed-extended-reality-applicatio ns-webxr-unity-unreal
4	https://tih.iitr.ac.in/AR-VR.html

Sr No	Suggested List of Assignments / Tutorials	Hrs
	Assignment 1	
01	While creating your project proposal, how did you identify the **main user problem** and define measurable **user goals**? Give an example from your project.	2 hrs
02	How did the **personas** you created influence your design decisions and the features included in your prototype?	
03	When designing your **wireframe**, what layout or navigation issues did you face, and how did you resolve them?	
	Assignment 2	
01	Describe the process you followed to create your **low-fidelity prototype in Figma**. How did it help before moving to the high-fidelity version?	2 hrs
02	During **usability testing**, what kind of feedback did you receive from users, and what changes did you make based on that feedback?	
03	In your **Unity VR project**, explain how you applied transformations (position, rotation, scale) to objects and added multimedia components like audio or video.	

Assessment:

Term Work: Term Work shall consist of at least 10 to 12 practical's based on the above list.

Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2165122	Computer Network Lab		2	-		1	-	1

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Practical/ Oral	Total
		Internal assessment			End Sem. Exam			
		IAT-I	IAT-II	IAT-I + IAT-II				
2165122	Computer Network	--	--	--	--	25	--	25

Lab Objectives:

1. To introduce students to networking components, commands, cables, and connectors essential for building and troubleshooting networks.
2. To enable students to design and simulate various network topologies and test their connectivity using modern simulation tools.
3. To develop understanding of data communication protocols and techniques such as sliding window and error detection/correction methods.
4. To provide hands-on experience in configuring routing protocols and core network services like FTP and DNS.
5. To enhance students' skills in network analysis, traffic monitoring, and performance evaluation using tools like Wireshark.
6. To acquaint students with advanced networking concepts such as VLANs, NAT, and Software Defined Networking (SDN).

Lab Outcomes:

1. Identify and use different network cables, connectors, and commands for establishing and troubleshooting network connections.
2. Design and simulate wired LANs and complex topologies, performing connectivity tests and file sharing between devices.
3. Implement and analyze data transmission techniques including sliding window protocols and error detection/correction algorithms.
4. Configure and verify static and dynamic routing protocols (RIP, OSPF) and network services like FTP and DNS.
5. Capture, analyze, and interpret network traffic using Wireshark to evaluate protocol behavior and network performance.
6. Design and simulate VLANs, NAT, and SDN topologies to understand modern network architectures and their applications.

List of Experiments.

Sr No	List of Experiments	LO
01	To study different networking commands and network cables & connectors.	LO1
02	Simulate a wired LAN using networking devices and test connectivity using PING utility.	LO1
03	Create different network topologies using any network simulator (e.g. Cisco Packet Tracer, GNS3). For each topology perform connectivity tests such as ping, file sharing between devices.	LO2
04	Simulate sliding window protocols using any packet tracer tool (any one from Stop and Wait, Go-Back-N, Selective Repeat)	LO3

05	Implementation of error detection and correction techniques (any one from CRC, Hamming Code, Checksum) using any programming language.	LO3
06	To analyze network traffic using the Wireshark tool. Perform following activities: • Packet capture and filtering • Analyze ethernet frames • IP packet analysis • ICMP analysis (PING test) • HTTP/HTTPs request analysis • Analyze header of TCP /UDP dump..	LO2, LO4, LO5
07	To configure and verify static routing and dynamic routing protocols (RIP and OSPF) in a simulated network environment using Cisco Packet Tracer.	LO4
08	To implement Socket programming using any programming language.	LO4, LO5
09	To configure FTP server and transfer files to demonstrate the working of the same.	LO5
10	To configure the DNS server and to demonstrate the working of the same.	LO5
11	To design and simulate VLANs on switch/router using any network simulator (e.g. Cisco Packet Tracer, GNS3)	LO6
12	To design and simulate NAT on router using any network simulator (e.g. Cisco Packet Tracer, GNS3)	LO6
13	Implementation of small SDN based topology using Mininet.	LO6

Online Resources:

Sr. No.	Website Name
1.	NPTEL Swayam Course: https://onlinecourses.nptel.ac.in/noc22_cs19/
2.	CCNA: Introduction to Networks https://www.netacad.com/courses/ccna-introduction-networks?courseLang=en-US

Sr No	List of Assignments / Tutorials	Hrs
01	Assignment covers the topics from the first three units.	CO1,CO2,CO3
02	Assignment covers the topics from the last three units.	CO4,CO5,CO6

Assessment :
Term Work: Term Work shall consist of at least 10 to 12 practicals based on the above list. Also, Term work Journal must include at least 2 assignments.
Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)

IKS

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
IKS501	**Indian Knowledge System**	--	2*+2	--	--	2	-	2

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		Test 1	Test 2	Total					
IKS501	**Indian Knowledge System**	--	--	--	--	--	50	--	50

*** Two hours of practical class to be conducted for full class as demo/discussion.**

Rationale:

The proposed IKS modules contextualize core engineering principles across Computer, IT, EXTC, Electronics, and Electrical engineering—within scientifically analyzable Indian knowledge traditions. By mapping indigenous practices to modern technical frameworks, students critically examine foundational engineering concepts through interdisciplinary and systems-based perspectives. This approach strengthens conceptual understanding, promotes analytical reinterpretation of traditional knowledge using contemporary tools, and aligns engineering education with culturally rooted yet technically rigorous inquiry.

Course Objectives:

1. Provide foundational knowledge of Indian acoustic practices.
2. To Understand Ancient Indian use of renewable energy.
3. Introduce Sanskrit grammar in simple language, with special attention to its rule-based structure.
4. Learners should be able to explain key contributions of ancient Indian mathematics in simple language.
5. Understand how Indian traditions of logic, language, knowledge organization, design thinking, governance, sustainability, and wellbeing can support modern IT solutions.
6. Understand the basic structure of Sanskrit grammar and relate selected grammatical ideas to computational modules used in information technology.

Course Outcomes:

On successful completion of the course learner will be able to:

1. Analyze traditional Indian acoustic practices and interpret their relevance to frequency analysis.
2. Apply the traditional knowledge to modern energy solutions and sustainable engineering.
3. Apply Indian Knowledge Systems in linguistics, Sanskrit-based NLP, traditional number systems, measurement practices, and ancient rainfall prediction techniques using calendars, nakshatras, and natural indicators.
4. Apply basic modular arithmetic, pattern recognition, and place-value ideas to simple IT examples such as coding, data representation, and checksums.
5. Apply basic concepts of Indian Knowledge Systems such as logic, language, knowledge organization, ethics, sustainability, and responsible decision-making to design simple technology-based solutions like apps, chatbots, databases, or awareness tools.
6. Apply basic Sanskrit grammar concepts and Pāṇinian rule-based principles to understand formal grammar, algorithms, and simple computational language processing applications in IT.

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping

I	**Acoustic Science in Vedic Chanting and Indian Music**	I.1 Concept of Nāda (Sound) and Shruti • Physical interpretation: frequency, pitch, harmonic content • Shruti system (22 microtones) and frequency discrimination I.2 Resonance and Architectural Acoustics • Temple architecture and echo control • Natural acoustic amplification • Relation to reverberation time and standing waves I.3 Structured Sound Encoding in Vedic Recitation • Pada, Krama, Ghana patha recitation methods • Built-in redundancy → error detection and correction • Parallel to digital communication redundancy schemes **Self-Study:** Study of sounds and analysis in time and frequency domains.	**10**	**CO1**
II	**Tradition**	1. Introduction to IKS in renewable energy 2. Ancient Indian use of renewable energy a) Solar Energy: Solar Architecture, Solar Heating Systems b) Wind Energy: Ancient Windmills, Wind-Powered Water Lifting c) Water (Hydro) Energy: Ancient Watermills, Irrigation Systems d) Bioenergy: Biomass Fuel, Bioenergy Production 3. Integration of ancient Traditional knowledge of renewable energy with Modern Technologies. **Students shall study the following to understand the use of ancient technologies in Renewable Energy:** • Gujarat Windmills • Watermills in the Indus Valley • Konark Sun Temple	**10**	**CO2**
III	**Linguistics, Number Systems, and rainfall Prediction**	Linguistics: • Components of language • Panini's work on Sanskrit grammar • Role of Sanskrit in Natural Language Processing Number System and Units of Measurement • Number system in India- Historical Evidence • Decimal System • Unique Approaches to represent Numbers-Katapayadi System • Pingala and the Binary System Ancient Indian techniques for predicting Rainfall • Introduction • Elements of the Indian Calendar • Panchang and Nakshtras • Non-bio indicators and Bio indicators **Self-Learning:** • An Ecosystem for Sanskrit Language Processing • Salient Features of the Indian Numeral System • Salient Features of the Indian Numeral System	**10**	**CO3**

		• Measurement for Time Distance and Weight • Measurement of Time- An Illustration from Purana		
IV	**Mathematics Foundations in Ancient India and Its Relevance to Information Technology**	4.1 Historical context and number ideas: Indian knowledge systems, the development of numeration, zero as placeholder and number, decimal place value. 4.2 Major contributors and procedural mathematics: Aryabhata, Brahmagupta, Bhaskara II, and examples of rule-based or sutra-like calculation methods in plain language. 4.3 Algorithmic thinking and computation: Historical mathematical procedures connect to algorithms, flowcharts, decomposition, and computational logic. 4.4 Modular arithmetic, binary-style ideas, and data representation: Clock arithmetic, remainders, cyclic reasoning, two-state representation, bits, bytes, and simple binary conversion. 4.5 Relevance to IT and synthesis: Coding, checksums/parity, simple cryptography idea, data representation, and how ancient mathematical thinking supports modern IT problem-solving. **Self-Learning Topics and Activities:** • Zero and place value in daily life • Algorithmic thinking • Modular arithmetic • Binary representation	**10**	**CO4**
V	**Indian Logic and its applications in Information Technology**	5.1 Introduction to IKS and its relevance to contemporary technology. 5.2 Indian logic, language, and computation with simple examples. 5.3 Knowledge organization, data, and decision-making. 5.4 Ethics, society, sustainability, and responsible digital systems. 5.5 Mini activity where students design a small app, chatbot, database, or awareness tool using one IKS principle. **Self-Learning Topics and Activities:** • Basic ideas of Nyaya logic: observation, inference, debate, and evidence. • Introduction to Paninian grammar and why structured language matters in NLP. • Traditional knowledge classification and its connection with databases and knowledge graphs. • Technology for Indian languages, heritage preservation, digital museums, and public services. • Design thinking activity: identify a local problem and propose an IKS-inspired IT solution.	**10**	**CO5**
VI	**Sanskrit Grammar and Computational Modules**	6.1 Introduction to Indian Knowledge Systems, Sanskrit as structured language, and relevance to IT. 6.2 Basic Sanskrit grammar concepts: sounds, words, roots, suffixes, sandhi, and sentence structure. 6.3 Pāṇinian grammar as a rule-based system; relation with algorithms and formal grammar.	**10**	**CO6**

		6.4 Computational modules: tokenization, morphology, parsing, language rules, and NLP examples. 6.5 Demonstration of Sanskrit digital tools, reflection activity, and discussion on interdisciplinary applications. **Self-Learning Topics and Activities:** • Basic Sanskrit alphabet and sound classification. • Simple sandhi examples and word joining rules. • Introduction to Pāṇini and the Aṣṭādhyāyī. • Dhātu, pratyaya, pada, and kāraka in simple terms. • Basics of natural language processing: tokenization, morphology, and parsing. • Exploration of Sanskrit digital dictionaries and Sanskrit reader tools. • Comparison of grammar rules and computer program rules.		

Text and Reference Books or Articles:

1. Acoustics of Chants, Conch-Shells, Bells & Gongs in Worship Spaces. Available: https://www.academia.edu/38208102/Acoustics_of_Chants_pdf?utm_source=chatgpt.com
2. V. R. S. Nalluri, V. J. K. S. Sonti, and G. Sundari, "Analysis of frequency dependent Vedic chanting and its influence on neural activity of humans," International Journal of Reconfigurable and Embedded Systems, vol. 12, no. 2, pp. 230–239, Jul. 2023, doi: 10.11591/ijres.v12.i2.pp230-239.
3. The Garland Encyclopedia of World Music: South Asia. Available: https://api.pageplace.de/preview/DT0400.9781351544399_A30892028/preview-9781351544399_A30892028.pdf?utm_source=chatgpt.com
4. J. Pattanayak, Y. Belur, and J. Ilavarasu, "Acoustic and psychological effects of Omkar chanting in novice Indian adults: A quasi-experimental study," Journal of Clinical and Diagnostic Research, 2025, doi: 10.7860/JCDR/2025/77365.21111. Available: https://www.jcdr.net/articles/PDF/21111/77365_CE%5BRa1%5D_F%28IS%29_QC%28PS_SS%29_P F1%28AB_IS%29_redo_PFA%28IS%29_PB%28AB_IS%29_PN%28IS%29.pdf?utm_source=ch atgpt. com
5. "Indian Knowledge System: Integrating Heritage with Engineering" by Gagan Bansa, Deep Science Publishing; ISBN: 978-93-49307-77-3 E-ISBN: 978-93-49307-29-2;
6. The textbook of Indian Knowledge Systems, University of Mumbai, ISBN: 978-93-5915-553-1
7. Introduction to Indian Knowledge System: Concepts and Applications, Mahadevan, B., Bhat, Vinayak Rajat, Nagendra Pavana R.N. by PHI Learning Pvt Ltd. Delhi, ISBN: 978-93-91818-20-3
8. Echoes of the Monsoon: Ancient Indian techniques for predicting Rainfall, Janak Joshi, Kandarp
9. Parmar, Lakhan Jain, Sharmistha Bhowmik and Bindu Bhatt, DOI: https://www.doi.org/10.22271/27067483.2025.v7.i7a.386.
10. Kim Plofker, Mathematics in India: A clear historical overview of Indian mathematical developments, suitable for faculty use and advanced self-learners.
11. George Gheverghese Joseph, The Crest of the Peacock: An accessible introduction to non-European mathematical traditions, including India.
12. Charles Petzold, Code: The Hidden Language of Computer Hardware and Software: A beginner-friendly bridge from numbers and symbols to data representation and computing.
13. Simon Singh, The Code Book: Useful for simple explanations of ciphers, encryption, and the mathematics behind communication security.
14. T. A. Sarasvati Amma, Geometry in Ancient and Medieval India: A more specialized reference for deeper exploration of traditional mathematical methods.
15. Mahadevan, B., Bhat, V. R., and Nagendra Pavana R. N. Introduction to Indian Knowledge System: Concepts and Applications. PHI Learning.

16. Kapur, K., and Singh, A. K. (Eds.). Indian Knowledge Systems, Vol. 1. Indian Institute of Advanced Study, Shimla.
17. Bora, M. C. Bharatiya Knowledge Systems. Khanna Publishing House.
18. Rao, S. S. Engineering and Technology in Ancient India. Indian National Science Academy / related academic editions.
19. Selected open articles or chapters from IKS Division / AICTE / Ministry of Education resources may be used as supplementary reading.
20. S. D. Joshi and J. A. F. Roodbergen, The Aṣṭādhyāyī of Pāṇini.
21. V. N. Jha, The Philosophy of Sanskrit Grammar.
22. Peter M. Scharf, Sanskrit Syntax and Discourse Structures.
23. Daniel Jurafsky and James H. Martin, Speech and Language Processing — for introductory NLP concepts.
24. Rajeev Sangal, Akshar Bharati, and Vineet Chaitanya, Natural Language Processing: A Paninian Perspective.

Other Online Resources:

Sr. No.	Website Name
1.	NPTEL course page: For a structured academic overview of mathematics in India. https://nptel.ac.in/courses/111101080
2.	NPTEL YouTube playlist: For lecture-based learning that can support flipped or self-paced study. https://www.youtube.com/playlist?list=PLbMVogVj5nJThf31TNSQzuN7zqxe7HdRN
3.	IKS India: For broader Indian Knowledge Systems context and additional learning resources. https://www.nist.gov/cryptography
4.	Responsible AI and India-context resources: https://indiaai.gov.in/
5.	IKS Division, Ministry of Education / AICTE: https://iksindia.org/
6.	NPTEL: Introduction to Pāṇinian Grammar URL: https://onlinecourses.nptel.ac.in/noc24_hs44/preview
7.	Sanskrit Heritage Reader URL: https://sanskrit.inria.fr/

List of Practical and List of Assignments:

Sr No	List of Experiments	Hrs
01	Record a temple bell sound, conch sound, or classical “Sa” note and analyze it by plotting its time-domain waveform and frequency spectrum, identifying the fundamental frequency and harmonics, and comparing the observed harmonic structure with theoretical frequency ratios of Just Intonation and Equal Temperament.	04
02	Observe the movement of sunlight and shadow around a building, monument, or open courtyard at different times of the day, record the shadow directions and lengths, and analyze how solar position can guide orientation, time estimation, and passive architectural planning in the context of traditional Indian solar design.	04
03	Examine images, plans, or a simple model of the Konark Sun Temple or any sun-oriented structure, identify how geometry, orientation, and sunlight interaction are used in the design, and prepare a visual explanation of how ancient solar architecture supports modern sustainable building concepts.	04
04	Construct a small paper or cardboard windmill, test its rotation under natural wind or fan airflow at different speeds, and document how blade movement demonstrates the basic principle of converting wind energy into mechanical motion, linking it to traditional wind-powered systems and modern turbines.	04
05	Build a simple working model using cups, straws, thread, or cardboard to simulate water lifting or channel-based irrigation, observe how water can be moved with minimal energy input, and relate the model to ancient Indian irrigation systems and watermill-based practices.	04

06	Survey homes or nearby community spaces to identify traditional biomass fuels such as wood, agricultural residue, or cow-dung cakes, record their current use, benefits, and limitations, and compare the observed practices with modern ideas of bioenergy and sustainable fuel management..	04
07	Pronounce and classify selected Sanskrit sounds into vowels and consonants using a simple sound chart, group them according to articulation patterns, and analyze how this systematic sound organization supports pronunciation training, language learning, and speech-processing applications.	04
08	Create all possible combinations of short and long syllabic patterns of a fixed length, tabulate them systematically, and compare the resulting patterns with binary 0/1 combinations to understand how ancient combinatorial thinking connects with digital representation.	04
09	Solve real-life problems based on clocks, calendars, weekdays, or repeated events using modular arithmetic, record the remainder-based reasoning used in each case, and explain how cyclic patterns support time management, scheduling, and introductory cryptographic logic.	04
10	Convert selected decimal numbers into binary form, arrange them in bit patterns, append parity bits for even or odd parity, and interpret how binary representation and simple error-detection methods are used in digital systems for reliable data storage and transmission.	04
11	Use a basic Sanskrit dictionary, transliteration tool, or reader tool to input selected words, observe the output such as root, meaning, or grammatical analysis, and describe how such tools support language technology and digital preservation.	04
12	Select a real-life problem such as fake-news verification, waste segregation, or campus awareness, identify observation, evidence, inference, and conclusion using the basic framework of Indian logic, and represent the reasoning process as a decision tree, chatbot flow, or awareness tool concept.	04
13	Take a simple historical or rule-based calculation procedure associated with Indian mathematics, rewrite it as a sequence of input-process-output steps, and convert it into a flowchart to demonstrate the connection between procedural mathematics and modern algorithmic thinking.	04
14	Represent selected large and small numbers using a place-value table, deliberately insert or remove zero in different positions, and analyze how zero functions both as a number and as a placeholder, showing its importance in arithmetic, accounting, and digital data representation.	04

Sr No	**List of Assignments**	**Hrs**
01	Study the concepts of Nāda and Shruti, temple acoustics, and Vedic recitation methods, and briefly explain their relation to modern concepts of frequency, resonance, reverberation, and redundancy/error correction in digital communication systems. Also write a short note on how Indian Knowledge Systems demonstrate scientific understanding of sound and communication.	01
02	Study the ideas of solar architecture in traditional Indian structures such as the Konark Sun Temple or courtyard-based houses, and explain how sunlight, orientation, shading, and ventilation were used for thermal comfort and energy efficiency. Also write a short note on how these traditional design ideas are relevant to modern sustainable architecture and climate-responsive planning..	01
03	Study one traditional Indian renewable energy practice such as wind-powered water lifting, watermills, solar heating, or biomass use, and compare it with one modern renewable energy technology. Explain how traditional knowledge systems supported sustainable living and discuss how these ideas can inspire present-day energy conservation and green technology.	01
04	Study the basic ideas of Pāṇinian grammar as a structured rule-based system, and explain	01

	how concepts such as sounds, roots, suffixes, and word formation relate to modern ideas in natural language processing, machine translation, and language analysis. Also write a short note on why Sanskrit grammar is often considered relevant to computational linguistics.	
05	Study the historical development of zero as both a number and a placeholder, and explain how the decimal place-value system simplified arithmetic and record keeping. Also write a short note on how zero and place value continue to be essential in digital computation, data storage, and computer-based numerical processing.	01
06	Study the basic ideas of Indian logic, especially observation, inference, debate, and evidence in the Nyāya tradition, and explain how these ideas support careful reasoning and decision-making. Also write a short note on how such logical methods can be applied in present-day contexts such as data analysis, fake-news verification, ethical decision-making, and responsible digital systems.	01
07	Identify one idea from Indian Knowledge Systems in the syllabus, such as renewable energy, Indian numeration, Indian logic, or Sanskrit grammar, and propose a simple real-life solution such as an app, chatbot, awareness campaign, digital archive, poster, database, or educational tool based on that idea. Explain the problem being addressed, the IKS principle used, and how the proposed solution can be useful in present-day society.	01
08	Study the basic Sanskrit grammar concepts of sounds, words, roots, suffixes, sandhi, and sentence structure, and explain how grammar rules work like structured instructions. Also compare grammar rules with computer program rules, and briefly discuss how tokenization, morphology, and parsing in computational systems resemble traditional rule-based language analysis.	01

Assessment:

Term Work: Term work shall consist of at least 10-12 practical's based on above list. Also Term Work Journal must include at least 4 assignments

Term Work Marks: 50 Marks (Total marks) = 25 Marks (Experiment) + 20 Marks (Assignments) + 5 Marks (Attendance)

Sem VI

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2166111	Machine Learning and Swarm Intelligence	3	-	-	3	-	-	3

Course Code	Course Name	Theory					Term Work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-I	IAT-I	IAT-I + IAT-II					
2166111	Machine Learning and Swarm Intelligence	20	20	40	60	2	--	--	100

Rationale :
This course provides a strong foundation in Machine Learning concepts and techniques, covering both traditional algorithms and modern bio-inspired optimization methods. It enables students to design, implement, and analyze intelligent models for solving real-world problems efficiently. The inclusion of swarm intelligence introduces learners to emerging approaches in adaptive and scalable computation.

Course Objectives:
1. To understand the fundamental concepts, types, and workflow of Machine Learning.
2. To learn and apply key algorithms for classification, clustering, and association rule mining.
3. To study and implement linear models for regression and classification problems.
4. To explore feature selection and dimensionality reduction techniques for handling high-dimensional data.
5. To understand the concepts, characteristics, and mechanisms of swarm intelligence and bio-inspired algorithms.
6. To analyze the use of swarm intelligence algorithms in optimization and Machine Learning applications.

Course Outcomes: On successful completion, of course, learner/student will be able to:
1. Explain the fundamental concepts of Machine Learning, its applications, and performance evaluation methods.
2. Apply classification, clustering, and association rule algorithms to analyze data patterns.
3. Implement linear models such as regression, logistic regression, and SVM for prediction and classification tasks.
4. Use dimensionality reduction and feature selection techniques to simplify high-dimensional data.
5. Describe the principles and working of swarm intelligence and bio-inspired optimization algorithms.
6. Analyze the performance and applications of swarm intelligence-based algorithms in solving real-world optimization and machine learning problems.
7.

DETAILED SYLLABUS:

Sr. No	Name of Module	Detailed Content	Hours	CO Mapping
0		Linear Algebra, Vectors, Probability and statistics, Programming Concepts		
I	**Introduction to Machine Learning**	1.1 Introduction to Machine Learning, Issues in Machine Learning, Application of Machine Learning, Steps of developing a Machine Learning Application. 1.2 Supervised and Unsupervised Learning: Concepts of Classification, Clustering and prediction 1.3 Training, Testing and validation dataset, cross validation, over fitting and under fitting of model Performance Measures: Measuring Quality of model- Confusion Matrix, Accuracy, Recall, Precision, Specificity, F1 Score, RMSE. 1.4 Symmetric Positive Definite Matrices, Determinant, Trace, Eigenvalues and vectors, Orthogonal Projections, Diagonalization, SVD and its applications **Self-Learning :** Transfer Learning and Self-Supervised Learning	**8**	**CO1**
II	**Classification Clustering and Association Rules**	2.1 Classification Algorithm: Random Forest 2.2 Clustering Algorithms - K means, Hierarchical 2.3 Rule based Algorithm- Apriori **Self-Learning :** Semi-Supervised and Active Learning Techniques	**6**	**CO2**
III	**Linear Models**	3.1 The least-squares method, Multivariate Linear Regression, Regularized Regression, Using Least-Squares Regression for classification 3.2 Support Vector Machines Linear SVM Maximum margin classifier Support vectors Hard margin vs. soft margin Non-Linear SVM Kernel trick Common kernels: polynomial, RBF, sigmoid 3.3 Logistic regression **Self-Learning :** Support Vector Regression and Multi-class SVMs	**6**	**CO3**
IV	**Dimensionality Reduction**	4.1 Curse of Dimensionality. 4.2 Feature Selection (Filter Method, Embedded Method, Wrapper Method) and Feature Extraction 4.3 Dimensionality Reduction Techniques, Principal Component Analysis. **Self-Learning :** t-SNE and UMAP for Data Visualization , - Feature Importance and Selection Metrics,- Relationship between Dimensionality and Model Accuracy	**5**	**CO4**
V	**Swarm Intelligence and Bio-Inspired Computation**	5.1 Introduction to Swarm Intelligence and Bio-inspired computation, 5.2 Current Issues in Bio-Inspired Computing, 5.3 Classifications and Terminology 5.3 Tuning of Algorithm-Dependent Parameters Necessity for Large-Scale and Real-World Applications Choice of Algorithms , Search for	**8**	**CO5**

		the Magic Formulas for Optimization, sense of an Algorithm, Algorithms and Self-Organization , Links Between Algorithms and Self-Organization,The Magic Formulas, Characteristics of Metaheuristics, Intensification and Diversification, Randomization Techniques , 5.4 Swarm-Intelligence-Based Algorithms , Ant Algorithms , Bee Algorithms , Bat Algorithm , Particle Swarm Optimization , Firefly Algorithm , Cuckoo Search , Flower Pollination Algorithm **Self-Learning :** Hybrid Optimization (PSO + GA) and Reinforcement Learning-based Optimization		
VI	**Analysis of Swarm Intelligence Based Algorithms**	6.1 Introduction , 6.2 Optimization Problems , 6.3 Applications of Swarm Intelligence in Machine Learning, swarm Intelligence Based Optimization Algorithms: Ant Colony Optimization, Particle Swarm Optimizer , ABC Algorithm , Firefly Algorithm , Cuckoo Search Algorithm Bat Algorithm **Self-Learning :** Quantum-Inspired and Multi-Objective Swarm Optimization	**6**	**CO6**

Text Books:

1. Ethem Alpaydın, ―Introduction to Machine Learning‖, MIT Press McGraw-Hill Higher Education
2. Marc Peter Deisenroth, Aldo Faisal, Cheng Soon Ong, ―Mathematics for machine learning‖,
3. James Kennedy & Russell Eberhart, Swarm Intelligence, Morgan Kaufmann
4. Handbook of Swarm Intelligence: Concepts, Principles and Applications (Panigrahi, Shi & Lim, Springer, 2011)

References:

a. Tom M. Mitchell, ―Machine Learning‖, McGraw Hill
b. Particle Swarm Optimization by Maurice Clerc (2006)
c. Swarm Intelligence Optimization: Algorithms and Applications- Abhishek Kumar, Pramod Singh Rathore, Vicente Garcia Diaz & Rashmi Agrawal, Published by Wiley
5. Shai Shalev-Shwartz, Shai Ben-David, ―Understanding Machine Learning‖, Cambridge University Press

Online References:

Sr. No.	Website Name
1.	https://www.coursera.org/learn/machine-learning
2.	https://onlinecourses.nptel.ac.in/noc25_me22/preview
3.	https://www.coursera.org/learn/evolutionary-computation-and-its-engineering-applications

Assessment:

Internal Assessment (IA) for 20 marks:

- IA will consist of Two Compulsory Internal Assessment Tests. Each test will be 20 marks. Approximately 40% to 50% of syllabus content must be covered in First IA Test and remaining 40% to 50% of syllabus content must be covered in Second IA Test.

Question paper format:

- Question Paper will comprise of a total of **six questions each carrying 15 marks. Q.1** will be **compulsory** and should **cover maximum contents of the syllabus.**
- **Remaining questions** will be **mixed in nature** (part (a) and part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module randomly selected from all the modules).
- A total of Four questions need to be answered.

University of Mumbai

Course Code: ________________and Course Name:

Date of Exam: Duration: 2 Hours Max. Marks: 60

NB: (1) Question 1 is compulsory. (2) Solve any three questions out of five.		
Q.1	**Solve any three questions out of five: (05 marks each)**	**15**
a)		05
b)		05
c)		05
d)		05
e)		05
Q.2,3, 4,5,6		**15**
a)	Part(a) and Part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module.	07or 08
b)		08 or 07

Note for QP Setters:

1. All COs should be mapped on the basis of the syllabus.
2. Module weightage as per contact hours mentioned in the syllabus.
3. Question 1 is compulsory, and it should include sub-questions covering all modules, with each sub-question carrying five marks.
4. The question paper should contain six questions, each of 15 marks.

****

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Theory	Pract.	Total
2166112	Data Analytics and Visualization	3	-	-	-	3	-	3

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-I	IAT-II	IAT-I + IAT-I					
2166112	Data Analytics and Visualization	20	20	40	60	2	--	--	100

Rationale:

Data visualization, prediction, and analytics play a crucial role in enabling engineering students to interpret complex technical data with clarity and precision. Visual analytics tools such as graphs, dashboards, and statistical charts help students quickly identify patterns, anomalies, performance issues, and design optimization opportunities. By applying analytical and predictive techniques—such as regression, trend forecasting, and data modelling—students can make informed, data-driven decisions in experiments. These skills enhance problem-solving abilities, improve the quality of engineering solutions, and prepare students for industry roles where data-centric thinking, predictive analytics, and evidence-based decision-making are essential.

Course Objectives:

1. To introduce the concept of fundamentals of Data Analytics.
2. To preprocess data and build regression models.
3. To understand the fundamental concepts of Data exploration and anomaly detection.
4. To understand, apply time series techniques in real world scenarios.
5. To understand and apply the knowledge and skills to perform exploratory data analysis visualizations using Python.
6. To enable students to develop interactive visualizations and dashboards using R and modern Tools.

Course Outcomes: At the end of the course, the students should be able to

1. Understand the basics of data analytics.
2. Apply data pre-processing techniques and evaluate regression models using analytical tools to derive accurate and data-driven predictions.
3. Apply data exploration and data anomaly detection techniques to solve real world data analysis problems.
4. Apply time series modelling techniques for trend analysis, forecasting, and decision-making in real-world scenarios.
5. Apply exploratory data analysis techniques, and develop visualizations in R to support data-driven decision making.
6. Create, customize, integrate visualizations using Python and BI tools to design interactive dashboards.

DETAILED SYLLABUS:

Sr. No	Name of Module	Detailed Content	Hours	CO Mapping
0	**Prerequisite**	Data warehouse and mining and Python Programming		
I	**Introduction to Data Analytics**	**1. Background and Overview of Data Analytics:** Key Roles for a Successful Analytics, Data Analytics life cycle, Analytics value chain. **2.** Types of analytics, Roles in Industry,	3	CO1

		Business Analytics in Practice, Ethics, and Privacy. Data Discovery, Data Preparation, Common tools for data analytics. **Self-learning Topics:** Data sources, relational DB design, NoSQL.		
II	**Data Pre-processing and Regression Model Evaluation**	**1. Data Pre-processing:** Data Cleaning, Normalization and Standardization, Outlier Detection and Handling, PCA, Feature Creation and Selection, Encoding Categorical Variables, Correlation and Multicollinearity Analysis. **2.Regression Models:** The Regression Equation Fitted Values and Residuals, Least Squares Method, Simple vs. Multiple Linear Regression, Model Assessment, Cross-Validation Techniques, Model Selection, Prediction Using Regression Models. **3.Logistic Regression and Generalized Linear Models:** Logistic Response Function and Logit Transformation, Logistic Regression as a Generalised Linear Model (GLM), Interpreting Coefficients and Odds Ratios, Model Evaluation, Linear vs. Logistic Regression, Step wise regression, Advanced and Nonlinear Regression, Performance matrix. **Self-learning Topics:** Regularization techniques.	**8**	**CO2**
III	**Data Exploration and Anomaly Detection**	**1. Data Exploration:** Descriptive Statistics: Univariate and multivariate Exploration, Measure of Central Tendency, Measure of Spread, Symmetry, Skewness: Karl Pearson Coefficient of skewness. **2. Anomaly Detection**: Outliers, Causes of Outliers, Anomaly detection techniques, Outlier Detection using Statistics, Outlier Detection using Distance based method, Outlier detection using density-based methods. **Self-learning Topics:** Anomaly Detection in Real-World Datasets	**6**	**CO3**
IV	**Time Series Analysis**	**1. Overview of Time Series Analysis:** Time Series Analysis Steps, Time series Forecasting, Components of Time series, Box Jenkins Methodology and its applications. **2.Time Series Models :** Autocorrelation Function (ACF), Partial Autocorrelation Function (PACF), Autoregressive Models, Moving Average Model (ARMA), Autoregressive Integrated Moving Average Model (ARIMA),Exponential smoothing, Seasonal decomposition, Forecasting accuracy (MAE, RMSE), Vector Auto Regression (VAR) and Multivariate time series, Time series feature engineering, Case studies of time series on real datasets.	**8**	**CO4**
		Self-learning Topics: Time series clustering		

V	**Data Visualization with Python**	1. Introduction to Data Visualization: Importance and need for visualization in analytics, Types of data visualization: Univariate, Bivariate, and Multivariate.Visualization before analysis, Role of Exploratory Data Analysis (EDA) its applications. 2. Introduction to Python for Visualization: Overview of Python for data analytics and visualization, Essential Data Libraries: NumPy, Pandas, Matplotlib, Seaborn. 3. Advanced Visualization Libraries: Plotly, Bokeh, Altair, Plotting and customizing graphs using Python. **Self-learning Topics:** Text analytics.	**6**	**CO5**
VI	**Data Visualization with R, Tools and recent Applications**	1. Introduction to R for Visualization : Data Import and Export techniques in R, Attributes and Data Types in R, Descriptive Statistics using R, Matrix and Array Operations, Exploratory Data Analysis in R, Visualizing single variable and multiple variables, Building Barplots, Histograms, Box plots, Whisker plots, and Scatterplots. 2. Visualization Tools and Dashboards: Overview of modern visualization tools like Tableau, Power BI- Building blocks, components, applications. Designing real time dashboards and integrating multiple visualizations 3. Recent Applications: Marketing analytics, finance, operations, IoT / sensor data, web analytics, manufacturing analytics and any other real trend application in industry. **Self-learning Topics:** Social media analytics	**8**	**CO6**

Text Books:

1. Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data,EMC Education services Wiley Publication
2. Peter Bruce, Andrew Bruce & Peter Gedeck, Practical Statistics for Data Scientists 50+ Essential Concepts Using R and Python, O'Reilly Publications 2nd Edition.
3. Mastering Data Visualization with Tableau by Dr. Arpana Chaturvedi, Prof. Praveen Malik.
4. Gagan Gupta Raj, Naresh Kumar Nagwani, Data Analytics and Visualization,, PHI Learning.
5. R for Data Science: Import, Tidy, Transform, Visualize, and Model Data by Hadley Wickham and Garrett Grolemund
6. Interactive Data Visualization with Python: Present Your Data As an Effective and Compelling Story, 2nd Edition by Abha Belorkar, Sharath Chandra Guntuku, and Shubhangi Hora.

References:

1. David G. Kleinbaum & Mitchel Klein, Logistic Regression: A Self-Learning Text , Springer
2. *Dr. Anil Maheshwari* , *"Data Analytics Made Accessible"* (2nd Edition, 2020) , Mcgraw Hill Education.

3. Dr. Bharti Motwani, Data Analytics using Python, Wiley India Pvt. Ltd

Online References:

Sr. No.	Website Name
1.	https://www.coursera.org/learn/introduction-to-data-analytics?utm_source=chatgpt.com
2.	https://www.udemy.com/course/data-storytelling-mastering-data-visualization-techniques/?couponCode=PMNVD2025
3.	https://www.datacamp.com/category/data-visualization?page=1
4.	https://onlinecourses.swayam2.ac.in/imb25_mg218/preview
5.	https://www.udemy.com/course/data-analytics-masters-basics-to-advanced/?couponCode=PMNVD2025

Assessment:

Internal Assessment (IA) for 20 marks:

- IA will consist of Two Compulsory Internal Assessment Tests. Each test will be 20 marks. Approximately 40% to 50% of syllabus content must be covered in First IA Test and remaining 40% to 50% of syllabus content must be covered in Second IA Test.

Question paper format:

- Question Paper will comprise of a total of **six questions each carrying 15 marks. Q.1** will be **compulsory** and should **cover maximum contents of the syllabus.**
- **Remaining questions** will be **mixed in nature** (part (a) and part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module randomly selected from all the modules).
- A total of Four questions need to be answered.

University of Mumbai

Course Code: ________________and Course Name:

Date of Exam: Duration: 2 Hours Max. Marks: 60

NB:
(1) Question 1 is compulsory.
(2) Solve any three questions out of five.

Q.1	**Solve any three questions out of five: (05 marks each)**	**15**
a)		05
b)		05
c)		05
d)		05
e)		05
Q.2,3, 4,5,6		**15**
a)	Part (a) and Part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module.	07or 08
b)		08 or 07

Note for QP Setters:

1. All COs should be mapped on the basis of the syllabus.
2. Module weightage as per contact hours mentioned in the syllabus
3. Question 1 is compulsory, and it should include sub-questions covering all modules, with each sub-question carrying five marks.
4. The question paper should contain six questions, each of 15 marks.

****

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2166113	Internet of Things	3	-	-	3	-	-	3

Course Code	Course Name	Theory					Term work	Pract. / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-I	IAT-II	IAT-I + IAT-II					
2166113	Internet of Things	20	20	40	60	2	--	--	100

Prerequisite: Basics of Microcontrollers, computer networks

Course Objectives:

1. Understand the fundamental concepts, architecture, and components of the Internet of Things (IoT).
2. Explore various communication technologies, protocols, and design principles for connected IoT devices.
3. Analyze IoT network architectures and data management frameworks involving edge, fog, and cloud computing.
4. Apply data analytics concepts to IoT applications using cloud computing paradigms.
5. Evaluate IoT privacy issues, security threats, and implement suitable protection mechanisms.
6. Develop IoT-based solutions and platforms for real-world use cases through systematic design methodologies.

Course Outcomes:

1. Explain the fundamental concepts, architecture, and components of the Internet of Things, including sensors, actuators, and IoT development boards.Apply properties of Relation sets in real-life problem-solving domains.
2. Apply suitable communication technologies and web connectivity protocols for designing efficient IoT-enabled systems
3. Analyze IoT network architectures and data management frameworks involving edge, fog, and cloud computing.
4. Demonstrate the process of data acquisition, organization, and analytics using cloud-based IoT platforms.
5. Evaluate IoT security requirements, identify vulnerabilities, and apply appropriate layered security models.
6. Design and develop IoT-based applications and platforms for real-world domains such as smart homes, cities, and agriculture.

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
I	**Introduction to IoT**	**Introduction:** Definition and Characteristics of IoT **Physical Design of IoT:** Things in IoT, Overview of IoT protocols **Logical design of IoT:** Functional Blocks, Communication Models, Communication API's. **Smart Objects and IoT boards:** Sensors, Actuators and Smart Objects. Boards: Comparison of Arduino, Node MCU, ESP and Ri Pi.	6	CO1

		Self-learning Topics: Pin diagrams of various sensors and boards		
II	**Design Principles for Connected Devices and Web Connectivity**	**Connected Devices:** Communication Criteria: Range, Frequency, Power, Topology. **Communication Technologies:** Wireless: NFC, RFID, BT, Zigbee, WiFi. Wired:- UART, SPI,I2C . A Comparison. **Web Connectivity Protocols:** HTTP: Limitations in constrained environment, Need for protocols in constrained environment: COAP, MQTT, XMPP, AMQP, SOAP. **Long Range Connectivity:** LoRA, NBIoT **Communication API's:** REST and Websockets **Self-learning Topics:** Application Layer protocols: HTTP, FTP, Telnet	8	CO2
III	**IoT Network Architecture and Design**	**Drivers Behind New Network Architectures:** Scale, Security, Constrained Devices and Networks, Data, Legacy Device Support. **Architecture:** The IoT World Forum (IoTWF) Standardized Architecture: Layer 1-7, IT and OT Responsibilities in the IoT Reference Model, Additional IoT Reference Models. **A Simplified IoT Architecture:** The Core IoT Functional Stack:Layer 1, Layer 2 (only overview) Layer-3: Analytics Versus Control Applications, Data Versus Network Analytics Data Analytics Versus Business Benefits, Smart Services. **IoT Data Management and Compute Stack:** Fog Computing, Edge Computing, The Hierarchy of Edge, Fog, and Cloud **Self-learning Topics:** Brief review of applications of IoT: Connected Roadways, Connected Factory, Smart Connected Buildings , Smart Creatures etc.	8	C03
IV	**IoT Analytics for cloud**	**IoT Data Acquiring, Organizing:** Introduction, Data sources in IoT, data generation, Data acquisition, data validation, data categorization for storage, Data store management. **IoT Data Organization:** Database, Query Processing, SQL, NOSQL, ETL. **IoT Data Analytics Phases:** Descriptive Analytics, Predictive Analytics, Prescriptive Analytics. **Cloud computing paradigm for data collection Storage and Computing:** Introduction, Cloud computing paradigm, cloud deployment models. **Overview of cloud computing models:** IaaS, PaaS, SaaS. *Self-learning Topics: Cloud Services for IoT Analytics: IoT Integration and Automation using IFTTT, Blynk App*	7	CO4

V	**IoT Privacy, Security and Vulnerabilities**	Introduction: Vulnerabilities, Security Requirements and Threat Analysis. IoT Security: Tomography and Layered Attacker Model, Security Models, Profiles and Protocols. **Self-learning Topics:** Study of Security issues in IoT referring reputed papers.	4	CO5
VI	**IoT platforms design Methodology and Case Studies**	**General Design Methodology (10 Steps):** Based on case study of Home automation, detail study of steps 1-4 and Overview from 5-10 **IoT Case Study:** Transportation: Connected Car and its applications, Smart City: Smart Parking, Waste Management, Street Light Control, Connected ATM premises monitoring system, Agriculture: Smart Irrigation System, Environmental Monitoring System. **Self-learning Topics:** Design of software for IoT applications	6	CO6

Text Books:

1. Arsheep Bahga (Author), Vijay Madisetti, Internet Of Things: A Hands-On Approach Paperback, Universities Press, Reprint 2020.
2. Raj Kamal, Internet of Things, Architecture and Design Principles, McGraw Hill Education, Reprint 2018.
3. David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Robert Barton, Jerome Henry, IoT
4. Fundamentals Networking Technologies, Protocols, and Use Cases for the Internet of Things CISCO.
5. Andrew Minteer: Analytics for the Internet of Things (IoT) Intelligent Analytics for Your Intelligent Devices.

Reference Books:

1. Pethuru Raj, Anupama C. Raman, The Internet of Things: Enabling Technologies, Platforms, and Use Cases by , CRC press,
2. Giacomo Veneri , Antonio Capasso,‖ Hands-On Industrial Internet of Things: Create a powerful Industrial IoT infrastructure using Industry 4.0‖, Packt.
3. Perry Lea, Internet of Things for Architects: Architecting IoT solutions by implementing sensors, communication infrastructure, edge computing, analytics, and security, Packt Publications, Reprint 2018.
4. Amita Kapoor, ―Hands on Artificial intelligence for IoT‖, 1st Edition, Packt Publishing, 2019.
5. Sheng-Lung Peng, Souvik Pal, Lianfen Huang Editors: Principles of Internet of Things (IoT)Ecosystem:Insight Paradigm, Springer

Online References:

1. NPTEL: Sudip Misra, IIT Khargpur, Introduction to IoT: https://nptel.ac.in/courses/106/1 05/106105166/
2. NPTEL: Prof. Prabhakar, IISc Bangalore, Design for Internet of Things.

Assessment:

Internal Assessment (IA) for 20 marks:

- IA will consist of Two Compulsory Internal Assessment Tests. Each test will be 20 marks. Approximately 40% to 50% of syllabus content must be covered in First IA Test and remaining 40% to 50% of syllabus content must be covered in Second IA Test.

Question paper format:

- Question Paper will comprise of a total of **six questions each carrying 15 marks. Q.1** will be **compulsory** and should **cover maximum contents of the syllabus.**
- **Remaining questions** will be **mixed in nature** (part (a) and part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module randomly selected from all the modules).
- A total of Four questions need to be answered.

University of Mumbai

Course Code: ________________and Course Name:

Date of Exam: Duration: 2 Hours Max. Marks: 60

NB: (1) Question 1 is compulsory. (2) Solve any three questions out of five.		
Q.1	**Solve any three questions out of five: (05 marks each)**	**15**
a)		05
b)		05
c)		05
d)		05
e)		05
Q.2,3, 4,5,6		**15**
a)	Part(a) and Part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module.	07or 08
b)		08 or 07

Note for QP Setters:

1. All COs should be mapped on the basis of the syllabus.
2. Module weightage as per contact hours mentioned in the syllabus
3. Question 1 is compulsory, and it should include sub-questions covering all modules, with each sub-question carrying five marks.
4. The question paper should contain six questions, each of 15 marks.

****

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2166114	Cyber Security	3	-	-	3	-	-	3

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-I	IAT-II	IAT-I + IAT-II					
2166114	Cyber Security	20	20	40	60	2	--	--	100

Rationale **:**

This course provides foundational knowledge of computer security, network security, cybercrimes, and cyberspace regulations essential for understanding today's digital threat landscape. It equips learners with core cryptographic concepts—including symmetric keys, public key systems, and hashing techniques—critical for securing information systems. The course also builds awareness of common attack techniques. Finally, the course introduces the role of AI and ML in modern cybersecurity to enhance threat detection and cyber defense capabilities.

Course Objectives:

1. To understand the fundamentals of computer, network, and cyber security along with cybercrimes and cyberspace
2. To understand symmetric and public key cryptography along with hashing techniques and their applications
3. To understand how cyberattacks are planned and executed
4. To understand various cyber-attacks, malware techniques, and security threats
5. To understand the fundamentals of e-commerce, digital payments, security mechanisms, and regulatory guidelines.
6. To understand the role of AI and ML in modern security, threat detection, and cyber defense applications.

Course Outcomes:

1. Analyze security principles, cybercrime classifications, and the structure and regulation of cyberspace
2. Apply symmetric keys, public key algorithms, and hashing techniques to secure data and communication processes.
3. Analyze attack methods, mobile security threats, and organizational measures for securing environments.
4. Analyze different attack techniques and vulnerabilities
5. Evaluate e-commerce security practices, digital payment modes, related frauds, and RBI guidelines for secure transactions.
6. Analyze how AI-based techniques enhance threat detection, vulnerability assessment, and real-world cyber defense.

Prerequisite: Computer Networks

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
0	**Prerequisite**	Computer Networks concepts		
I	**Introduction to security**	Defining Computer security and Network Security, CIA, Services, Mechanisms and attacks. Information Security (InfoSec) **Cyber Security:** Concept of cyber security, Issues and challenges of cyber security**,** Cybercrime definition, Cybercrime and information security, Classifications of cybercrime, a global Perspective on cybercrimes. **Cyber Space:** Cyberspace, Architecture of cyberspace, Regulation of cyberspace **Self-learning Topics:** Security Policies, Standards, and Frameworks	6	CO1
II	**Cryptography**	**Symmetric key ciphers:** Block ciphers, stream ciphers, Data Encryption standard, Advanced Encryption Standard **Public key cryptography:** Principles Public key crypto Systems, Diffie Hellman Key Exchange, the RSA algorithm, Key Management **Hashing Techniques:** SHA256, SHA-512, HMAC and CMAC. **Self-learning Topics:** Modes of Operation for Block Ciphers, Public Key Infrastructure (PKI) and Digital Certificates, ECC	**7**	CO2
III	**Cybercrime Techniques and Attack Vectors**	Types of attacks, Social Engg, Cyber stalking, Cyber café and Cybercrimes, Bot nets, Attack vector, Cloud computing, Proliferation of Mobile and Wireless Devices, Trends in Mobility, Credit Card Frauds in Mobile and Wireless Computing Era, Security Challenges Posed by Mobile Devices, Registry Settings for Mobile Devices, Authentication Service Security, Attacks on Mobile/Cell Phones Mobile Devices: Security Implications for Organizations, Organizational Measures for Handling Mobile, Devices-Related Security Issues **Self-learning Topics:** Lifecycle of a Cyberattack, Real-World Case Studies	**7**	CO3
IV	**Tools and Methods Used in Cyber**	Phishing, Password Cracking, Key loggers and Spywares, Virus and Worms, Steganography, DoS and DoS Attacks, SQL Injection, Buffer Over Flow, Attacks on Wireless Networks, Phishing, Identity Theft (ID Theft). Penetration Testing Methodologies (OWASP, PTES, NIST) **Self-learning Topics:** Tools Used for Ethical Hacking, Secure Coding Practices	**5**	CO4

V	**E-Commerce and Digital Payments**	Definition of E- Commerce, Main components of E-Commerce, Elements of E-Commerce security, E-Commerce threats, E-Commerce security best practices, Introduction to digital payments, Components of digital payment and stake holders, Modes of digital payments Banking Cards, Unified Payment Interface (UPI), e-Wallets, Unstructured Supplementary Service Data (USSD), Aadhar enabled payments, Digital payments related common frauds and preventive measures. RBI guidelines on digital payments and customer protection in unauthorized banking transactions. **Self-learning Topics:** Case Studies of E-Commerce Security Breaches, Fraud Detection Techniques	**7**	CO5
VI	**AI in Cyber Security**	Need and importance of AI for modern security, Data Preparation for Security:Cybersecurity data (logs, network traffic, malware samples), Feature Engineering (extracting meaningful features like packet size, API calls), and Data Labeling (creating ground truth for attacks/benign activities) The Dual Role of AI: Offensive and Defensive: Adversarial AI Attacks (Attacks at Training Time - Poisoning Attack & Attacks at Deployment Time - Evasion Attack AI for Phishing and Social Engineering Detection, NLP models to detect phishing emails/messages,AI-powered URL reputation scoring, Deepfake detection Role of ML and DL in threat detection, AI for threat intelligence and pattern mining, AI in vulnerability scanning and patch prioritization, Real-world AI applications in cyber defense. **Self-learning Topics:** Case Studies of AI Implementation in Cyber Defense, AI-Based Malware Detection, Generative AI for creating sophisticated decoys or cyberattack simulations.	**7**	CO6

Text Books:

1. William Stallings, Cryptography and Network Security, Principles and Practice, Pearson Education
2. Behrouz A. Ferouzan, “Cryptography & Network Security”, Tata Mc Graw Hill
3. Nina Godbole, Sunit Belapure, “Cyber Security”, Wiley India
4. Advocate Prashant Mali, “Cyber Law & Cyber Crimes”, Snow White Publications
5. Elias M. Awad, “Electronic Commerce”, Prentice Hall of India Pvt Ltd
6. Leslie F. Sikos, “Artificial Intelligence for Cybersecurity”, Springer
7. Alessandro Parisi, Hands-On Artificial Intelligence for Cybersecurity: Implement smart AI systems for preventing cyber attacks and detecting threats and network anomalies, Packt Publishing.

References:

1. Atul Kahate, “Cryptography and Network Security”, Tata Mc Graw Hill

2. Kennetch J. Knapp, “Cyber Security &Global Information Assurance”, Information Science Publishing
3. Yadav & De, AI in Cybersecurity, CRC Press

Online References:

Sr. No.	Website Name
1.	https://www.nist.gov/cybersecurity
2.	https://www.cisco.com/c/en_in/products/security/what-is-cybersecurity.html
3.	https://www.npci.org.in

Assessment:

Internal Assessment (IA) for 20 marks:

- IA will consist of Two Compulsory Internal Assessment Tests. Each test will be 20 marks. Approximately 40% to 50% of syllabus content must be covered in First IA Test and remaining 40% to 50% of syllabus content must be covered in Second IA Test.

Question paper format:

- Question Paper will comprise of a total of **six questions each carrying 15 marks. Q.1** will be **compulsory** and should **cover maximum contents of the syllabus.**
- **Remaining questions** will be **mixed in nature** (part (a) and part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module randomly selected from all the modules).
- A total of Four questions need to be answered.

University of Mumbai

Course Code: _______________and Course Name:

Date of Exam: Duration: 2 Hours Max. Marks: 60

NB: (1) Question 1 is compulsory. (2) Solve any three questions out of five.		
Q.1	**Solve any three questions out of five: (05 marks each)**	**15**
a)		05
b)		05
c)		05
d)		05
e)		05
Q.2,3, 4,5,6		**15**
a)	Part (a) and Part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module.	07or 08
b)		08 or 07

Note for QP Setters:

1. All COs should be mapped on the basis of the syllabus.
2. Module weightage as per contact hours mentioned in the syllabus
3. Question 1 is compulsory, and it should include sub-questions covering all modules, with each sub-question carrying five marks.
4. The question paper should contain six questions, each of 15 marks.

....

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2166115	Nature-Inspired Optimization Algorithms	3	-	-	3	-	-	3

Course Code	Course Name	Theory					Term Work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-I	IAT-II	IAT-I + IAT-II					
2166115	Nature-Inspired Optimization Algorithms	20	20	40	60	2	-	--	100

Rationale:

This course introduces algorithms inspired by natural processes for solving complex optimization problems. Students will learn theoretical foundations, algorithm design, implementation, parameter tuning, and application to engineering problems. Emphasis is placed on hands-on implementation, experimental comparison, and critical evaluation.

Course Objectives:

1. To introduce the foundational principles and motivation behind nature-inspired optimization techniques.
2. To explore evolutionary algorithms and their application in multi-objective optimization problems.
3. To study swarm intelligence algorithms and their role in solving complex optimization tasks.
4. To understand novel and hybrid swarm-based optimization models and their underlying mechanisms.
5. To examine modern metaheuristic algorithms and their application in edge computing and IoT optimization.
6. To apply advanced optimization algorithms in real-world engineering and machine learning problems.

Course Outcomes: Upon successful completion, of course, learner/student will be able to

1. Explain the fundamental principles and categories of nature-inspired optimization algorithms.
2. Apply evolutionary algorithms to handle multi-objective optimization problems.
3. Analyse the working and performance of advanced swarm intelligence algorithms.
4. Implement novel swarm intelligence and hybrid optimization techniques.
5. Evaluate modern metaheuristic algorithms for optimization in Edge and IoT environments.
6. Design efficient optimization solutions for engineering and AI-based applications.

DETAILED SYLLABUS:

Sr. No	Name of Module	Detailed Content	Hours	CO Mapping
0	**Prerequisite**	Students should be familiar with: Linear Algebra, Probability and Statistics, Fundamentals of Optimization, Computational & Programming Skills, Knowledge of search and optimization in AI	**0**	
I	**Introduction to Nature Inspired Optimization**	1.1Basic principles of optimization: motivation and need for nature-inspired algorithms, Overview of classical vs. nature-inspired optimization. 1.2 Examples of nature-based algorithms: biological, physical process inspired, Types of nature-inspired algorithms: evolutionary, swarm intelligence, bio-inspired, **Self-learning Topics:** Problem solving and Optimization: A Super-Simple optimization problem. Performance Metrics: Measuring performance, Solution Quality (Error Rate, F1 Score, Accuracy), Convergence Speed, and Robustness.	**3**	**CO1**
II	**Advanced Evolutionary Algorithms**	2.1 Introduction to Multi-Objective Optimization: Concepts of Pareto optimality, dominance, Pareto front, and trade-off analysis. methods of MOO: classical and evolutionary Evolutionary algorithm framework for MOO; population-based search for Pareto front. solution 2.2 MOEA Algorithms: NSGA-II, SPEA-II, MOEA/D, and NSGA-III mechanisms. 2.3 Case Studies & Applications: MOEAs in feature selection, neural network training, scheduling, and energy optimization. **Self-learning Topics:** Parameter Tuning: Genetic Algorithms (GA): principles, operators, representation, selection, crossover, mutation. Evolution Strategies and Evolutionary Programming. Applications and variants.	**6**	**CO2**
III	**Advanced Swarm Algorithms**	3.1Advance Algorithms: Introduction to Particle Swarm Optimization (PSO), Grey Wolf Optimizer (GWO): α, β, δ, ω hierarchy guides search around prey, Whale Optimization Algorithm (WOA): Encircling prey and spiral updating mechanism. 3.2Dragonfly Algorithm: Separation, alignment, cohesion, attraction, and distraction rules. 3.3Multi-Objective Swarm Intelligence: Pareto-based PSO, Hybrid Swarm Models: PSO–GA hybrid, Firefly–ANN hybrid, ACO with local search; **Self-learning Topics:** Swarm Intelligence for AI & Edge Systems: Application in feature selection, hyperparameter tuning, federated learning, IoT & Edge resource optimization.	**8**	**CO3**

IV	**Novel Swarm Intelligence Optimization Algorithm (SIOA)**	4.1 Water Wave Optimization: Objective Function, Power Balance Constraints, Generator Capacity Constraints, Water Wave Optimization Algorithm, Mathematical Model of WWO Algorithm. 4.2 Brain Storm Optimization: Multi-Objective Brain Storm Optimization Algorithm, Clustering Strategy, Generation Process, Mutation Operator, Selection Operator. 4.3 Swarm Cyborg: Swarm Intelligence Cyborg, Swarm Cyborg Taxis Algorithms, Cyborg Alpha Algorithm. **Self-learning Topics:** Application of Swarm Intelligence: Swarm Intelligence Robotics, Swarm Robotics, System-Level Properties, Coordination Mechanisms.	**6**	**CO4**
V	**Modern Algorithms**	5.1 Ant Lion Optimizer (ALO): Random walks of ants influenced by antlion traps, Moth Flame Optimization (MFO): Logarithmic spiral motion towards flames. 5.2 Harris Hawks Optimization (HHO): Dynamic switching between soft and hard besiege. 5.3 NIOA for Edge Resource Optimization - Optimization of task scheduling, load balancing, and energy consumption in Edge networks. PSO for IoT data routing, Firefly for energy-efficient edge nodes. Apply NIOA techniques for optimizing performance in distributed Edge systems. **Self-learning Topics:** Emerging Research Directions: Quantum-behaved PSO, Deep Swarm Learning, Chaotic maps, Self-adaptive swarm models.	**8**	**CO5**
VI	**Advanced Topics & Engineering Applications**	6.1 Machine Learning & Deep Learning Optimization Applications: Cost-based & Correlation-based feature selection with swarm intelligence. Predictions with reduced features, SVM and random forest, Crime forecasting with PSO-SVM. CNN and LSTM optimization. 6.2 Case Studies in Engineering: Application of metaheuristics in: Scheduling and Logistics (e.g., vehicle routing). Machine Learning (e.g., hyperparameter tuning, feature selection). **Energy Consumption Prediction** using Multi-objective GA, **Scheduling Tasks in Cloud or Social Networks** using ACO, **Stock Market Forecasting** using Swarm Intelligence. **Self-learning Topics:** Feature Subset Selection: In-depth case studies using GAs, PSO, and Binary versions of swarm algorithms to find the optimal subset of features for classification tasks.	**8**	**CO6**

Books:

1. Nature-Inspired Optimization Algorithms, Xin-She Yang,
2. Introduction to Evolutionary Computing, A.E. Eiben and J.E. Smith.
3. Swarm Intelligence, James Kennedy et al.
4. Swarm Intelligence and Evolutionary Computation, Georgios N. Kouziokas.
5. Advanced Optimization by Nature-Inspired Algorithms, O. Bozorg-Haddad et al.
6. Swarm Intelligence by Kuldeep Singh Kaswan.
7. K. Deb, Multi-objective Optimization using Evolutionary Algorithms, Wiley, 2001.
8. Carlos Coello Coello, Gary B. Lamont, David A. van Veldhuizen, Evolutionary Algorithms for Solving Multi-Objective Problems, Springer, 2007
9. Seyedali Mirjalili, Evolutionary Algorithms and Metaheuristics in Engineering Applications, Springer, 2020.
10. Coello, C.A. et al., Evolutionary Algorithms for Solving Multi-Objective Problems, Springer, 2019.

Online References:

Sr. No.	Website Name
1.	https://onlinecourses.nptel.ac.in/noc25_me22
2.	https://www.youtube.com/watch?v=E2dJOYDz7b0

Assessment:

Internal Assessment (IA) for 20 marks:

- IA will consist of Two Compulsory Internal Assessment Tests. Each test will be 20 marks. Approximately 40% to 50% of syllabus content must be covered in First IA Test and remaining 40% to 50% of syllabus content must be covered in Second IA Test.

Question paper format:

- Question Paper will comprise of a total of **six questions each carrying 15 marks. Q.1** will be **compulsory** and should **cover maximum contents of the syllabus.**
- **Remaining questions** will be **mixed in nature** (part (a) and part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module randomly selected from all the modules).
- A total of Four questions need to be answered.

University of Mumbai

Course Code: ________________and Course Name:

Date of Exam: Duration: 2 Hours Max. Marks: 60

NB:
(1) Question 1 is compulsory.
(2) Solve any three questions out of five.

Q.1	**Solve any three questions out of five: (05 marks each)**	**15**
a)		05
b)		05
c)		05
d)		05
e)		05
Q.2,3, 4,5,6		**15**
a)	Part (a) and Part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module.	07or 08
b)		08 or 07

Note for QP Setters:

1. All COs should be mapped on the basis of the syllabus.
2. Module weightage as per contact hours mentioned in the syllabus
3. Question 1 is compulsory, and it should include sub-questions covering all modules, with each sub-question carrying five marks.
4. The question paper should contain six questions, each of 15 marks.

....

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2166116	**Edge Computing**	**3**	**-**	**-**	**3**	**-**	**-**	**3**

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-I	IAT-II	IAT-I + IAT-II					
2166116	**Edge Computing**	20	20	40	60	2	--	--	100

Rationale **:**

This course provides students with essential knowledge of modern IoT and Edge Computing systems, addressing the need for real-time and low-latency processing. It introduces practical skills using Raspberry Pi, sensor interfacing, and lightweight protocols like MQTT. By studying real-world edge applications, learners become capable of designing efficient and scalable IoT-Edge solutions for industrial and smart environments.

Course Objectives:

1. To understand the basics of Cloud, Fog, and Edge Computing for low-latency IoT systems.
2. To study IoT and Edge architectures and device-to-cloud integration.
3. To learn Raspberry Pi hardware, OS setup, and remote-access tools.
4. To develop skills in sensor interfacing and lightweight protocols like MQTT.
5. To explore Raspberry Pi as an edge device for real-time processing.
6. To analyze edge computing applications and AI-enabled edge solution**s.**

Course Outcomes:

1. Explain Cloud, Fog, and Edge Computing concepts and the need for edge services.
2. Describe IoT–Edge architecture, communication models, and data flow.
3. Configure Raspberry Pi, install OS, use GPIO, and enable remote access.
4. Interface sensors with Raspberry Pi and implement MQTT communication.
5. Apply Raspberry Pi for edge processing in Edge–Fog–Cloud systems.
6. Evaluate edge computing applications and use AI tools at the edge.

Prerequisite:

Basic knowledge of Computer Networks, Foundational understanding of IoT concepts, Hands-on experience with programming, preferably using Python or C. Basic Linux/command-line knowledge

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
0	**Prerequisite**	Concepts of IoT, Computer Network	01	
I	**Introduction to Cloud Computing, Fog Computing And Edge Computing**	Overview of Cloud Computing, Cloud computing & its limitation to support low latency and RTT, , Introduction to Fog Computing, Introduction to Edge Computing, Why move cloud services to Edge? **Self-learning Topics:** Edge Data Centres	04	CO1
II	**IoT and Edge Architecture**	Fundamentals of IoT and Edge Architecture: Overview of IoT ecosystem, device-to-cloud communication models, role of sensors, gateways, and cloud.	06	CO2

		IoT Edge Architecture Components & Data Flow: Architecture blocks: IoT devices, edge nodes, gateways, communication interfaces, cloud integration. **Self-learning Topics:** Running Docker containers on edge nodes like Raspberry Pi. Security in IoT and Edge Systems		
III	**Introduction to Raspberry Pi**	Introduction to Raspberry Pi, evolution of Raspberry Pi boards, features and capabilities of Raspberry Pi. Raspberry Pi board layout, GPIO pinouts, hardware components, interfaces, and peripheral connectivity.Installing and configuring operating systems on Raspberry Pi Connecting Raspberry Pi via SSH, using remote access tools such as VNC, PuTTY, and VS Code Remote, remote file transfer methods. **Self-learning Topics:** Overview of different Raspberry Pi models (Pi Zero, Pi 3, Pi 4, Pi 5) Basic Linux commands for Raspberry Pi users Virtual Raspberry Pi simulators (Wokwi, Pi-emulators)	06	CO3
IV	**Interfacing Raspberry Pi & Edge - to - Cloud Protocols**	IInterfacing DHT temperature –humidity sensor with Raspberry Pi, reading sensor values, data logging, and real-time monitoring. Need for lightweight protocols for IoT communication; Comparison of protocols: MQTT, CoAP, HTTP for IoT environments. Introduction to MQTT protocol and its role in IoT; Detailed MQTT architecture and workflow. MQTT Packet Structure & Data Formats. **Self-learning Topics:** Introduction to Mosquitto MQTT Broker installation on Raspberry Pi, Using Node-RED for MQTT-based IoT workflows, Cloud MQTT services (AWS IoT Core, Azure IoT Hub, Google IoT Core)	08	CO4
V	**Edge Computing With Raspberry Pi**	Fundamentals of edge computing, need for low-latency processing, comparison with cloud computing. Raspberry Pi as an edge device for data acquisition and local processing. Edge–Fog–Cloud hierarchy and deployment models. **Self-learning Topics:** Setting up Docker containers on Raspberry Pi for edge deployments, Edge security techniques (device authentication, secure boot, encrypted communication)	08	CO5
VI	**Applications of Edge Computing**	Study of edge computing use cases in smart cities, industrial automation, healthcare, transportation, agriculture, and retail. Implementation of edge nodes for video analytics, environmental monitoring, energy management, and autonomous systems. **Self-learning Topics:** Hands-On Exploration of Edge AI Tools and Platforms, Experimenting with TensorFlow Lite, NVIDIA Jetson, Google Coral, or EdgeX Foundry to develop AI-enabled edge	06	

		applications.		

Text Books:

1. **Handbook of Internet of Things and Big Data** – Pethuru Raj, Anupama C. Raman
2. **Raspberry Pi User Guide** – Eben Upton & Gareth Halfacree
3. **Architecting the Internet of Things** – Dieter Uckelmann, Mark Harrison

References:

1. **Internet of Things: A Hands-On Approach** – Arshdeep Bahga & Vijay Madisetti
2. **MQTT Essentials: A Lightweight IoT Protocol** – Gastón C. Hillar
3. **Edge Computing: A Primer** – Jie Cao, Qun Li, and Philip Yu

Online References:

Sr. No.	Website Name
1.	**NPTEL — Introduction to Internet of Things:** https://nptel.ac.in/courses/106/105/106105166/
2.	**Coursera — Internet of Things and Embedded Systems** https://www.coursera.org/specializations/iot
3.	**Udemy — Raspberry Pi for Beginners & IoT Developers** https://www.udemy.com/course/raspberry-pi-full-stack/

Assessment:

Internal Assessment (IA) for 20 marks:

- IA will consist of Two Compulsory Internal Assessment Tests. Each test will be 20 marks. Approximately 40% to 50% of syllabus content must be covered in First IA Test and remaining 40% to 50% of syllabus content must be covered in Second IA Test.

Question paper format:

- Question Paper will comprise of a total of **six questions each carrying 15 marks. Q.1** will be **compulsory** and should **cover maximum contents of the syllabus.**
- **Remaining questions** will be **mixed in nature** (part (a) and part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module randomly selected from all the modules).
- A total of Four questions need to be answered.

University of Mumbai

Course Code: _______________and Course Name:

Date of Exam: Duration: 2 Hours Max. Marks: 60

NB: (1) Question 1 is compulsory. (2) Solve any three questions out of five.		
Q.1	**Solve any three questions out of five: (05 marks each)**	**15**
a)		05
b)		05
c)		05
d)		05
e)		05
Q.2,3, 4,5,6		**15**
a)	Part (a) and Part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module.	07or 08
b)		08 or 07

Note for QP Setters:

1. All COs should be mapped on the basis of the syllabus.
2. Module weightage as per contact hours mentioned in the syllabus
3. Question 1 is compulsory, and it should include sub-questions covering all modules, with each sub-question carrying five marks.
4. The question paper should contain six questions, each of 15 marks.

****

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2166117	Digital Imaging Techniques and Analysis	3	-	-	3	-	-	3

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT -I	IAT-II	IAT-I + IAT-II					
2166117	Digital Imaging Techniques and Analysis	20	20	40	60	2	--	--	100

Rationale **:**
The course Digital Image Processing Techniques and Analysis enables students to understand and apply fundamental and advanced methods for processing, enhancing, and analyzing digital images. It bridges theory and practicality through tools like OpenCV and Python, preparing students to solve real-world problems in domains such as medical imaging, satellite image analysis, and machine vision. The course builds a foundation for higher studies and research in computer vision, artificial intelligence, and pattern recognition.

Course Objectives:

1. To introduce the fundamental concepts of digital image formation, acquisition, and representation.
2. To develop understanding of spatial and frequency domain methods for image enhancement.
3. To apply image restoration techniques in image processing.
4. To impart knowledge of segmentation and feature extraction techniques for image analysis.
5. To apply geometric and statistical approaches for image representation and classification.
6. To acquaint principles of image compression and applications of Image processing.

Course Outcomes:

1. Understand the fundamental principles of digital image formation and representation.
2. Apply image enhancement techniques for quality improvement.
3. Apply image restoration techniques to improve degraded image quality.
4. Implement Image segmentation and feature extraction techniques for analysis.
5. Represent and classify image data using geometric and statistical methods.
6. Evaluate image compression techniques and their applications in real-world systems.

Prerequisite: Mathematics for Computer Science, Computational Theory, Python Programming

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
0	**Prerequisite**	Mathematics for Computer Science, Computational Theory, Python Programming		
I	**Introduction to Digital Imaging**	Fundamentals of digital image processing system, Components of digital imaging system, Image formation and acquisition, Sampling and quantization. Sampling and quantization, Pixel relationships, Connectivity Color models (RGB, CMY, HSI), Human visual perception and image display techniques. **Self-learning Topics:** Image file Formats	05	CO1
II	**Image Enhancement Techniques**	Spatial domain techniques: point and neighborhood processing, Histogram processing – equalization, specification, Smoothing and sharpening filters. Frequency domain enhancement – Fourier Transform, 2D-DFT and its properties,, filtering concepts, Homomorphic filtering. **Self-learning Topics:** Comparison of Enhancement techniques	07	CO2
III	**Image Restoration**	Image degradation model Restoration filters – inverse, Wiener, and adaptive filters. Geometric transformations and image registration. Motion blur and deblurring. **Self-learning Topics:** Image reconstruction from projections	06	CO3
IV	**Image Segmentation and Feature Extraction**	Point, Line and Edge detection: Detection of isolated points, Line detection, Edge models, Sobel, Prewitt, Canny operators. Region-based segmentation: growing, splitting, merging. Morphological operations – dilation, erosion, opening, closing. Feature extraction: shape, texture, and color descriptors. **Self-learning Topics:** Thresholding techniques, Morphological watersheds	08	CO4
V	**Image Representation, Description, and Classification**	Boundary representation, chain codes, and polygonal approximation. Region descriptors: area, perimeter, moments, compactness. Principal Component Analysis (PCA). Image classification and clustering (K-Means, hierarchical). **Self-learning Topics:** Pattern recognition through case study on handwritten digits (0 - 9) from scanned images	06	CO5
VI	**Image Compression and**	Fundamentals of image compression – redundancy, fidelity criteria.	07	CO6

	Applications	Lossless compression: Run-length, Huffman, LZW. Lossy compression: Transform coding (DCT), JPEG, MPEG. Emerging Image Processing applications: Case study on medical. imaging, satellite imaging, machine vision **Self-learning Topics:** Real life applications of image processing techniques		

Text Books:

1. Rafael C. Gonzalez and Richard E. Woods, _Digital Image Processing‘, Pearson Education Asia, Third Edition, 2009
2. Christopher M. Bishop; Pattern Recognition and Machine Learning, Springer, 2006
3. Milan Sonka, Vaclav Hlavac and Roger Boyle, Digital Image Processing and Computer Vision, Cengage Learning, 2008
4. S. Jayaraman, E. Esakkirajan and T. Veerkumar, ―Digital Image Processing, TataMcGraw Hill Education Private Ltd, 2009

References:

1. W Pratt, "Digital Image Processing", Wiley Publication, 3rd Edition, 2002
2. Frank Y Shish ,Image Processing and Pattern Recognition: Fundamentals and Techniques, Wiley Wiley-IEEE Press, 2010
3. Fundamentals of digital image processing, by Anil K. Jain.

Online References:

Sr. No.	Website Name
1.	**NOC: Digital Image Processing, IIT Kharagpur** Prof. Prabir Kumar Biswas https://nptel.ac.in/courses/117105135
2.	Computer Vision And Image Processing - Fundamentals And Applications By Prof. M. K. Bhuyan | IIT Guwahati https://nptel.ac.in/courses/108103174
3.	**NOC: Computer Vision, IIT Kharagpur** Prof. Jayanta Mukhopadhyay https://nptel.ac.in/courses/106105216

Assessment:

Internal Assessment (IA) for 20 marks:

- IA will consist of Two Compulsory Internal Assessment Tests. Each test will be 20 marks. Approximately 40% to 50% of syllabus content must be covered in First IA Test and remaining 40% to 50% of syllabus content must be covered in Second IA Test.

Question paper format:

- Question Paper will comprise of a total of **six questions each carrying 15 marks. Q.1** will be **compulsory** and should **cover maximum contents of the syllabus.**
- **Remaining questions** will be **mixed in nature** (part (a) and part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module randomly selected from all the modules).
- A total of Four questions need to be answered.

University of Mumbai

Course Code: ________________and Course Name:

Date of Exam: Duration: 2 Hours Max. Marks: 60

NB: (1) Question 1 is compulsory. (2) Solve any three questions out of five.		
Q.1	**Solve any three questions out of five: (05 marks each)**	**15**
a)		05
b)		05
c)		05
d)		05
e)		05
Q.2,3, 4,5,6		**15**
a)	Part (a) and Part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module.	07or 08
b)		08 or 07

Note for QP Setters:

1. All COs should be mapped on the basis of the syllabus.
2. Module weightage as per contact hours mentioned in the syllabus
3. Question 1 is compulsory, and it should include sub-questions covering all modules, with each sub-question carrying five marks.
4. The question paper should contain six questions, each of 15 marks.

Course Code	Course Name	Teaching Scheme (Contact Hours)						
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2166118	Blockchain Technology	3	-	-	3	-	-	3

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT -I	IAT-II	IAT-I + IAT-II					
2166118	Blockchain Technology	20	20	40	60	2	--	--	100

Course Objectives:

1. Understand the fundamentals of blockchain technology, its history, evolution, components, and structure of blocks, enabling students to differentiate blockchain from traditional databases.
2. Analyze consensus mechanisms and blockchain platforms, comparing their features, strengths, and limitations, and understand mining processes in Bitcoin and other platforms.
3. Develop and deploy smart contracts on Ethereum using Solidity, understanding the Ethereum ecosystem, accounts, transactions, and related tools.
4. Examine private and consortium blockchains, including Hyperledger Fabric, and understand permissioned networks and consensus protocols like Paxos and Raft.
5. Evaluate cryptocurrencies, tokens, ICOs, security threats, and mitigation strategies, along with practical usage scenarios and wallet management.
6. Explore emerging trends in AI integration with blockchain, including AI-enabled smart contracts, decentralized AI models, and case studies for real-world applications.

Course Outcomes:

1. Describe the Blockchain fundamentals, block structures, Merkle trees, nodes, and miners, and differentiate Blockchain from traditional databases.
2. Compare consensus algorithms (PoW, PoS, PoET, etc.), analyze Blockchain platforms (Bitcoin, Ethereum, Hyperledger, Corda), and explain mining and transaction processes.
3. Design and implement smart contracts using Solidity, demonstrate Ethereum transactions, and understand Gas, EVM, and associated tools.
4. Analyze private and consortium Blockchains, evaluate Hyperledger Fabric frameworks, and explain consensus mechanisms like Paxos and Raft.
5. Evaluate various cryptocurrencies, tokens, ICOs, wallet types, and security threats, and propose mitigation strategies for Blockchain vulnerabilities.
6. Apply AI models to Blockchain use-cases, develop AI-enabled smart contracts, and evaluate real-world case studies integrating AI with Blockchain applications.

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
0	**Prerequisite**	Cryptography and system Security and Computer Network.		
I	**Introduction to Blockchain & Block Design**	**Introduction to Blockchain**: History, evolution, fundamentals concepts, components, types Key components in a Blockchain: Structure of a Block,	4	CO 1

		Block Header Hash and Block Height, The Genesis Block, Linking Blocks in the Blockchain, Merkle Tree, nodes, miners **Self-learning Topics** Difference Between Blockchain and Traditional Databases, Blockchain Architecture Layers		
II	**Consensus Mechanisms and Blockchain Platforms**	**Consensus:** Byzantine Generals Problem, consensus algorithms: PoW, PoS, PoET, PoA, LPoS, pBFT, Proof-of-Burn (PoB), Life of a miner, Mining difficulty, Mining pool and its methods. **Bitcoin:** History, Common terminologies, Bitcoin Mining, propagation Script and transactions **Overview of Blockchain platforms**: Bitcoin, Ethereum, Hyperledger, Corda, Ripple Comparative analysis of platform features, Blockchain as a Service (BaaS), Case study of platform applications. **Self-learning Topics:** Hybrid models combining PoW + PoS, How platforms address transaction throughput, latency, and cross-chain communication	7	CO 2
III	**Ethereum and Smart Contracts**	**Ethereum:** History, Components, Architecture of Ethereum, Consensus, Miner and mining node, Ethereum virtual machine, Ether, Gas, Transactions,Accounts, Patricia Merkle Tree, Swarm, Whisper and IPFS, complete transaction working and steps in Ethereum. **Smart Contracts:** Introduction to smart contracts, Solidity programming: set -up tools and installation, Basics, functions, Visibility and Activity Qualifiers, Ethereum networks, solidity compiler, solidity files and structure of contracts, data types, storages, array, functions. **Self-learning Topics:**Overview of Ethereum's shift from PoW to PoS consensus, Tools for auditing and securing smart contracts	8	CO 3
IV	**Private and Consortium Blockchains**	**Introduction to Private Blockchain**: Key characteristics, need, Private vs Consortium Blockchains, Smart contracts in private Blockchain. **Hyperledger:** Introduction to Hyperledger, Tools and Frameworks, Hyperledger Fabric, Fabric Architecture, Comparison between Hyperledger Fabric & Other Technologies, Paxos and Raft consensus. **Self-learning Topics:** Real -world implementations in supply chain, banking, healthcare etc., Role of MSPs in authenticating participants in permissioned networks	7	CO 4
V	**Cryptocurrencies and digital tokens**	Cryptocurrency basics, types, usage, ERC20 and ERC721 Tokens, **ICO**: basics and related terms, launching an ICO, pros and cons, evolution and platforms, **STO**, Different Crypto currencies, Defi, Metaverse, **Cryptocurrency wallets**: Hot and cold wallets, Cryptocurrency usage, Transactions in Blockchain, UTXO and double spending problem **Common security threats in Blockchain**:	7	CO 5

		Mitigating attacks: 51% attack, Sybil attack, replay attack. **Self-learning Topics:**Stablecoins and Central Bank Digital Currencies, Tokenomics and Cryptocurrency Valuation Models		
VI	**Emerging trends of AI in Blockchain and case studies**	**AI–Blockchain Integration:** Overview, need, and synergy between AI and Blockchain, benefit, AI Models on Blockchain: decentralized AI. **Case study:** AI for transaction validation, fraud detection, and performance optimization, AI-based consensus mechanisms for various applications like Supply chain, Healthcare, Energy, Education. **AI-enabled Smart Contracts:** Intelligent decision-making and automation using AI agents in smart contracts, AI in Crypto Trading, Anti-Money Laundering & Risk Scoring LLMs in Smart Contract Lifecycle (Dev, Audit, Test, Docs). **Self-learning Topics:** Case Studies on AI in Blockchain	6	CO 6

Text Books:

1. Chandramouli Subramanian, Asha A. Georgeet al., "Blockchain Technology", Universities Press, 2020.
2. Dr. Asharaf S. & Sivadas Neelima, "Blockchain Technology: Algorithms and Applications", Wiley India, 2023.
3. Andreas M. Antonopoulos & Dr. Gavin Wood, "Mastering Ethereum: Building Smart Contracts and DApps", O'Reilly Media..
4. Ritesh Modi, Solidity Programming Essentials: A Beginner's Guide to Build Smart Contracts for Ethereum and Blockchain, Packt Publishing
5. Imran Bashir, "Mastering Blockchain: Inner Workings of Blockchain, Cryptocurrencies & Smart Contracts" ,Packt Publishing.
6. Ashwani Kumar,"Hyperledger Fabric In-Depth: Learn, Build and Deploy Blockchain Applications Using Hyperledger Fabric", BPB Publications.
7. Dr. Vaibhav Agrawal, *Practical Artificial Intelligence and Blockchain*, BPB Publications, 2021.

References:

1. Andreas M. Antonopoulos, Mastering Bitcoin: Programming the Open Blockchain, 2nd Edition, O'Reilly Media, Inc., 2017. ISBN: 9781491954386.
2. Dr. Gavin Wood, Mastering Ethereum: Building Smart Contracts and DApps, O'Reilly Media, 2018.
3. Kumar Saurabh & Ashutosh Saxena, Blockchain Technology: Concepts and Applications, Wiley Publication, 2020.
4. Antony Lewis, The Basics of Bitcoins and Blockchains: An Introduction to Cryptocurrencies and the Technology that Powers Them, Mango Publishing, 2018.

Online References:

Sr. No.	Website Name
1.	Solidity (smart contract language) - https://docs.soliditylang.org/
2.	web3.js (Ethereum JS library) - https://web3js.readthedocs.io/
3.	ethers.js (alt JS library for Ethereum) - https://docs.ethers.org/v5/
4.	Hyperledger Fabric (permissioned blockchain framework) - https://hyperledger-fabric.readthedocs.io/

Assessment:

Internal Assessment (IA) for 20 marks:

- IA will consist of Two Compulsory Internal Assessment Tests. Each test will be 20 marks. Approximately 40% to 50% of syllabus content must be covered in First IA Test and remaining 40% to 50% of syllabus content must be covered in Second IA Test.

Question paper format:

- Question Paper will comprise of a total of **six questions each carrying 15 marks. Q.1** will be **compulsory** and should **cover maximum contents of the syllabus.**
- **Remaining questions** will be **mixed in nature** (part (a) and part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module randomly selected from all the modules).
- A total of Four questions need to be answered.

University of Mumbai

Course Code: ________________and Course Name:

Date of Exam: Duration: 2 Hours Max. Marks: 60

NB: (1) Question 1 is compulsory. (2) Solve any three questions out of five.		
Q.1	**Solve any three questions out of five: (05 marks each)**	**15**
a)		05
b)		05
c)		05
d)		05
e)		05
Q.2,3, 4,5,6		**15**
a)	Part (a) and Part (b) of each question must be from different modules. For example, if Q.2 has part (a) from Module 3 then part (b) must be from any other Module.	07or 08
b)		08 or 07

Note for QP Setters:

1. All COs should be mapped on the basis of the syllabus.
2. Module weightage as per contact hours mentioned in the syllabus
3. Question 1 is compulsory, and it should include sub-questions covering all modules, with each sub-question carrying five marks.
4. The question paper should contain six questions, each of 15 marks.

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2166411	Mini Project-II	-	4	-	-	2	-	2

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Practical/ Oral	Total
		Internal assessment			End Sem. Exam			
		IAT-I	IAT-II	IAT-I+IAT-II				
2166411	Min Project-II	-	-	-	-	50	--	50

Objectives:	
1	To enable students to identify real-world problems through need-based surveys, literature review, and structured requirement analysis.
2	To develop skills in formulating clear problem statements and project objectives based on feasibility, innovation, and societal relevance.
3	To train students in designing and planning project execution using tools such as Gantt charts, PERT/CPM, and structured methodologies.
4	To provide hands-on experience in developing and implementing functional prototypes or models using appropriate engineering tools and domain-specific technologies.
5	To enhance teamwork, leadership, and project management skills through collaborative group work, progress tracking, and review presentations.
6	To cultivate professional competencies in documentation, technical communication, validation, and evaluation of project outcomes with sustainability and continuous learning focus.

Outcome: Learner will be able to…	
1	Formulate a real-world problem by conducting need-based surveys and demonstrating professional responsibility in problem selection.
2	Implement appropriate engineering methods to develop a functional solution while collaboratively exhibiting ethical teamwork behaviour.
3	Demonstrate effective leadership or collaboration by actively coordinating project tasks and responding constructively to group dynamics.
4	Validate experimental results through analytical techniques and justify findings with disciplined technical documentation.

5	Evaluate the societal and environmental impact of the developed solution while applying sustainable engineering practices.
6	Execute project activities using standard engineering tools while exhibiting commitment towards continuous self-directed learning.

Guidelines for Min Project-II	
1	Students shall form a group of 3 to 4 students, while forming a group shall not be allowed less than three or more than four students, as it is a group activity.
2	Interdisciplinary project is also allowed.
3	Students should do survey and identify needs, which shall be converted into problem statement for project in consultation with faculty supervisor/head of department/internal committee of faculties.
4	Students shall submit implementation plan in the form of Gantt/PERT/CPM chart, which will cover weekly activity of project.
5	A logbook to be prepared by each group, wherein group can record weekly work progress, guide/supervisor can verify and record notes/comments.
6	Faculty supervisor may give inputs to students during project activity; however, focus shall be on self-learning.
7	Students in a group shall understand problem effectively, propose multiple solution and select best possible solution in consultation with guide/ supervisor.
8	Students shall convert the best solution into working model using various components of their domain areas and demonstrate.
9	The solution to be validated with proper justification and report to be compiled in standard format of University of Mumbai.
10	With the focus on the self-learning, innovation, addressing societal problems and entrepreneurship quality development within the students through the project.

Term Work		
The review/ progress monitoring committee shall be constituted by head of departments of each institute. The progress of project to be evaluated on continuous basis, minimum two reviews in the semester.		
In continuous assessment focus shall also be on each individual student, assessment based on individual's contribution in group activity, their understanding and response to questions.		
Distribution of Term work marks shall be as below:		**Marks**
1	Marks awarded by guide/supervisor based on logbook	5
2	Marks awarded by review committee (Average of Review 1 & Review 2)	40
	Project Review-1	

		Identification of Problem	4
		Requirement analysis and Feasibility of the proposed work	4
		Literature Review	4
		Objectives of the proposed work	4
		Methodology of the proposed work	4
		Total Marks	20
		Project Review-2	
	a	Planning of project work and team structure	4
	b	Design Methodology	4
	c	Conceptual and Technical Demonstration	4
	d	Presentation: Oral delivery, contact with audience, slides, timing	4
	e	Quality of answers	4
		Total Marks	20
3		**Quality of Project report**	**5**

References to get Project ideas:

- https://www.geeksforgeeks.org/project-idea-college-network/?ref=ml_lbp
- https://www.simplilearn.com/tutorials/artificial-intelligence-tutorial/ai-project-ideas
- https://roadmap.sh/backend/project-ideas
- https://webflow.com/blog/website-ideas
- https://gist.github.com/MWins/41c6fec2122dd47fdfaca31924647499
- https://www.projectpro.io/article/artificial-intelligence-project-ideas/461
- https://github.com/The-Cool-Coders/Project-Ideas-And-Resources
- https://nevonprojects.com/project-ideas/software-project-ideas/
- https://roadmap.sh/projects

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2166119	Machine Learning & Swarm Intelligence Lab		2	-		1	-	1

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Practical/ Oral	Total
		Internal assessment			End Sem. Exam			
		IAT-I	IAT-II	IAT-I + IAT-II				
2166119	Machine Learning & Swarm Intelligence Lab	--	--	--	--	25	25	50

Lab Objectives:

1. To introduce the foundational concepts of Machine Learning and its mathematical underpinnings.
2. To implement and evaluate supervised and unsupervised learning models on real-world datasets.
3. To understand neural network learning mechanisms and dimensionality reduction techniques.
4. To study the fundamental principles of Swarm Intelligence and nature-inspired computation.
5. To design and simulate key Swarm Intelligence algorithms such as PSO, ACO, Firefly, and Cuckoo Search.
6. To integrate Swarm Intelligence methods with Machine Learning models for optimization and feature selection.

Lab Outcomes:

After successful completion, students will be able to:

1. Explain the Machine Learning workflow and evaluate model performance using standard metrics.
2. Apply supervised and unsupervised algorithms for prediction, clustering, and pattern recognition.
3. Implement neural network architectures and perform dimensionality reduction using PCA.
4. Demonstrate understanding of Swarm Intelligence principles and apply them to optimization problems.
5. Design and evaluate metaheuristic algorithms such as PSO, ACO, Firefly, and Cuckoo Search.
6. Create hybrid Swarm–ML systems for real-world optimization and decision-support applications.

Suggested List of Experiments:

Sr No	List of Experiments	Hrs
1.	Introduction to platforms such as Anaconda, COLAB, Study of Machine Learning Libraries and tools (Python library, tensorflow, keras,)	02
2.	Implement data preprocessing and normalization on a given dataset; split into training / testing sets.	02
3.	Apply Linear Regression and evaluate model performance using RMSE and R^2.	02
4.	Implement Logistic Regression/ Random Forest for binary classification (e.g., diabetes / spam dataset).	02
5.	Implement Support Vector Machines	02
6.	Perform clustering using K-Means and visualize cluster separation.	02
7.	Design a simple Perceptron or MLP using Python / TensorFlow; observe learning curve.	02
8.	Perform Principal Component Analysis (PCA) for dimensionality reduction and visualize results.	02
9.	Implement Particle Swarm Optimization (PSO) to minimize benchmark function (Sphere / Rastrigin).	02
10.	Implement Ant Colony Optimization (ACO) for shortest-path (TSP / graph) problem.	02
11.	Implement Firefly or Cuckoo Search algorithm for optimization task; record convergence.	02
12.	Apply PSO for feature selection in a classification model and compare results with baseline ML model.	02
13.	Comparative study of two Swarm algorithms based on convergence and accuracy.	02
14.	*Implement a Machine Learning model using fundamental principles in any programming language (C/C++/Java) without using pre-built libraries.	02

* is Compulsory Experiment

Sr No	List of Assignments (Any Two)	Hrs
01	Case Study: Applications of Machine Learning in engineering and data-driven decision systems.	02
02	Comparison of Supervised vs Unsupervised Learning Algorithms.	02
03	Comparative study of Linear Regression vs. Logistic Regression vs. SVM with suitable use-cases.	02
04	Short report on neural architectures: Perceptron vs. MLP and effect of activation functions on learning.	02
05	Critical note on the role of PCA in dimensionality reduction and impact on ML model performance.	02
06	Swarm Intelligence: Concept of self-organization, intensification vs diversification, and metaheuristic motivation.	02
07	Performance comparison of PSO, ACO, Firefly, and Cuckoo Search for optimization — factors affecting convergence.	02
08	Study on hybrid Swarm-ML feature selection / hyperparameter tuning — real-world applications.	02

Text Books:
1. Ethem Alpaydın, —Introduction to Machine Learning‖, MIT Press McGraw-Hill Higher Education
2. Marc Peter Deisenroth, Aldo Faisal, Cheng Soon Ong, —Mathematics for machine learning‖,
3. James Kennedy & Russell Eberhart, *Swarm Intelligence*, Morgan Kaufmann
4. Handbook of Swarm Intelligence: Concepts, Principles and Applications (Panigrahi, Shi & Lim, Springer, 2011)

References:
1. Tom M. Mitchell, —Machine Learning‖, McGraw Hill
2. Particle Swarm Optimization by Maurice Clerc (2006)
3. Swarm Intelligence Optimization: Algorithms and Applications- Abhishek Kumar, Pramod Singh Rathore, Vicente Garcia Diaz & Rashmi Agrawal, Published by Wiley

4. Shai Shalev-Shwartz, Shai Ben-David, ―Understanding Machine Learning‖, Cambridge University Press

Online Resources:

Sr. No.	Website Name
1.	https://www.coursera.org/learn/machine-learning
2.	https://onlinecourses.nptel.ac.in/noc25_me22/preview
3.	https://www.coursera.org/learn/evolutionary-computation-and-its-engineering-applications

Assessment:

Term Work: Term Work shall consist of at least 10 to 12 practical's based on the above list. Also, Term work Journal must include at least 2 assignments.

Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)

Practical& Oral Exam: An Oral & Practical exam will be held based on the above syllabus.

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2166120	**Data Analytics and Visualization Lab**	-	2	-	-	1	-	1

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Practical/ Oral	Total
		Internal assessment			End Sem. Exam			
		IAT-I	IAT-II	IAT-I + IAT-II				
2166120	**Data Analytics and Visualization Lab**	--	--	--	--	25	25	50

Lab Objectives:

1. To learn data cleaning, preparation, and feature engineering using Python and R, including normalization and dimensionality reduction.
2. To develop skills in building, evaluating, and interpreting regression and logistic models with data preparation and performance assessment using Python or R.
3. To develop skills in data exploration and anomaly detection, including identifying outliers using statistical, distance-based, and density-based methods for real-time applications.
4. To develop skills in analyzing, modelling, and forecasting time series data using Python or R.
5. To develop skills in creating and interpreting static and interactive visualizations using Python libraries like Matplotlib, Seaborn, Plotly, and Dash.
6. To develop skills in creating and interpreting static and interactive visualizations in R using ggplot2, plotly, shiny, and base R functions.

Lab Outcomes:

1. Understand the fundamentals of data analytics, normalization and feature engineering and PCA to improve predictive model performance.
2. Apply and analyze different **regression and evaluate model performance using Python or R**
3. Comprehend and apply exploratory data analysis and anomaly detection techniques using Python or R.

4. Evaluate time-series data by exploring and decomposing it, fitting ARIMA models, and assessing forecast accuracy using RMSE, MAE, and MAPE.

5. Create charts, plots, and interactive dashboards to visualize data patterns and identify outliers using Python
6. Apply data visualization techniques to create and interpret charts, plots, heatmaps, and dashboards to analyze trends, distributions, and relationships in data using R and other visualization tools like power BI/Tabulea/ open source tools

DETAILED SYLLABUS:

Sr. No.	Module	Detailed Content	LO Mapping
0	**Prerequisite**	Data mining, python Programming	
I	**Introduction to data analytics**	**1.** Overview of data analytics, Types of data: structured, unstructured, semi-structured, Introduction to Python and R for data analytics. **2.**Python Libraries: NumPy, Pandas, Scikit-learn R Libraries: dplyr, tidyr **3.** Importance of data cleaning and preparation Handling missing values and duplicates Data normalization and standardization techniques. **4.** Concept of features and importance in predictive modelling Creating new features from existing data. Feature importance and dimensionality reduction overview (PCA). **Self- learning Topics:** Different feature engineering techniques to improve model performance	LO1
II	**Regression Model Evaluation.**	**1.** Concept of dependent and independent variables. The regression line and equation. Assumptions of linear regression. Estimation of coefficients using Least Squares Method, Residual analysis, Extension of simple linear regression with multiple predictors. **2.**Model evaluation metrics: RMSE, MAE, MAPE,nonlinear regression ,Model fitting and interpretation **3.**Introduction to logistic regression for classification Logit function and odds ratio Model fitting and interpretation **Self- learning Topics**: Explore Advance Regression Model	LO2
III	**Data Exploration and Anamoly Detection**	**1. Concepts of Data Exploration:** Different techniques: EDA and feature exploration. 2.**Anomaly Detection:** Outliers, Causes of Outliers, Anomaly detection techniques, Outlier Detection using Statistics, Outlier Detection using Distance based method, Outlier detection using density-based methods for real time applications **Self- learning Topics**: Anomaly detection in high-volume data.	LO3
IV	**Time Series Analysis**	**1. Types of time series:** univariate and multivariate, Importing and exploring time series data in Python (pandas, numpy) / R (ts, xts) Extracting time series components, Visualizing decomposed components, Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF). **2.**Interpretation for model identification AR, MA, ARMA, and ARIMA concepts ,Fitting ARIMA models in Python (statsmodels) / R (forecast package) ,	LO4

		accuracy metrics for forecasting: MAPE, RMSE, MAE for real time applications. **Self-learning Topics:** Deep learning models for time series forecasting	
V	**Data Visualization using Python**	**1.Python Libraries for Visualization :** Matplotlib, Seaborn, **2.Plotly and Dash:** interactive visualizations and dashboards, stacked bar charts , Scatter plots for relationships, Boxplots and violin plots for distribution and outliers, Interactive plots using Plotly, Histograms and density plots. **Self-learning Topics**: making Seaborn plots interactive using plotting libraries such as Bokeh	LO5
VI	**Data visualization using R and Tools like Power BI/Tableau/open source tools**	*1. Base R plotting functions: plot(), hist(), barplot(), boxplot(), ggplot2, plotly for interactive plots, shiny for interactive dashboards.* *2.Other packages: lattice, corrplot, dplyr (for data manipulation), Line plots for trend analysis, Bar charts and histograms, Pie charts and stacked bar charts, Scatter plots for analyzing relationships, Boxplots and violin plots for distributions and outliers, Heatmaps for correlation matrices, Density plots and area plots, Histograms , Interactive plots with plotly and ggplotly(): Adding tooltips, hover info, and clickable elements, Combining static (ggplot2) and interactive (plotly) visualizations.* **3.**Overview of Power BI or Tableau capabilities, types of dashboards (operational, analytical, strategic), sources and structure of business data, importing and preprocessing datasets, creating relationships between tables, and applying visualization techniques using Power BI or Tableau, or open-source tools. **Self- learning Topics:** Storytelling with dashboards	L06

Text Books:

1. Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data,EMC Education services Wiley Publication
2. Peter Bruce, Andrew Bruce & Peter Gedeck, Practical Statistics for Data Scientists 50+ Essential Concepts Using R and Python, O'Reilly Publications 2nd Edition.
3. Mastering Data Visualization with Tableau by Dr. Arpana Chaturvedi, Prof. Praveen Malik.
4. Gagan Gupta Raj, Naresh Kumar Nagwani, Data Analytics and Visualization,, PHI Learning.
5. R for Data Science: Import, Tidy, Transform, Visualize, and Model Data by Hadley Wickham and Garrett Grolemund
6. Interactive Data Visualization with Python: Present Your Data As an Effective and Compelling Story, 2nd Edition by Abha Belorkar, Sharath Chandra Guntuku, and Shubhangi Hora.

References:

1. David G. Kleinbaum & Mitchel Klein, Logistic Regression: A Self-Learning Text , Springer
2. *Dr. Anil Maheshwari* , *"Data Analytics Made Accessible"* (2nd Edition, 2020) , Mcgraw Hill Education.
3. Dr. Bharti Motwani, Data Analytics using Python, Wiley India Pvt. Ltd

Online Resources:

Sr. No.	Website Name
1.	https://www.coursera.org/learn/introduction-to-data-analytics?utm_source=chatgpt.com
2.	https://www.udemy.com/course/data-storytelling-mastering-data-visualization-techniques/?couponCode=PMNVD2025
3.	https://www.datacamp.com/category/data-visualization?page=1
4.	https://onlinecourses.swayam2.ac.in/imb25_mg218/preview
5.	https://www.udemy.com/course/data-analytics-masters-basics-to-advanced/?couponCode=PMNVD2025

List of Experiments:

Note: * marked experiments are compulsory

Sr No	List of Experiments	Hrs
1	To demonstrate the use of data analytics libraries in Python and R for dataset pre-processing and analysis.	2
2	To perform data normalization, outlier detection, and categorical data encoding to improve dataset quality.	2
3	* To apply feature engineering techniques for creating, transforming, and selecting relevant features to enhance model performance.	2
4	Simple Linear Regression in Python/R.: To develop, train, and evaluate regression models for prediction and classification tasks, and to compare their performance using suitable evaluation metrics.	2
5	*Multiple Linear Regression in Python/R:To develop, train, and evaluate regression models for prediction and classification tasks, and to compare their performance using suitable evaluation metrics.	2
6	Logist Regression in Python/R:To develop, train, and evaluate regression models for prediction and classification tasks, and to compare their performance using suitable evaluation metrics.	2
7	Outlier detection using distance based/density based method.	2
8	Perform EDA on the given data set	2
9	*To Build and Evaluating an ARIMA Model for Applications of ARIMA (Examples: Economics and Finance, Healthcare, weather Forecasting)	2
10	*To perform Time Series Analysis in Python/R	2
11	To analyze and interpret data using data visualization techniques in Python	2
12	To analyze and interpret data using data visualization techniques in R	2
13	*To create an interactive dashboard using real-world business data for decision-making insights by visualization tool (Power BI/Tableau)	2

Note: It is suggested to assign one assignment covering Units 1 to 3, and another assignment covering Units 4 to 6 or refer the below suggested assignments.

Sr No	List of Assignments
01	To understand and apply various pre-processing techniques to improve dataset quality for regression modelling. Select any real-world dataset (e.g., from Kaggle, UCI Repository, or Government open data).
02	To build and evaluate regression models for predictive accuracy and interpretability. (Use the pre-processed dataset from Assignment 1.)
03	A company wants to predict customer churn using attributes such as Age, Income, Tenure, Satisfaction_Score, and Usage_Frequency. Tasks: 1. Illustrate with an example how PCA reduces redundancy in features. 2. Explain how PCA can improve the performance of a machine learning model.
04	Suppose you have the monthly revenue of a telecom company for 6 years. Describe the complete process you would follow to forecast revenue for the next year
05	Compare strengths and weaknesses of different python visualization tools
06	Write a short note on :Analyzing Recent Trends in Data Analytics and Visualization Across Industry applications Applications:

Assessment:

Term Work: Term Work shall consist of at least 10 to 12 practical's based on the above list. Also, Term work Journal must include at least 2 assignments.

Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)

Practical& Oral Exam: An Oral & Practical exam will be held based on the above syllabus.

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2166121	Internet of Things Lab	-	2	-	-	1	-	1

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Practical/ Oral	Total
		Internal assessment			End Sem. Exam			
		IAT-I	IAT-II	IAT-I + IAT-II				
2166121	Internet of Things Lab	--	--	--	--	25	25	50

Lab Objectives:

1. To understand the interfacing of sensors and actuators with IoT development boards for data acquisition and control.
2. To learn how to transmit, visualize, and store real-time sensor data on cloud platforms.
3. To implement web-based and mobile-based IoT control mechanisms for automation applications.
4. To explore IoT communication protocols such as MQTT and HTTP for device-to-cloud connectivity.
5. To design and develop IoT-based applications integrating cloud platforms like ThingSpeak, Firebase, and Blynk.
6. To analyze and evaluate the performance of IoT systems for real-world use cases such as home automation and environmental monitoring.

Lab Outcomes:

After successful completion of the lab, students will be able to:

1. Interface various sensors and actuators with IoT boards such as NodeMCU, ESP32, and Raspberry Pi.
2. Acquire and upload real-time sensor data to cloud services using appropriate IoT protocols.
3. Develop simple IoT applications to monitor and control devices via web and mobile interfaces.
4. Implement MQTT and REST-based communication for device data exchange and automation.
5. Integrate IoT systems with platforms like Thing Speak, Firebase, and Blynk for data visualization and control.
6. Design and demonstrate end-to-end IoT solutions for smart home, environment, and automation applications.

Prerequisite: Knowledge of Basic sensors and Arduino Boards

DETAILED SYLLABUS:

Sr. No.	Module	Detailed Content	LO Mapping
0	**Prerequisite**	Basic introduction to Microprocessor and microcontroller programming, hands-on with C and python programming	

I	**Arduino**	Role of Arduino in IoT, Introduction to Arduino Boards (Uno, Nano, Mega), Arduino IDE Setup and Tools, Structure of Arduino Sketch (setup and loop functions) Variables, Data Types, Operators, Digital I/O: Reading from and Writing to Pins, Serial Communication and Serial Monitor Self-learning Topics: Interrupts and Power Management, Integrating Arduino with Other Platforms	**LO1**
II	**ESP32**	Overview of ESP32 microcontroller, ESP32 architecture and features, Comparison with Arduino and Raspberry Pi, Power modes and energy efficiency, Typical IoT applications using ESP32 Self-learning Topics: Interfacing Sensors and Actuators, Web and Cloud Integration	**LO2**
III	**Raspberry Pi**	Overview of Raspberry Pi models and features, Comparison with Arduino, Components of a Raspberry Pi board, Installing and configuring Raspberry Pi OS, SSH, VNC, and remote access setup, Basic Linux terminal commands, Python Programming on Raspberry Pi Self-learning Topics: Operating Systems & Customization with Raspberry Pi	**LO3**
IV	**Sensors and Actuators in IoT Applications**	Temperature sensors (LM35, DHT11, TMP36), Pressure sensors (BMP180, MPX5010), Motion and position sensors (PIR, Ultrasonic HC-SR04, IR), Light sensors (LDR, TSL2561). Sound sensors (microphone, sound detector module), DC motors, Relays, Interfacing with microcontrollers (Raspberry Pi / Arduino / ESP32) Self-learning Topics: Practical Applications and Case Studies	**LO4**
V	**Web & IoT Applications**	Setting up a **local web server** (Apache, Flask, Node.js), Interfacing IoT devices with web servers, Data storage, retrieval, and visualization techniques, Building simple dashboards for sensor monitoring, Self-learning Topics: practice with Popular IoT cloud platforms like ThingSpeak, Adafruit IO, Blynk, Google Firebase, AWS IoT Core, Azure IoT Hub etc.	**LO5**
VI	**Project Case Studies**	Development of IoT based Applications Self-learning Topics: Software development for complete IoT applications	**LO6**

Text Books:

1. 21 IoT experiments, Yashawant Kanetkar, Srirang Korde, BPB Publications.
2. Developing IoT Projects with ESP 32, Vedat Ozan Oner, Packt Publishing, 2023
3. Raspberry Pi IoT Projects, J Shovic, Apress.

References:

1. "Exploring Arduino" by Jeremy Blum, John Wiley & Sons, 22 Jul 2013
2. "Arduino Cookbook" by Michael Margolis, *Michael Margolis* with Nick Weldin (O'Reilly).

Online Resources:

Sr. No.	Website Name
3.	Official Arduino documentation: https://www.arduino.cc
4.	https://thingsboard.io
3.	https://mqttlab.iotsim.io
4	https://openremote.io/
5	https://www.instructables.com/projects/
6	https://iotdesignpro.com/esp8266-projects

Sr No	List of Assignments / Tutorials
Module 1: Introduction to IoT	
01	Explain the physical and logical design of IoT with suitable examples. Illustrate the role of sensors, actuators, and IoT boards such as Arduino, NodeMCU, and Raspberry Pi in IoT systems.
02	Compare various IoT communication models and APIs. Discuss their significance in connecting smart devices.
Module 2: Design Principles for Connected Devices and Web Connectivity	
03	Differentiate between wired (UART, SPI, I²C) and wireless (NFC, RFID, Bluetooth, ZigBee, Wi-Fi) communication technologies based on range, power, and topology.
04	Develop a short report on MQTT or CoAP protocol implementation for IoT data transfer. Include message structure and use cases.
Module 3: IoT Network Architecture and Design	
05	Draw and explain the IoT World Forum (IoTWF) Reference Architecture highlighting key layers and their responsibilities.
06	Discuss the role of edge, fog, and cloud computing in IoT data management. Provide an example illustrating their interrelation.
Module 4: IoT Analytics for Cloud	
07	Describe the phases of IoT data analytics — descriptive, predictive, and prescriptive — with a suitable example for each.
08	Compare cloud service models (IaaS, PaaS, SaaS) in the context of IoT analytics and application deployment.
Module 5: IoT Privacy, Security, and Vulnerabilities	
09	Identify and analyze common vulnerabilities in IoT systems. Suggest techniques for securing IoT devices and data using layered security models.
Module 6: IoT Platform Design Methodology and Case Studies	
10	Design an IoT solution for any one of the following case studies: Smart City (Parking or Waste Management) * Connected Car System * Smart Agriculture (Irrigation or Environmental Monitoring) Include problem definition, architecture, components used, and expected outcomes.

Suggested List of Experiments
Students can **Use Any Board** such as Arduino/ Node MCU/ ESP32/ R-pi as per requirement.
Assessment:

Sr No	List of Experiments	Hrs
Prerequisite	To study and explore Tinkercad Simulator.	02
01	Interfacing a Temperature and Humidity Sensor (DHT11) with any board.	02
02	Interfacing LED/Buzzer/ Motion Sensor (PIR)/ ultrasonic sensor using a Web-Based Interface via any board.	02
03	IoT-Based Light Intensity Monitoring System using LDR and any board.	02
04	Python program for communication between clients based on MQTT.(using paho library).	02
05	Displaying Sensor Data on a Local Web Server using Flask and RPi.	02
06	Any Application of IFTTT	02
07	Cloud Data Logging using Raspberry Pi and ThingSpeak.	02
08	Interfacing Soil Moisture Sensor with any board and Uploading Data to ThingSpeak Cloud. OR on AWS cloud	02
09	Any application of Blynk App	02
10	To study and implement IoT Data processing using Pandas	02
11	To control home devices (LED) using self-hosted page on Amazon AWS.	02
12	To send email alert using ESP32/nodeMCU via SMTP server.	02

Term Work: Term Work shall consist of at least 10 to 12 practicals' based on the above list. Also, Term work Journal must include at least 2 assignments.
Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)

Practical & Oral Exam: An Oral & Practical exam will be held based on the above syllabus

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2166122	Cyber Security Lab	-	2	-	-	1	-	1

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Practical/ Oral	Total
		Internal assessment			End Sem. Exam			
		IAT-I	IAT-II	IAT-I + IAT-II				
2166122	Cyber Security Lab	--	--	--	--	25	25	50

Lab Objectives:

1. To provide hands-on skills in identifying and analyzing cyberattack techniques.
2. To enable students to use modern tools for network, web, and mobile security testing.
3. To build practical knowledge of malware analysis, phishing simulation, and forensic tools.
4. To introduce AI/ML techniques for anomaly detection and cybersecurity automation.
5. To train students to secure digital payment systems, cloud systems, and social media accounts.
6. To promote ethical practices, safe online behavior, and understanding of cyber laws through practical scenarios.

Lab Outcomes:

1. Demonstrate understanding of cyberattacks through hands-on experiments.
2. Apply security tools (Wireshark, Nmap, Burp Suite, Autopsy) to analyze threats.
3. Conduct vulnerability tests on websites, networks, and applications.
4. Analyze malware, logs, and forensic artifacts using automated and AI-assisted tools.
5. Implement AI/ML models for detecting anomalies and phishing attacks.
6. Evaluate best practices for securing social media, digital payments, mobile devices, and cloud resources.

Prerequisite: Computer Network, Cryptography, Various Operating Systems, Linux, Windows, Android etc.

DETAILED SYLLABUS:

Sr. No.	Module	Detailed Content	LO Mapping
0	**Prerequisite**	Introduction to Kali Linux, Python basics for AI labs, setting up a cybersecurity lab environment (VMware/VirtualBox).	-
I	**Cybercrime Techniques & Fundamental Security Tools**	Understanding cybercrime techniques using practical demos. Footprinting, scanning, and reconnaissance. Packet capture basics using Wireshark. Hands-on Social Engineering simulation. Creating a simple phishing simulation safely **Self-learning Topics:** OSINT & Recon-ng, Wireshark Filters	LO1,LO2
II	**Attack Vectors & Mobile Security**	Network scanning using Nmap. Password cracking using John the Ripper. Mobile device vulnerability assessment. Man-in-the-middle (MITM) attack demonstration. Securing mobile devices & authentication mechanisms	LO2, LO3

		Self-learning Topics: Hydra basics, Android security architecture	
III	**Web, Network & Application Security**	SQL Injection on DVWA / Juice Shop, XSS, CSRF, Cookie stealing. Burp Suite hands-on - DoS simulation in a controlled environment & Steganography detection tools **Self-learning Topics:** Understanding OWASP Top 10	LO2, LO3
IV	**Digital Forensics & AI-Assisted Incident Response**	Windows & Linux log analysis, File system forensics using Autopsy, Memory forensics using Volatility, Network forensics (PCAP investigations), AI-based log anomaly detection, Malware behavior analysis sandbox **Self-learning Topics:** MITRE ATT&CK techniques, Introduction to ELK / Wazuh	LO4, LO5
V	**Social Media & Digital Payment Security**	Privacy configuration on social media platforms, Identifying fake profiles, scams & misinformation, Practical case studies on digital payment fraud, Securing UPI / Cards / Wallets, RBI guidelines practical interpretation **Self-learning Topics:** How WhatsApp / Instagram handle reports, Digital fraud case logs	LO5, LO6
VI	**AI/ML for Cybersecurity**	Building a simple ML model for anomaly detection, Email phishing classifier using Python, AI for vulnerability prediction (demo), Prompt injection attack testing on AI models, Hands-on with AI-driven cyber tools **Self-learning Topics:** Basics of data preprocessing for cybersecurity datasets, Kaggle cybersecurity datasets	LO5, LO6

Text Books:

1. William Stallings, Network Security Essentials: Applications and Standards, Pearson, 2020.
2. Nina Godbole, Cyber Security, Wiley India, 2011.
3. Soma Halder, Machine Learning for Cybersecurity, CRC Press, 2020.
4. Olivier Beg, *Hands-On Ethical Hacking and Network Defense*, Packt Publishing, 2019.
5. Chris Sanders, *Practical Packet Analysis: Using Wireshark to Solve Real-World Network Problems*, No Starch Press, 2017.

References:

1. OWASP Foundation, OWASP Testing Guide, Open Web Application Security Project (OWASP), 2014.
2. Michael Sikorski & Andrew Honig, Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software, No Starch Press, 2012.
3. Chris Hart, Digital Payment Security: Threats and Controls, Wiley, 2019.
4. Chet Hosmer, Digital Forensics with Open Source Tools, Syngress, 2011.

Online Resources:

Sr. No.	Website Name
1.	NPTEL: Practical Cyber Security for Cyber Security Practitioners: https://nptel.ac.in/courses/106104467
2.	https://tryhackme.com/
3.	https://attack.mitre.org/

List of Experiments.

Sr No	List of Experiments	Hrs
01	Wireshark-based packet analysis	2
02	Footprinting & reconnaissance using OSINT tools	2
03	Password cracking using John the Ripper	2
04	SQL Injection & XSS attacks on DVWA/Juice Shop	2
05	Burp Suite intercept & modify traffic	2
06	Mobile security & MITM demonstration	2
07	Steganography encoding & decoding	2
08	Digital forensics using Autopsy	2
09	Memory forensics using Volatility	2
10	AI-based anomaly detection model	2
11	Phishing detection ML classifier	2
12	Log analysis using ELK / open-source SIEM	2

List of Assignments (Any Two)

Sr No	List of Assignments / Tutorials	Hrs
01	Case study on cybercrime investigation	2
02	Report on social engineering attacks & prevention	2
03	Research on digital payment frauds	2
04	Documentation on securing social media accounts	2
05	Creating AI model documentation for anomaly detection	2
06	Report on cloud security basics	2

Assessment :

Term Work: Term Work shall consist of at least 10 to 12 practical's based on the above list. Also, Term work Journal must include at least 2 assignments.

Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)

Practical & Oral Exam: An Oral & Practical exam will be held based on the above syllabus

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2166123	Nature- inspired Optimization Algorithms Lab	-	2	-	-	1	-	1

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-I	IAT-II	IAT-I + IAT-II					
2166123	Nature-Inspired Optimization Algorithms Lab	-	-	-	-	-	25	25	50

Lab Objectives

The laboratory aims to:

1. Understand the working principles of classical and nature-inspired optimization algorithms.
2. Implement single and multi-objective evolutionary algorithms such as GA, PSO, and NSGA-II.
3. Explore swarm intelligence algorithms for solving real-world optimization and classification problems.
4. Develop hybrid and novel optimization models to enhance convergence and accuracy.
5. Apply metaheuristic algorithms to optimize machine learning and deep learning model parameters.
6. Evaluate the performance of optimization algorithms for practical engineering and AI-based applications.

Lab Outcomes

After successful completion of the lab, students will be able to:

1. Demonstrate the implementation and working of basic nature-inspired algorithms for optimization tasks.
2. Implement and analyze multi-objective optimization algorithms like NSGA-II and MOPSO.
3. Apply swarm-based algorithms (e.g., GWO, WOA) for feature selection and classification.
4. Design and test hybrid metaheuristic algorithms (e.g., GA-PSO) for faster and more robust convergence.
5. Integrate optimization algorithms with machine learning or edge computing systems for performance improvement.
6. Assess and compare the effectiveness of advanced optimization techniques in solving complex, real-world problems.

Suggested list of experiments

Sr. No.	Experiment Title	Mapped LO
1	Explain and demonstrate the functioning of a basic nature-inspired optimization algorithm (e.g., Genetic Algorithm or Particle Swarm Optimization) for solving simple optimization problems.	LO1

Sr. No.	Experiment Title	Mapped LO
2	Implementation of a Multi-Objective Genetic Algorithm (NSGA-II).	LO2
3	Multi-Objective Particle Swarm Optimization (MOPSO) for Energy-Delay Trade-off.	LO2
4	GWO algorithm to select optimal subsets of features for a supervised ML model (e.g., Logistic Regression or SVM).	LO3
5	Implementation of Whale Optimization Algorithm (WOA) for Solving Continuous Optimization Problems.	LO3
6	Design and test a hybrid GA-PSO algorithm for faster convergence on benchmark problems.	LO4
7	Apply metaheuristic optimization (GWO or PSO) for fake news or sentiment classification.	LO3, LO5
8	Application of Water Wave Optimization (WWO) Algorithm for Parameter Tuning in ML/DL Models.	LO5
9	Implementation of Swarm Cyborg Intelligence for Adaptive Optimization in Autonomous Systems.	LO5
10	Application of Ant Lion Optimizer (ALO) Algorithm for Feature Selection in Data Classification Tasks.	LO3, LO6
11	Implementation of Harris Hawks Optimization (HHO) Algorithm for Energy-Efficient Task Scheduling in Edge Computing Systems.	LO5, LO6
12	Crime Forecasting Using PSO-Based SVM Model for Predictive Policing Applications.	LO6

Books:

1. Nature-Inspired Optimization Algorithms, Xin-She Yang,
2. Introduction to Evolutionary Computing, A.E. Eiben and J.E. Smith.
3. Swarm Intelligence, James Kennedy et al.
4. Swarm Intelligence and Evolutionary Computation, Georgios N. Kouziokas.
5. Advanced Optimization by Nature-Inspired Algorithms, O. Bozorg-Haddad et al.
6. Swarm Intelligence by Kuldeep Singh Kaswan.
7. K. Deb, Multi-objective Optimization using Evolutionary Algorithms, Wiley, 2001.
8. Carlos Coello Coello, Gary B. Lamont, David A. van Veldhuizen, Evolutionary Algorithms for Solving Multi-Objective Problems, Springer, 2007
9. Seyedali Mirjalili, Evolutionary Algorithms and Metaheuristics in Engineering Applications, Springer, 2020.
10. Coello, C.A. et al., Evolutionary Algorithms for Solving Multi-Objective Problems, Springer, 2019.

Online References:

Sr. No.	Website Name
4.	https://onlinecourses.nptel.ac.in/noc25_me22
5.	https://www.youtube.com/watch?v=E2dJOYDz7b0

Suggested list of assignments

Assignment 1: questions should be based on first three chapters

Assignment 2 : questions should be based on chapter four to six

Term Work: Term Work shall consist of at least 10 labs based on the above list. Also, Term Work Journal must include at least 2 assignments as mentioned in above syllabus.
Term Work Marks: 25 Marks (Total marks) = 15 Marks (experiments) + 5 Marks (Assignments) + 5 Marks (Attendance)

Practical & Oral Exam: An Oral & Practical exam will be held based on the above syllabus

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2166124	Edge Computing Lab	-	2	-	-	1	-	1

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Practical/ Oral	Total
		Internal assessment			End Sem. Exam			
		IAT-I	IAT-II	IAT-I + IAT-II				
2166124	Edge Computing Lab	--	--	--	--	25	25	50

Lab Objectives:

1. To set up and configure a Linux environment, execute basic commands, manage files, and automate tasks using shell scripts for edge computing applications.
2. To understand network configurations, measure and analyze network performance parameters, and troubleshoot edge-to-cloud connectivity issues.
3. To deploy virtual servers on cloud platforms, host lightweight web services using Flask/FastAPI, and simulate cloud-based task offloading for edge applications.
4. To design and simulate edge and fog computing architectures using tools like iFogSim, configure edge nodes, and implement basic task-offloading strategies.
5. To simulate IoT sensor data, implement device-to-edge communication protocols such as MQTT and HTTP/WebSockets, and deploy lightweight ML models at the edge.
6. To compare edge vs cloud performance metrics, design simple dashboards for real-time monitoring, and optimize edge applications for resource-constrained environments.

Lab Outcomes:

1. Install and configure a Linux environment, execute command-line operations, and write shell scripts to automate tasks.
2. Analyze network performance metrics such as latency, bandwidth, and packet loss, and troubleshoot connectivity issues.
3. Deploy virtual servers, host Flask/FastAPI web services, and simulate cloud compute tasks for edge offloading.
4. Design and simulate edge/fog computing environments, configure edge nodes, and implement task-offloading strategies.
5. Simulate IoT sensor data, implement MQTT and HTTP/WebSocket communication, and deploy lightweight ML models at the edge.
6. Measure edge vs cloud performance, build basic dashboards for monitoring, and optimize edge applications.

DETAILED SYLLABUS:

Sr. No.	Module	Detailed Content	LO Mapping
0	**Prerequisite**	Knowledge of Programming, Computer Networks, Web Programming.	
I	**Linux Fundamentals for Edge Computing**	Introduction to Linux OS and distributions suitable for edge computing, Installation of Linux (desktop/server) and basic configuration, Command-line basics: executing commands, navigating directories, file handling, and permissions, Shell scripting essentials for automating tasks, Setting up a development environment for Python and IoT experimentation **Self-learning Topics:** Linux Networking Configurations and commands	LO 1
II	Networking Fundamentals for Edge Systems	Introduction to computer networks and IP addressing, Measuring network latency, packet loss, and throughput, Edge-to-cloud connectivity and network considerations, Network troubleshooting for edge devices. **Self-learning Topics: Various** Network Performance parameters	LO 2
III	Cloud Virtualization and Web Services	Creating and deploying virtual servers on cloud platforms (AWS, Azure, or GCP), Introduction to Flask and FastAPI for lightweight web services, Hosting a basic web service inside a virtual machine, Simulating cloud compute for edge offloading scenarios, Security basics for cloud-hosted services **Self-learning Topics: Exploring** various Cloud computing and Edge computing platforms	LO 3
IV	Edge and Fog Computing Architecture	Introduction to edge and fog computing concepts and use cases,,Simulating edge and fog computing environments using iFogSim or similar tools,,Configuring local machines as edge nodes,,Designing basic edge and fog architectures for IoT data processing,Task offloading strategies: local vs cloud execution **Self-learning Topics:** Comparative analysis between Cloud, Edge and Fog Computing platforms and Services	LO 4
V	IoT Integration & Communication Protocols	Simulating IoT sensor data (temperature, humidity, light intensity) with Python, Implementing MQTT with HiveMQ Cloud and Raspberry Pi, Device-to-edge communication using HTTP POST and WebSockets, Data collection, logging, and real-time transmission, Lightweight ML model deployment at the edge (moving average, anomaly detection, simple regression) **Self-learning Topics: Applications of AI in Edge Computing**	LO 5

VI	Edge Analytics, Performance, and Visualization	Comparing edge vs cloud performance: latency, bandwidth usage, processing time, Data storage and basic visualization techniques (CSV, console, HTML dashboards), Building simple edge dashboards for real-time monitoring, Optimizing edge applications for resource-constrained environments, Case studies and best practices in edge and fog computing. **Self-learning Topics:** Realtime applications in Edge and Fog Computing	LO 6

Text Books:

1. Rajkumar Buyya, Satish Narayana Srirama, "Fog and Edge Computing: Principles and Paradigms", Wiley, 1st Edition, 2019.
2. Wei Gao & Qi Zhang, "Machine Learning for Edge Computing", Wiley, 1st Edition, 2023
3. Peter Elger & Eóin Shanaghy, "AI as a Service: Serverless Machine Learning with AWS", O'Reilly, 2020
4. Joe Papa, "TinyML Cookbook: Over 80 Practical Recipes for Deploying ML on Microcontrollers and Edge Devices", O'Reilly, 2022

References

1. Chii Chang, Satish Narayana Srirama & Rajkumar Buyya, "Fog and Edge Computing: Principles and Paradigms", Wiley, 1st Edition, 2017.
2. Deepak Gupta & Aditya Khamparia, "Fog, Edge and Pervasive Computing in Intelligent IoT-Driven Applications", Wiley, 1st Edition, 2021.

Online Resources:

Sr. No.	Website Name
1.	https://github.com/Apress/AWS-IoT-With-Edge-ML-and-Cybersecurity
2.	NPTEL: Edge Computing: https://nptel.ac.in/courses/106104449
3.	Courseera: Implementing Security in IoT and Edge Devices: https://www.coursera .org/learn/security-at-the-edge-course-2

List of Experiments.

Sr No	List of Experiments	Hrs
01	Set up a basic Linux environment: installation, command execution, directory navigation, and file handling. (Foundation needed for Edge)	2
02	Network Basics for Edge: Test IP addressing, ping nodes, check latency, and explore edge–cloud connectivity using basic commands.	2
03	To create and deploy a virtual server on AWS	
04	Host a basic Flask / FastAPI web service inside the VM (simulating cloud compute).	2
05	Simulate basic IoT sensor data (temperature, humidity, light intensity) using a Python script.	2
06	Design and deploy an edge computing architecture using edge simulators such as ifogsim	2
07	Design and deploy a fog computing architecture using simulators such as iFogSim	2
08	To Implement MQTT using HiveMQ cloud and R-Pi	2
09	Configure a local machine as an Edge Node and deploy a small edge application.	2
10	Implement a basic task-offloading mechanism using conditional logic (local vs cloud execution).	2
11	Deploy a very lightweight ML model (moving average, anomaly detection, or simple regression) at the Edge.	2
12	Compare performance metrics (latency, bandwidth usage, processing time)	2

	between edge and cloud execution.	
13	Implement device-to-edge communication using HTTP POST / WebSockets.	2
14	Build a simple data visualization or logging dashboard on the edge node (CSV / console / HTML).	2

List of Assignments (Based on Theory Syllabus):

Sr No	List of Assignments / Tutorials	Hrs
	Assignment 1	
01	Explain the limitations of cloud computing in supporting low-latency and real-time applications. How do fog computing and edge computing address these challenges?	
02	Describe the architecture of an IoT Edge system. Explain the roles of sensors, gateways, communication interfaces, and cloud integration in the data flow of an IoT ecosystem.	2
03	Discuss the hardware architecture of Raspberry Pi, highlighting its GPIO pinout, interfaces, and supported peripherals. Explain how SSH and VNC are used for remote access and configuration.	
	Assignment 2	
01	Explain the working principle of the MQTT protocol. Discuss its architecture, workflow, and why MQTT is preferred for lightweight IoT communication compared to HTTP and CoAP.	
02	What makes Raspberry Pi suitable as an edge computing device? Explain with reference to low-latency processing, data acquisition, and the Edge–Fog–Cloud hierarchy.	2
03	Describe any two real-world applications of edge computing in domains such as smart cities, healthcare, industrial automation, or transportation. Explain how edge nodes improve system efficiency.	

Assessment :
Term Work: Term Work shall consist of at least 10 to 12 practicals' based on the above list. Also, Term work Journal must include at least 2 assignments.
Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)

Practical & Oral Exam: An Oral & Practical exam will be held based on the above syllabus

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2166125	**Digital Imaging Techniques and Analysis Lab**	**-**	**2**	**-**	**-**	1	**-**	**1**

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Practical/ Oral	Total
		Internal assessment			End Sem. Exam			
		IAT-I	IAT-II	IAT-I + IAT-II				
2166125	**Digital Imaging Techniques and Analysis Lab**	--	--	--	--	25	25	50

Lab Objectives:

1. To introduce students to image acquisition, representation, and basic operations using OpenCV.
2. To develop skills in applying spatial and frequency domain techniques for image enhancement.
3. To provide practical experience in noise modeling, filtering, and image restoration.
4. To apply segmentation, morphological, and feature extraction methods for image analysis.
5. To understand and evaluate lossy and lossless image compression techniques.
6. To solve real-world problems using digital image processing techniques.

Lab Outcomes:

1. Perform image acquisition, representation, and basic processing using OpenCV.
2. Apply spatial and frequency domain techniques to enhance and restore images.
3. Analyze and reduce different types of noise using appropriate filtering methods.
4. Implement segmentation, morphological, and feature extraction techniques for image analysis.
5. Apply and compare lossy and lossless image compression techniques.
6. Solve real-world problems in domains such as medical imaging, satellite imaging, and machine vision using image processing methods.

List of Experiments.

Sr No	List of Experiments	LO Mapping
01	Study of image acquisition and basic operations using OpenCV.	LO1
02	Implementation of image enhancement using point processing techniques.	LO2
03	Perform histogram equalization and contrast stretching on grayscale images.	LO2
04	Apply smoothing and sharpening spatial filters.	LO2
05	Implement frequency domain filtering using Fourier Transform.	LO2
06	Study and remove noise using mean, median, and adaptive filters.	LO2, LO3
07	Perform geometric transformations: scaling, rotation, and translation.	LO3
08	Apply edge detection operators (Sobel, Prewitt, Canny).	LO4
09	Implement image segmentation using thresholding (global and adaptive).	LO4

10	Perform morphological operations (dilation, erosion, opening, closing).	LO4
11	Extract color, texture, and shape features from images.	LO4
12	Implement image compression using Huffman coding / RLE /DCT.	LO5,
13	Apply lossy and lossless compression techniques and compare performance.	LO5,LO6
14	Apply a complete image analysis pipeline to real-world data (Cat and dog classification, License Plate Number recognition, Facial expression and emotion recognition).	LO6

Online Resources:

Sr. No.	**Website Name**
1.	Virtual Lab https://cse19-iiith.vlabs.ac.in/List%20of%20experiments.html
2.	https://nptel.ac.in/courses/117105135
3.	https://nptel.ac.in/courses/108103174
4.	https://nptel.ac.in/courses/106105216

Sr No	List of Assignments / Tutorials	CO Mapping
01	Assignment covers the topics from the first three units.	CO1,CO2,CO3
02	Assignment covers the topics from the last three units.	CO4,CO5,CO6

Assessment :

Term Work: Term Work shall consist of at least 10 to 12 practical's based on the above list. Also, Term work Journal must include at least 2 assignments.
Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)

Practical & Oral Exam: An Oral & Practical exam will be held based on the above syllabus

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2166126	**Blockchain Technology Lab**	**-**	**2**	**-**	**-**	**1**	**-**	**1**

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Practical/ Oral	Total
		Internal assessment			End Sem. Exam			
		IAT-I	IAT-II	IAT-I + IAT-II				
2166126	**Blockchain Technology Lab**	--	--	--	--	25	25	25

Lab Objectives:

1. Set up local and testnet blockchain environments.
2. Develop secure smart contracts using Solidity.
3. Deploy contracts on Ethereum and interact using Web3 frameworks.
4. Deploy chain code on Hyperledger Fabric.
5. Build a full-fledged DApp or cross-chain prototype.
6. Explore Layer-2 scaling or interoperability bridges.

Lab Outcomes:

1. To install, configure, and operate a local blockchain development environment using tools such as Ganache, Truffle, and Hardhat to compile, deploy, and test basic smart contracts.
2. To develop and analyze Solidity smart contracts using OpenZeppelin libraries, applying secure coding principles to identify and prevent vulnerabilities like reentrancy and unauthorized access.
3. To create and deploy ERC-20/721/1155 tokens on Ethereum testnets (Goerli/Sepolia) and verify smart contracts using tools such as Etherscan and Web3/Ethers libraries.
4. To design, deploy, and evaluate chaincode in Hyperledger Fabric by creating networks, channels, and endorsement policies, and performing invoke/query operations using CLI and SDK.
5. To build and integrate a blockchain-enabled front-end application using React and Web3/Ethers.js, enabling wallet connectivity, smart contract interactions, and event handling.
6. To design and develop a complete blockchain application including smart contract/chaincode, UI integration, deployment, documentation, and GitHub repository.

DETAILED SYLLABUS:

Sr. No.	Module	Detailed Content	LO Mapping
0	**Prerequisite**	Computer Network: Concepts like P2P networks, IP addressing, ports, client-server architecture. Cryptography and security: Hash functions, public/private keys, digital signatures, encryption.	
I	Local Blockchain Environment Setup & Basics	● Install & configure development tools: Node.js, VS Code, Git ● Set up local blockchain: Ganache (GUI + CLI), Truffle initialization, Hardhat project setup ● Compile, deploy, and test a simple “Hello Blockchain” smart contract ● Write basic unit tests using Truffle/Hardhat	LO 1

		● Interact with local blockchain using scripts (JS or Python) **Self-learning Topics:** Deploy and Compare local blockchain simulators: Ganache vs Hardhat Node vs Anvil (Foundry).	
II	**Smart Contract Development with Solidity and OpenZeppelin**	● Solidity basics: state variables, functions, modifiers, events, mappings ● Inheritance, libraries, interfaces ● Using OpenZeppelin for secure contract development ● Implement access control & secure coding practices ● Write unit tests to avoid common vulnerabilities like Reentrancy : Integer overflow/underflow and Unauthorized access **Self-learning Topics:** Study how AI tools like ChatGPT, CodeQL are now used for smart contract audits.	LO 2
III	**Token Development & Deployment on Testnets**	● Create ERC-20, ERC-721, ERC-1155 tokens using OpenZeppelin ● Mint & transfer tokens locally ● Connect to Ethereum Testnets: Goerli/Sepolia ● Use Infura/Alchemy for RPC access ● Deploy via Truffle/Hardhat to testnet ● Interaction with Web3.js or Web3.py scripts ● Verify contracts on Etherscan **Self-learning Topics:** Explore the Automated NFT metadata generation using AI (e.g., DALL-E, Midjourney) and upload to IPFS.	LO 3
IV	**Hyperledger Fabric – Permissioned Chaincode Development**	● Install Fabric binaries & prerequisites (Docker, Go, Fabric Samples) ● Launch local Fabric network using test-network ● Create channels, peers, and certs ● Write chaincode in Go/Node.js ● Invoke & query chaincode via CLI and SDK ● Endorsement policies & private data collections **Self-learning Topics:** Explore the Hyperledger Aries & Indy for Decentralized Digital Identity.	LO 4
V	**DApp Integration & Frontend (React/Node + Web3/Ethers)**	● Connect front-end with blockchain using Ethers.js/Web3.js ● Wallet connection using MetaMask ● Build React UI to interact with smart contracts ● Read/write blockchain data from UI ● Handle events & notifications **Self-learning Topics:** Build a chatbot-based DApp assistant using AI (LLM) that reads contract ABI & guides users how to use the DApp	LO 5
VI	**Capstone Mini-Project**	● Build a complete blockchain project using tools learned. ● Students choose any one track:	LO 6

		• Ethereum Dapp, NFT/Metaverse App Hyperledger Enterprise App and Layer-2 Dapp • Deliverables: • Working project with UI & smart contract/chaincode, GitHub repo and Report **Self-learning Topics:** Explore and Implement AI in blockchain projects	

Text Books:

1. Andreas M. Antonopoulos and Gavin Wood, Mastering Ethereum: Building Smart Contracts and DApps, O'Reilly Media.
2. Elad Elrom, The Blockchain Developer: A Practical Guide for Designing, Implementing, Publishing, Testing, and Securing Distributed Blockchain-based Projects, Apress.
3. Mayukh Mukhopadhyay, Ethereum Smart Contract Development: Build Blockchain-based Decentralized Applications Using Solidity, Packt Publishing.
4. Xun (Brian) Wu, Zhihong Zhang, and Liang Wang, Blockchain and Ethereum Smart Contract Solution Development: DApp Programming with Solidity, Apress.
5. Samanyu Chopra, Blockchain Developer's Guide: Develop Smart Applications with Blockchain Technologies – Ethereum, JavaScript, Hyperledger Fabric, and Corda, Packt Publishing.

References:

1. Roberto Infante, Building Ethereum DApps, Manning Publications.
2. Narayan Prusty, Building Blockchain Projects: Building Decentralized Blockchain Applications with Ethereum and Solidity, Packt Publishing.

Online Resources:

Sr. No.	Website Name
1.	Solidity (smart-contract language) — official docs: https://docs.soliditylang.org/
2.	OpenZeppelin (library for secure smart contracts) — docs: https://docs.openzeppelin.com/
3.	Hardhat (Ethereum dev tool) — tutorial: https://hardhat.org/tutorial
4.	Hyperledger Fabric (permissioned blockchain) — chaincode dev: https://hyperledger-fabric.readthedocs.io/en/release-1.4/chaincode4ade.html

List of Experiments.

Sr No	List of Experiments	Hrs
01	Write a program for Creating a block in Blockchain using Java/Python	2
02	Installation & Setup of Local Ethereum Development Environment: Install Node.js, VS Code, Git, MetaMask, Truffle, Hardhat, Ganache. Create first Hardhat project & compile a simple contract.	2
03	Deploy & Interact with Smart Contract on Local Blockchain: Deploy a sample contract using Truffle/Hardhat + Ganache. Write scripts to read/write data. View transactions in Ganache.	2
04	Build a Basic Solidity Smart Contract (CRUD Contract) : Functions, modifiers, events, data types, mappings. Test using Hardhat tests.	2
05	Secure Smart Contract Development using OpenZeppelin: Add Ownable, Pausable, AccessControl; prevent reentrancy & data leaks.	2
06	Smart Contract Testing & Debugging: Unit tests (Chai/Mocha), coverage report, Hardhat console debugging, fixing security issues.	2
07	Create & Test ERC-20 Token: Write ERC-20 token contract using OpenZeppelin; mint, transfer, burn tokens locally.	2
08	Create & Mint ERC-721 NFT: Create NFT with metadata/IPFS; mint NFT; view on OpenSea Testnet.	2

09	Deploy Smart Contracts on Testnet (Goerli/Sepolia): Connect via Infura/Alchemy, verify on Etherscan, interact using Web3.js or Web3.py.	2
10	Hyperledger Fabric Network Setup & Chaincode Deployment: Install Fabric samples, set up test network, deploy sample asset chaincode.	2
11	Build a Simple DApp: Frontend + Smart Contract Interaction	2

List of Assignments (Based on Theory Syllabus):

Sr No	List of Assignments / Tutorials	Hrs
	Assignment 1 (Any two Questions from each Unit)	
01	**Unit I: Introduction to Blockchain & Block Design** **Q1:** Explain the structure of a blockchain block and the role of the block header. **Q2:** What is a Merkle Tree, and why is it used in blockchain? **Q3:** Differentiate between a blockchain and a traditional database.	1
02	**Unit II: Consensus Mechanisms & Blockchain Platforms** **Q1:** Describe the Proof-of-Work (PoW) consensus algorithm. **Q2:** Compare two blockchain platforms of your choice in terms of features and consensus mechanisms. **Q3:** What is a 51% attack, and how can it affect blockchain security?	1
03	**Unit III: Ethereum & Smart Contracts** **Q1:** What is the role of Gas in Ethereum transactions? **Q2:** List the basic components of a Solidity smart contract. **Q3:** Explain the working of an Ethereum transaction from sender to miner validation.	1
	Assignment 2 (Any two Questions from each Unit)	
01	**Unit IV: Private and Consortium Blockchains** **Q1:** Differentiate between private and consortium blockchains. **Q2:** What is the role of MSPs in Hyperledger Fabric? **Q3:** Briefly explain Paxos and Raft consensus mechanisms.	1
02	**Unit V: Cryptocurrencies and Digital Tokens** **Q1:** What are the differences between ERC-20 and ERC-721 tokens? **Q2:** Explain the double spending problem in blockchain. **Q3:** List two common security threats in blockchain and their mitigation strategies	1
03	**Unit VI: Emerging Trends of AI in Blockchain and Case Studies** **Q1:** How can AI be integrated with blockchain for fraud detection? **Q2:** What are AI-enabled smart contracts, and how do they enhance automation? **Q3:** Give one example of a real-world application where AI-based consensus is used.	1

Assessment :
Term Work: Term Work shall consist of at least 10 to 12 practicals' based on the above list. Also, Term work Journal must include at least 2 assignments.
Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)

Practical & Oral Exam: An Oral & Practical exam will be held based on the above syllabus

Letter Grades and Grade Points:

Semester GPA/ Programme CGPA Semester/ Programme	% of Marks	Alpha-Sign/ Letter Grade Result	Grading Point
9.00 - 10.00	90.0 – 100	O (Outstanding)	10
8.00 - < 9.00	80.0 - < 90.0	A+ (Excellent)	9
7.00 - < 8.00	70.0 - < 80.0	A (Very Good)	8
6.00 - < 7.00	60.0 - < 70.0	B+ (Good)	7
5.50 - < 6.00	55.0 - < 60.0	B (Above Average)	6
5.00 - < 5.50	50.0 - < 55.0	C (Average)	5
4.00 - < 5.00	40.0 - < 50.0	P (Pass)	4
Below 4.00	Below 40.0	F (Fail)	0
Ab (Absent)	-	Ab (Absent)	0

Sd/-

Dr. Subhash K. Shinde
BoS-Chairman-Computer Engineering
Faculty of Technology

Sd/-

Dr. Deven Shah
Associate Dean
Faculty of Science & Technology

Sd/-

Prof. Shivram S. Garje
Dean
Faculty of Science & Technology

AC –
Item No. –

As Per NEP 2020

University of Mumbai

Syllabus for Major Vertical – 1, 4 & 5		
Name of the Programme – B.E. (Electronics and Computer Engineering)		
Faulty of Engineering		
Board of Studies in Electronics and Computer Engineering		
U.G. Third Year Programme	**Exit Degree**	**B.Sc. Tech Engineering- Electronics and Computer Engineering**
Semester		**V & VI**
From the Academic Year		**2026-27**

University of Mumbai

(As per NEP 2020)

Sr. No.	Heading	Particulars
1	**Title of program** **O:____________________**	**B.E. (Electronics and Computer Engineering)**
2	**Exit Degree**	**B.Sc. Tech Engineering- Electronics and Computer Engineering**
3	**Scheme of Examination** **R: ____________**	NEP 40% Internal 60% External, Semester End Examination Individual Passing in Internal and External Examination
4	**Standards of Passing** **R: ______________**	**40%**
5	**Credit Structure** **R: ______________**	**Attached herewith**
6	**Semesters**	**Sem. V & VI**
7	**Program Academic Level**	**5.5**
8	**Pattern**	**Semester**
9	**Status**	**New**
10	**To be implemented from Academic Year**	**2026-27**

Sd/-
Dr. R.N.Awale
BoS-Chairman-Electronics and Computer Science
Faculty of Technology

Sd/-
Dr. Deven Shah
Associate Dean
Faculty of Science & Technology

Sd/-
Prof. Shivram S. Garje
Dean
Faculty of Science & Technology

Preamble

To meet the challenge of ensuring excellence and NEP 2020 policy in engineering education, the issue of quality needs to be addressed, debated, and taken forward systematically. Accreditation is the principal means of quality assurance in higher education. The major emphasis of the accreditation process is to measure the outcomes of the program that is being accredited. In line with this Faculty of Science and Technology (in particular Engineering) of the University of Mumbai has taken the lead in incorporating the philosophy of NEP 2020 education in the process of curriculum development.

The third-year engineering course consists of a core and elective training program to impart scientific and logical thinking training to learners in general, with a choice of course selection from the program core course, program elective course, multidisciplinary minor, IKS, PCE and vocational skill-enhanced course. Simultaneously, the objectives of NEP 2020 demand nurturing the core program, elective program, communication skills and skills required for the Electronics and Computer Engineering Branch of engineering in the learner. Keeping this in view, a pool of courses is offered in Core Courses covering fundamentals required to understand core and modern engineering practices and emerging trends. Elective courses are offered to the learners to understand the concepts of emerging technology. This will help with the holistic development of learners through interdisciplinary skills development and mobility among other departments as real-time solutions. Considering the shift in pedagogy and the convenience of a stress-free learning process, a choice-based subject pool is offered in the coursework under the heads of Electronics and Computer Engineering for open electives and multidisciplinary minor courses in sem V and VI. Essentially, to give a glimpse of trends in the industry under vocational and enhanced skill practices, the pool is offered to nurture and develop creative skills in contemporary industrial practices. Criteria met in the structure is the opportunity for learners to choose the course of their interest in all disciplines.

Program Core Course and Program Elective Courses Cover Electronics and Computer Engineering core and elective courses. Also, MDM is where a pool of subjects is given for selection. In addition to this learner will have IKS and PCE courses. Considering the present scenario, diverse choices need to be made available to fulfil the expectation of a learner to aspire for a career in the field of current trends of Technology and interdisciplinary research. The IKS course will be program-specific, which can be achieved in Undergraduate training by giving an objective viewpoint to the learning technology, from ascent time and how it was learned and the process and transitioning a learner from a rote learner to a creative professional. For this purpose, PCE is introduced in the First Semester to orient a communication skill to become skilled professionals for placement and higher education entrance exams. Considering the NEP-2020 structure of award of Certificate after U.G. First Year, U. G. Diploma after the Second Year and B.Sc. Tech Engineering after the Third Year at multiple exit-point pools of Vocational skills, is arranged to give exposure to the current Industry practices.

The faculty resolved that course objectives and course outcomes are to be clearly defined for every core and elective course so that all faculty members in affiliated higher education institutes understand the depth and approach of the course to be taught to the learners, which will enhance the learners' learning process. NEP 2020 grading system enables a much-needed shift in focus from teacher-centric to continuous-based learner-centric education since the workload estimated is based on the investment of time in learning and not in teaching. It also focuses on the holistic development of the learners through mobility and continuous evaluation, which will enhance the quality of education. Credit assignment for courses is based on a 15-week teaching-learning process for NEP 2020; however, the content of courses is to be taught in 12-13 weeks, and the remaining 2-3 weeks are to be utilized for revision, self-learning, tutorial, guest lectures, coverage of content beyond the syllabus, etc.

There was a concern that in the present system, the third-year syllabus must not be heavily loaded for the learner and it is of utmost importance that the learner entering the third year of an engineering course should feel at ease by lowering the burden of syllabus and credits. This is necessary for a learner to get accustomed to the new environment of a college and to create a bond between the teacher and the learner. The present curriculum will be implemented for the Third Year of Engineering from the academic year 2026-27. Subsequently, this system will be carried forward for Final Year Engineering in the academic years 2027-28, respectively.

Sd/-
Dr. R. N. Awale
BoS-Chairman-Electronics and Computer Science
Faculty of Technology

Sd/-
Dr. Deven Shah
Associate Dean
Faculty of Science & Technology

Sd/-
Prof. Shivram S. Garje
Dean
Faculty of Science & Technology

B.Sc. Tech Engineering- Electronics and Computer Engineering

Credit Structure (Sem. V & VI)

Level	Semester	Major		Minor	OE	VSC, SEC (VSEC)	AEC, VEC, IKS	OJT, FP, CEP, CC, RP	Cum. Cr. / Sem.	Degree/ Cum. Cr.
	R: ________________C									
		Mandatory	**Electives**							
5.5	V	PCC501:3 PCC502:3 PCC503:3 PCL501: 1 PCL502:1 PCL503:1	PEC501:3 PEL501:1	MDC:3 MDL:1	OE:2	--	IKS: 2	--	24	B.Sc. Tech Engineering 48
	R:_________________D									
	VI	PCC601:3 PCC602:3 PCL601:1 PCL602:1	PEC601:3 PEC602:3 PEL601:1 PEL602:1	MDC: 3 MDL:1	--	VSEC:2	AEC: 2	--	24	
	Cum Cr.	20	12	8	2	2	*2+2*	--	48	

Exit option: Award of UG Diploma in Major and MDM with 90 credits and additional 4 credits core **one** theory subject with 3 credits and **one** lab with 1 credit from one third year from where they want to take Exit degree. Along with theory and practical course student must compulsory do internship for **one month or 160 hours** which internship is equal to 4 credits.

[Abbreviation - OE – Open Electives, VSC – Vocation Skill Course, SEC – Skill Enhancement Course, (VSEC), AEC – Ability Enhancement Course, VEC – Value Education Course, IKS – Indian Knowledge System, OJT – on Job Training, FP – Field Project, CEP – Continuing Education Program, CC – Co-Curricular, RP – Research Project]

Sem. - V

T.E. Electronics and Computer Engineering Scheme

Program Structure for Third Year of Electronics and Computer Engineering

UNIVERSITY OF MUMBAI (With Effect from 2026-2027)

SEMESTER V

Course Code	Course Description	Teaching Scheme (Contact Hours)			Credit Assigned			
		Theory	Practical	Tutorial	Theory	Tutorial	Practical	Total Credits
2275111	Control System	3	--	--	3	--	--	3
2275112	Computer Network	3	–	--	3	--	--	3
2275113	Software Engineering	3	--	--	3	--	--	3
2275114	PEC-1	3	--	--	3	--	--	3
2275211	Embedded System and RTOS	3	--	--	3	--	--	3
2275115	Control System Lab	--	2	--	--	--	1	1
2275116	Computer Network Lab	--	2	--	--	--	1	1
2275117	Software Engineering Lab	--	2	--	--	--	1	1
2275118	PEL-1 lab.	--	2	--	--	--	1	1
2275212	Embedded System and RTOS Lab	--	2@	--	--	--	1	1
2275311	Industrial Automation	2#	--	---	2	--	--	2
2275511	IKS as per Groups	--	2*+2	--	--	--	2	2
Total		**17**	**14**	**00**	**17**	**00**	**07**	**24**

* Two hours of practical class to be conducted for full class as demo/discussion.

Theory / Tutorial 1 credit for 1 hour and Practical 1 credit for 2 hours

Institute shall offer a course for Open Elective from Science/Commerce/Management stream bucket provided by the University of Mumbai.

@ Institute shall offer a course for MDM from other Engineering Boards Group.

Course Code	**Course Description**
22751141	Semiconductor Devices and Characteristics
22751142	Machine Learning
22751143	DBMS
22751144	DATA COMPRESSION
Lab Code	**Lab Description**
22751181	Semiconductor Devices and Characteristics
22751182	Machine Learning
22751183	DBMS
22751184	DATA COMPRESSION

course Code	Course Description	Examination scheme							
		Internal Assessment Test (IAT)			End Sem. Exam Marks	End Sem. Exam Duration (Hrs)	Term Work (Tw)	Oral & Pract.	Total
		IAT-I	IAT-II	Total (IAT-I) + IAT-II)					
2275111	Control System	20	20	40	60	2	--	--	100
2275112	Computer Network	20	20	40	60	2	--	--	100
2275113	Software Engineering	20	20	40	60	2	--	--	100
2275114	PEC-1	20	20	40	60	2	--	--	100
2275211	Embedded System and RTOS	20	20	40	60	2	--	--	100
2275115	Control System Lab	--	--	--	--	--	25	--	25
2275116	Computer Network Lab	--	--	--	--	--	25	25	50
2275117	Software Engineering Lab	--	--	--	--	--	25	25	50
2275118	PEL-1 lab.	--	--	--	--	--	25	25	50
2275212	Embedded System and RTOS Lab	--	--	--	--	--	25	25	50
2275311	Industrial Automation	--	--	--	--	--	50	--	50
2275511	IKS as per Groups						50	--	50
Total		**100**	**100**	**200**	**300**	**10**	**225**	**125**	**825**

Program Structure for Third Year of Electronics and Computer Engineering

UNIVERSITY OF MUMBAI (With Effect from 2026-2027)

SEMESTER VI

Course Code	Course Description	Teaching Scheme (Contact Hours)			Credit Assigned			
		Theory	Practical	Tutorial	Theory	Tutorial	Practical	Total Credits
2276111	Analog and Digital Communication	3	--	--	3	--	--	3
2276112	Artificial Intelligence	3	--	--	3	--	--	3
2276113	PE-2(Program Elective) *	3	--	--	3	--	--	3
2276114	PE-3(Program Elective) *	3	--	--	3	--	--	3
2276211	IOT	3	--	--	3	--	--	3
2276411	Mini-Project – II	--	4	--	--	--	2	2
2276611	Data Structures	--	2*+2	--	--	--	2	2
2276115	Analog and Digital Communication Lab	--	2	--	--	--	1	1
2276116	Artificial Intelligence Lab	--	2	--	--	--	1	1
2276117	PEL-2	--	2	--	--	--	1	1
2276118	PEL-3	--	2	--	--	--	1	1
2276212	IOT Lab	--	2@	--	--	--	1	1
Total		**15**	**18**	**00**	**15**	**00**	**09**	**24**

* Two hours of practical class to be conducted for full class as demo/discussion.

Theory / Tutorial 1 credit for 1 hour and Practical 1 credit for 2 hours

@ Institute shall offer a course for MDM from other Engineering Boards.

Course Code	**Course Description**
22761131	ASIC
22761132	Natural Language Processing
22761133	ADBMS
22761134	NETWORK AND SECURITY
22761141	MEMS & Automation
22761142	Signals and Image Processing
22761143	Cryptography
22761144	Computer Communication Network
Lab Code	**Lab Description**
22761171	ASIC
22761172	Natural Language Processing
22761173	ADBMS
22761174	NETWORK AND SECURITY
22761181	MEMS & Automation
22761182	Signals and Image Processing
22761183	Cryptography
22761184	Computer Communication Network

Course Code	Course Description	Examination scheme							
		Internal Assessment Test (IAT)			End Sem. Exam Marks	End Sem. Exam Duration (Hrs)	Term Work (Tw)	Oral & Pract.	Total
		IAT-I	IAT-II	Total (IAT-I) + IAT-II)					
2276111	Analog and Digital Communication	20	20	40	60	2	--	--	100
2276112	Artificial Intelligence	20	20	40	60	2	--	--	100
2276113	PE-2(Program Elective) *	20	20	40	60	2	--	--	100
2276114	PE-3(Program Elective) *	20	20	40	60	2	--	--	100
2276211	IOT	20	20	40	60	2	--	--	100
2276411	Mini-Project – II	--	--	--	--	--	50	--	50
2276611	Data Structures	--	--	--	--	--	25	--	25
2276115	Analog and Digital Communication Lab	--	--	--	--	--	25	25	50
2276116	Artificial Intelligence Lab	--	--	--	--	--	25	25	50
2276117	PEL-2	--	--	--	--	--	25	25	50
2276118	PEL-3	--	--	--	--	--	25	25	50
2276212	IOT Lab	--	--	--	--	--	25	25	50
Total		**100**	**100**	**200**	**300**	**10**	**200**	**125**	**825**

Vertical – 1
Major

Course Code	Course Name	Theory (Hrs.)	Practical (Hrs.)	Tutorial (Hrs.)	Theory (Credits)	Practical/ Oral (Credits)	Tutorial (Credits)	Total (Credits)
2275111	Control System	3	--	--	3	--	--	3

Course Code	Course Name	Examination Scheme							
		Theory Marks				Term Work	Practical	Oral	Total
		Internal assessment			End Sem. Exam				
		1	2	Average					
2275111	Control System	40	40	40	60	—	---	----	100

Prerequisite: Applied Mathematics (Laplace Transform, Ordinary differential equations), Applied Physics, Basic Electrical Engineering

Course Objectives:

1. To calculate the mathematical models of a given system using block diagram reduction rules and Mason's gain formula.

2. To provide fundamental concepts of control system such as Time response and Frequency response.

3. To develop concepts of stability and its assessment criteria.

4. Design performance specification-based controller for a given system.

5. To develop the concept of stability and its assessment for linear-time-invariant systems.

6. To introduce the design of controllers in frequency-domain and state-space.

Course Outcomes: After successful completion of the course students will be able to

1. Determine the mathematical models of a given system using block diagram reduction rules and Mason's gain formula.

2. Analyze the performance of first order and second order control systems based on the time domain and frequency domain specifications.

3. Evaluate the stability of the control systems in time domain and frequency domain.

4. Design performance specification-based controller for a given system.

5. Analyze the control systems using state-space methods and design state feedback controllers.

6. Design performance specification-based controller for a given system.

DETAILED THEORY SYLLABUS:

Sr. No.	Module	Detailed Content	Hrs.
1.	**Introduction to Control System**	Examples of control systems; open and closed loop systems; Servomechanism. Mathematical modeling of Electrical & Mechanical Systems; Transfer function model. Block diagram and Signal Flow Graph (SFG) representation of control systems; Block diagram reductions; Mason's gain formula.	**08**
2	**Time Response Analysis**	Standard test input signals; time response of first and second order systems for standard test inputs; Transient response specification for second order system; Error constants and type of the system. Effect of Damping, Time Response Specifications: Delay time, Rise time, Peak Time, Peak Overshoot, settling time, Numerical Problems.	**08**
3	**Stability Analysis in Frequency Domain**	Stability in time domain: The concept of stability, necessary conditions for stability, Hurwitz stability criterion, Routh stability criterion, relative stability analysis. Root locus Analysis: Root locus concept, general rules for constructing root-locus, root locus analysis of control system.	**07**
4	**Frequency Response Analysis**	Introduction to frequency response; Frequency response plots: Polar plot and Bode plot; Stability margins in frequency domain, Nyquist stability criterion and stability analysis using Nyquist plot.	**07**
5	**Introduction to Controller Design and State-space Analysis**	Characteristics of feedback: Sensitivity to parametric variation; Disturbance rejection; Steady-state accuracy. Feedback controller design using Root-locus; Reshaping the root-locus; Cascade lead, lag and lag-lead compensator. Concept of state variables; State-space model; Canonical forms; Conversion between canonical forms using similarity transforms. Solution of state-space equation; Eigen-values and eigenvectors; Stability in state-space; Concept of controllability and observability.	**08**
6	**Controller Design in State-space**	State-feedback controller design: Pole-placement method; Ackerman's formula.	**04**

Theory Assessments:

1. Internal Assessment: Two Internal assessments will be conducted for 40 marks each with average marks of both assessments as final score.

2. End Sem Theory Examination:

- Question paper will consist of 4 questions, each carrying 20 marks.
- Total 3 questions need to be solved.
- Q.1 will be compulsory, based on the entire syllabus.
- Remaining questions will be randomly selected from all the modules.

Weightage of marks should be proportional to the number of hours assigned to each module.

Books:

1. K. Ogata, “Modern Control Engineering”, PHI, New Delhi

2. I. J. Nagrath, M. Gopal, “Control System Engineering”, 5th edition, New Age International Publishers

Reference Books:

1. Golnaraghi Farid, B. C. Kuo, “Automatic Control Systems”, 10th edition, McGraw Hill, 2017.

2. K. Ogata, “Modern Control Engineering”, 6th edition, Prentice Hall, 2010.

3. I.J. Nagrath, M. Gopal, “Control System Engineering”, New Age International, 2009.

4. Norman Nise, “Control Systems Engineering”, Wiley, 8th edition, 2019.

<table>
<tr><th rowspan="2">Course Code</th><th rowspan="2">Course Name</th><th colspan="3">Teaching Scheme (Hrs/week)</th><th colspan="4">Credits Assigned</th></tr>
<tr><td>L</td><td>T</td><td>P</td><td>L</td><td>T</td><td>P</td><td>Total</td></tr>
<tr><td rowspan="4">2275115</td><td rowspan="4">Control Systems Lab</td><td>--</td><td>--</td><td>2</td><td>--</td><td>--</td><td>1</td><td>1</td></tr>
<tr><td colspan="7">Examination Scheme</td></tr>
<tr><td colspan="2">Term work</td><td colspan="3">Orals</td><td colspan="2">Total</td></tr>
<tr><td colspan="2">25</td><td colspan="3">25</td><td colspan="2">50</td></tr>
</table>

COURSE OBJECTIVES:

1.To study the system performance using time domain analysis and methods for improving it.
2.To assess the system performance using frequency domain analysis and techniques for improving the performance to design various controllers and compensators to improve system performance.
3.To design various controllers and compensators to improve system performance.
4.To study the characteristics of servo mechanisms which are helpful in automatic control systems.
5.To understand the different ways of system representations such as transfer function representation and state space representations and to assess the system dynamic response.

COURSE OUTCOMES:

At the end of the course, students would be able to:

1. Determine the performance and time domain specifications of first order systems.
2. Analyze various time domain and frequency domain techniques to assess the system performance.
3. Study the system performance by selecting a suitable controller and/or a compensator for a specific application.
4. Get the knowledge about the effect of poles and zeros location on transient and steady state behavior of second order systems and can implement them to practical systems.
5. Study system controllability and observability using state space representation and applications of state space representation to various systems.

Laboratory Experiments:

PART-A

The following experiments are required to be conducted compulsory experiments:
1. Time response of Second order system
2. Characteristics of Synchros
3. Effect of feedback on DC servomotor
4. Transfer function of DC motor
5. Transfer function of DC generator
6. Temperature controller using PID
7. Characteristics of AC servomotor
8. Effect of P, PD, PI, PID Controller on a second order systems

PART-B

In addition to the above eight experiments, at least any two of the experiments from the Following list are required to be conducted.

9. Lag and lead compensation – Magnitude and phase plot
10. Programmable logic controller – Study and verification of truth tables of logic gates, Simple Boolean expressions and application of speed control of motor.
11. a) Simulation of P, PI, PID Controller.
12. b) Linear system analysis (Time domain analysis, Error analysis) using suitable software
13. Stability analysis (Bode, Root Locus, Nyquist) of Linear Time Invariant system using Suitable software
14. State space model for classical transfer function using suitable software – Verification.
15. Design of Lead-Lag compensator for the given system and with specification using Suitable software

Laboratory Assessment:

Assessment:

Term Work: Term Work shall consist of at least 10 to 11 practicals' based on the above list. Also, Term work Journal must include at least 2 assignments.

Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)

Practical/ Oral Exam: An Oral examination will be held based on the above syllabus.

Term Work:

At least 10 experiments covering the entire syllabus of should be set to have well predefined inference and conclusion. The experiments should be student centric and attempts should be made to make the experiments meaningful and interesting. Experiments must be graded from time to time. The grades should be converted into marks as per the Credit and Grading System manual and should be added and averaged. The grading and term work assessment should be done based on this scheme. The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work. Practical and Oral exam will be based on the entire syllabus.

Note:

Suggested List of Experiments is indicative. However, flexibility lies with individual course instructors to design and introduce new, innovative and challenging experiments, (limited to maximum 30% variation to the suggested list) from within the curriculum, so that the fundamentals and applications can be explored to give greater clarity to the students and they can be motivated to think differently.

<table>
<tr><td rowspan="2">Course Code</td><td rowspan="2">Course Name</td><td colspan="3">Teaching Scheme (Hrs/week)</td><td colspan="4">Credits Assigned</td></tr>
<tr><td>L</td><td>T</td><td>P</td><td>L</td><td>T</td><td>P</td><td>Total</td></tr>
<tr><td rowspan="5">2275112</td><td rowspan="5">Computer Network</td><td>3</td><td>--</td><td>2</td><td>3</td><td>--</td><td>1</td><td>4</td></tr>
<tr><td colspan="7">Examination Scheme</td></tr>
<tr><td></td><td>IA1</td><td>IA2</td><td colspan="2">ESE</td><td colspan="2">Total</td></tr>
<tr><td>Theory</td><td>20</td><td>20</td><td colspan="2">60</td><td colspan="2">100</td></tr>
</table>

<table>
<tr><td>Pre-requisite Course Codes</td><td colspan="2">PC303 Computer Organisation and Architecture</td></tr>
<tr><td rowspan="7">Course Outcomes</td><td colspan="2">After the successful completion students should be able to</td></tr>
<tr><td>CO1</td><td>Describe the architecture, services, and protocol hierarchies of computer networks.</td></tr>
<tr><td>CO2</td><td>Explain physical layer transmission media and demonstrate the role of network devices in communication.</td></tr>
<tr><td>CO3</td><td>Analyze data link layer design issues and apply error detection, flow control, and MAC protocols.</td></tr>
<tr><td>CO4</td><td>Compute IP addressing schemes and evaluate routing algorithms for efficient network design.</td></tr>
<tr><td>CO5</td><td>Differentiate between TCP and UDP services and implement flow and congestion control mechanisms.</td></tr>
<tr><td>CO6</td><td>Illustrate application layer protocols and develop client-server communication scenarios.</td></tr>
</table>

<table>
<tr><td>Module No.</td><td>Unit No.</td><td>Topics</td><td>Reference</td><td>Hrs.</td></tr>
<tr><td rowspan="4">Module 1</td><td>1</td><td>Introduction to Data Communications and Networking</td><td rowspan="4">T1, T2, R1, R4</td><td rowspan="4">6Hrs</td></tr>
<tr><td>1.1</td><td>Introduction to computer networks, Network software, Layers and services, Network topologies, protocol hierarchies, design issues for the layers, connection oriented and connectionless services.</td></tr>
<tr><td>1.2</td><td>Reference models: Layer details of OSI, TCP/IP models. Communication between layers. Internet</td></tr>
<tr><td>1.3</td><td>Addressing: physical/logical/port addressing/socket addressing.</td></tr>
<tr><td>Module</td><td>2</td><td>Physical Layer</td><td>T1, T2,</td><td>5</td></tr>
</table>

2	2.1	Guided Transmission Media: Twisted pair, Coaxial, Fiber optics.	R2, R3	Hrs
	2.2	Unguided media (Wireless Transmission): Radio Waves, Microwave, Bluetooth, Infrared, Circuit and Packet Switching.		
	2.3	Network Devices: Repeaters, Hubs, Switches, Routers and Gateways		
Module 3	3	Data Link Layer	T1, T2, R1, R4	8Hrs
	3.1	DLL Design Issues - Services, Framing, Error Control, Flow Control, Error Detection and Correction, Elementary Data Link Protocols, Stop and Wait, Sliding Window - Go Back N, Selective Repeat.		
	3.2	Medium Access Control sublayer: Channel Allocation problem, Multiple access Protocol (Aloha, Carrier Sense Multiple Access (CSMA/CD), Local Area Networks - Ethernet (802.3), Introduction to wireless LAN: 802.11x		
Module 4	4	Network Layer	T1, T2, R1, R2, R4	10 hrs
	4.1	Network Layer design issues, Communication Primitives: Unicast, Multicast, Broadcast. Network Layer Protocols: IPv4 Datagram Format, IPv4 Addresses, IPv4 Addressing (classfull and classless), Sub-netting and Super-netting design problems, IPv4 Protocol, IPv6 Packet Format, IPv6 Addressing, Transition from IPv4 to IPv6.		
	4.2	Routing algorithms: Intra-domain Routing -Shortest Path, Distance Vector Algorithms, Link State Routing, Inter-domain Routing Protocols.		
	4.3	Introduction to ICMP, ARP, RARP		
Module 5	5	Transport Layer	T1, T2, R1, R4	6 hrs
	5.1	Connectionless and Connection–oriented services at transport layer, Transmission Control Protocol (TCP): TCP Services, TCP Segment, TCP three-way handshake.		
	5.2	User datagram Protocol (UDP), UDP Services, UDP Datagram		
	5.3	TCP Flow control, error control and congestion control		
Module 6	6	Application Layer	T1, T2, R1	4 hrs
	6.1	Application layer Paradigms, Client-Server Paradigm: Application Programming Interface		
	6.2	Introduction to Application layer Protocols: HTTP, FTP, DNS, SMTP, TELNET, SSH, DHCP.		
Total				39

Course Assessment:

Theory:

IA1: 20 Marks written examination for one hour
IA2: 20 Marks written examination for one hour
ESE: 60 Marks written examination for two hours

Recommended Books:

Textbooks:

1. Andrew S Tanenbaum, Computer Networks -, 5th Edition, Pearson Education
2. Behrouz A. Forouzan, Forouzan Mosharrat, Computer Networks A Top down Approach, McGraw Hill education
3. Ranjan Bose, Information Theory, Coding and Cryptography, Ranjan Bose, Tata McGraw Hill, Second Edition

Reference books:

1. James F. Kurose, K. W. Ross, Computer Networking: A Top-Down Approach Featuring the Internet, 3rd Edition, Pearson Education.
2. S. Keshav, An Engineering Approach to Computer Networks, 2nd Edition, Pearson Education.
3. W. A. Shay, Understanding Communications and Networks, 3rd Edition, W. A. Shay, Cengage Learning.
4. L. L. Peterson and B. S. Davie, Computer Networks: A Systems Approach, 4th Ed, Elsevier India

Online References:

1. NPTEL – Computer Networks (Prof. Bhaskaran Raman / Prof. Rajesh Sundaresan) https://nptel.ac.in/courses/106/105/106105081
2. Cisco Networking Academy – Introduction to Networks (Free Modules) https://www.netacad.com/courses/networking
3. RFC Repository for Internet Standards https://www.rfc-editor.org
4. IETF (Internet Engineering Task Force) – Protocol Standards and drafts https://www.ietf.org

<table>
<tr><td rowspan="6">**Course Code**

2275116</td><td rowspan="6">**Course Name**

Computer Network Lab</td><td colspan="3">**Teaching Scheme (Hrs/week)**</td><td colspan="4">**Credits Assigned**</td></tr>
<tr><td>L</td><td>T</td><td>P</td><td>L</td><td>T</td><td>P</td><td>Total</td></tr>
<tr><td>--</td><td>--</td><td>2</td><td>--</td><td>--</td><td>1</td><td>1</td></tr>
<tr><td colspan="7">Examination Scheme</td></tr>
<tr><td colspan="2">Term work</td><td colspan="3">Orals</td><td colspan="2">Total</td></tr>
<tr><td colspan="2">25</td><td colspan="3">25</td><td colspan="2">50</td></tr>
</table>

Pre-requisite Course Codes		
Laboratory Objectives	1.	To analyze network protocols and packet structures using tools like Wireshark.
	2.	To practice basic networking commands and physical cable preparation for LAN connectivity.
	3.	To configure and compare network topologies and hardware components using simulation tools.
	4.	To implement static and dynamic routing protocols for efficient network communication.
	5.	To perform subnetting, IP addressing, and socket programming for data transmission.
	6.	To configure firewalls and routing tables for secure network operations
Laboratory Outcomes	After the successful completion students should be able to	
	LO1	Analyze TCP/IP stack and interpret protocol headers and encapsulation.
	LO2	Execute networking commands and prepare/test UTP cables for LAN connectivity.
	LO3	Configure and compare network topologies and hardware components using Cisco Packet Tracer.
	LO4	Implement static and dynamic routing protocols (RIP, OSPF, BGP, EIGRP) and verify routing tables.
	LO 5	Perform subnetting, configure IP addressing schemes, and develop socket programs for TCP/UDP.
	LO 6	Configure firewalls and manage routing tables for secure network operations.

Laboratory Experiments:

Sr. No.	Title of experiment	Hardware /Software	Module	Reference
1.	Analysis of TCP/IP Stack Using Packet Capturing Tool (Wireshark)	Software	1	1, 5
2.	Study and Execution of Basic Networking Commands in Linux	Software	2	2, 7
3.	Configuration and Comparison of Network Topologies Using Cisco Packet Tracer	Software	2	2, 3, 6
4.	Configuration and Comparison of Network Hardware Components Using Cisco Packet Tracer	Software	3	2, 3, 6
5.	Configuration of Static Routing in a Network Using Cisco Packet Tracer	Software	3	3, 6
6	Configuration of Distance Vector Routing Protocol (RIP)	Software	3	3, 6
7	Configuration of Link State Routing Protocol (OSPF)	Software	4	3, 6
8	Configuration of Path Vector Routing Protocol (BGP)	Software	4	3, 6
9.	Implementation of Subnetting and IP Addressing	Software	4	1, 2, 6
10.	Implementation of Socket Programming Using TCP and UDP	Software	4	1, 4
11.	Configuration of Firewall Using IP Tables	Software	5	7

Laboratory Assessment:
Assessment:
Term Work: Term Work shall consist of at least 10 to 11 practicals’ based on the above list. Also, Term work Journal must include at least 2 assignments.
Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)
Practical/ Oral Exam: An Oral examination will be held based on the above syllabus.

Recommended Books:

1. Computer Networking: A Top-Down Approach, Pearson Education.
   (Useful for TCP/IP, socket programming, protocol analysis)
2. Hands-On Networking Fundamentals, Cengage Learning.
   (Useful for practical networking concepts and network devices)
3. Network Simulator Experiments Manual
   (Useful for simulation-based networking experiments)
4. TCP/IP Sockets in C, Morgan Kaufmann.
   (Useful for TCP/UDP socket programming experiments)
5. Practical Packet Analysis, No Starch Press.
   (Useful for Wireshark and packet analysis experiments)

6. CCNA 200-301 Official Cert Guide, Cisco Press.
   (Useful for routing, switching, RIP, OSPF, BGP practicals)
7. Linux Firewalls, Novell Press.
   (Useful for IPTables and firewall configuration)

Online Resources:

· https://www.netacad.com/9tut.com – CCNA Tutorials & Labs

Term Work:

At least 10 experiments covering the entire syllabus of PCL 502 (Computer Network Lab) should be set to have well predefined inference and conclusion. The experiments should be student centric and attempts should be made to make the experiments meaningful and interesting. Experiments must be graded from time to time. The grades should be converted into marks as per the Credit and Grading System manual and should be added and averaged. The grading and term work assessment should be done based on this scheme. The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work. Practical and Oral exam will be based on the entire syllabus.

Note:

Suggested List of Experiments is indicative. However, flexibility lies with individual course instructors to design and introduce new, innovative and challenging experiments, (limited to maximum 30% variation to the suggested list) from within the curriculum, so that the fundamentals and applications can be explored to give greater clarity to the students and they can be motivated to think differently.

Course Code	Course Name	Teaching Scheme (Hrs/week)			Credits Assigned			
2275113	Software Engineering	L	T	P	L	T	P	Total
		2	--	--	2	--	1	3
		Examination Scheme						
			IA1	**IA2**	**ESE**		**Total**	
		Theory	20	20	60		100	

Pre-requisite Course	1.Knowledge of Software Application Domains, Software Engineering Practices. 2. Knowledge of any Programming Language	
	After the successful completion students should be able to	
Course Outcomes	CO1	To learn basics of software engineering & software development process models, agile software development & other agile practices.
	CO2	To Identify, Specify, analyse Software Requirements and prepare models.
	CO3	To understand concepts and principles of software design and Development.
	CO4	To learn about Project Scheduling concept and Software Cost Estimation Techniques.
	CO5	To understand the concept of software quality assurance and Risk Management.
	CO6	To learn different software testing strategies and tactics.

Module No.	Unit No.	Topics	Reference	Hrs.
Module 1	**1**	**Introduction to Software Engineering and Process Models**		6Hrs
	1.1	Nature of Software, Software Process framework	1,2	
	1.2	Prescriptive Models: Waterfall Model, Incremental, RAD Models Evolutionary Process Models: Prototyping, Spiral and Concurrent Development Model. Specialized Models: Component based		
	1.3	Agile process, Agility Principles, Extreme Programming (XP), Scrum.		

Module 2	**2**	**Requirement Engineering and Modelling**		8Hrs
	2.1	Types of Requirements, Requirement Engineering Task, Software Requirement Specification (SRS), Developing Use Cases (UML)	1,2	
	2.2	Requirement Model: Scenario-based model, Class-based model, Behavioural model.		
Module 3	**3**	**Design Engineering**	1,2 3	6 Hrs
	3.1	Design Process, Software design principles, Design Concepts, Characteristics of Good Design,Effective Modular Design – Cohesion and Coupling.		
	3.2	Software Architecture, architectural Genres, user interface design principles.		
Module 4	**4**	**Project scheduling & Cost Estimation**		8 Hrs
	4.1	Project Scheduling, defining a Task Set for the Software Project, Gantt charts, Program Evaluation Review Techniques (PERT), Tracking the Schedule	1,2 ,3	
	4.2	Software Project Estimation, Decomposition Techniques, LOC based, FP based and Use case-based estimations, Empirical estimation Models. COCOMO II Model.		
Module 5	**5**	**Software Risk &vQuality Management**		
	5.1	Software Risk, Types of Risk, Risk Identification, Risk Assessment, Risk Projection, RMMM	1,2,3	5Hrs
	5.2	Software Quality Assurance Task and Plan, McCall’s Quality Factors, Software Reliability, Formal,Technical Review (FTR), Walkthrough		
Module 6	**6**	**Software Testing Strategies and Tactics**		6Hrs
	6.1	Software Testing Fundamentals, Testing strategies for conventional and Object-Oriented architectures, Unit testing, Integration testing, System Testing, Validation and System Testing	1,2,3	
	6.2	Testing Tactics: White-Box Testing, Basis Path Testing, Control Structure Testing, Black-Box Testing		
Total				**39**

Course Assessment:

Theory:

IA1: 20 Marks written examination for one hour
IA2: 20 Marks written examination for one hour

ESE: 60 Marks written examination for two hours

Recommended Books:

[1] Roger S Pressman "Software Engineering: A Practitioner's Approach" 8th Edition McGraw-Hill.
[2] Rajib Mall, "Fundamentals of Software Engineering", Prentice Hall India.
[3] Pankaj Jalote, "An integrated approach to Software Engineering", Springer/Narosa

Online References:
NPTEL courses on Software Engineering

Course Code	**Course Name**	**Teaching Scheme (Hrs/week)**			**Credits Assigned**			
2275117	Software Engineering Lab	L	T	P	L	T	P	Total
		--	--	2	--	--	1	1
		Examination Scheme						
		Term work		Orals			Total	
		25		25			50	

Pre-requisite Course Codes	**Nil**	
Laboratory Objectives	1.	To understand and apply software process models
	2	To develop software requirement and design models using UML
	3.	To perform software project planning and estimation techniques
	4.	To implement software quality assurance and testing techniques
Laboratory Outcomes	After the successful completion students should be able to	
	LO1	Identify requirements and apply process models to selected case studies.
	LO2	Analyse and design models for the selected case study using UML modeling
	LO3	Use various Software Engineering and Project Management Tools
	LO4	Create project schedule, estimate budget and RMMM plan
	LO 5	Design test cases for achieving software quality.

Laboratory Experiments:

Sr. No.	Title of experiment	Hardware /Software	Module	Reference
1.	Compare various SDLC models and justify the selection of an appropriate model to solve a given problem statement.	Software	1	1,2,3
2.	Develop a basic Software Requirement	Software	2	1,2,3

	Specification (SRS) for a specified software application			
3.	Design Use case diagrams for given problem statements using star UML.	Software	2	1,2,3
4.	Design DFD diagram for given problem statement using online tools like star UML.	Software	2	1,2,3
5.	Develop Activity / State Transition diagram for the project.	Software	2	1,2,3
6.	Develop Class diagram /sequence diagram for the project.	Software	2	1,2,3
7.	Prepare a schedule for the project using any project management tool like project Libre .	Software	4	1,2,3
8.	Perform Cost Estimation of Software.	Software	4	1,2,3
9.	Prepare a RMMM plan for the project.	Software	5	1,2,3
10.	Develop test cases for the project using white box testing.	Software	6	1,2,3
11.	Design test cases for the project using black box testing .	Software	6	1,2,3
12	Develop a small working prototype based on a selected case study.	Software	1,2,3,4,5,6	1,2,3

Laboratory Assessment:
Assessment:
Term Work: Term Work shall consist of at least 10 to 12 practicals' based on the above list. Also, Term work Journal must include at least 2 assignments.
Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)
Practical/ Oral Exam: An Oral examination will be held based on the above syllabus.

Recommended Books:

[1] Ian Sommerville, "Software Engineering", Pearson Education (9th edition)
[2] Jibitesh Mishra and Ashok Mohanty, "Software Engineering", Pearson edition
[3] Hans Van Vliet "Software Engineering Principles and Practice" 3rd edition Wiley.

Term Work:

At least 10 experiments covering the entire syllabus of PCL 602 (Software Engineering Lab) should be set to have well predefined inference and conclusion. The experiments should be student centric and attempts should be made to make the experiments meaningful and interesting. Experiments must be graded from time to time. The grades should be converted into marks as per the Credit and Grading System manual and should be added and averaged. The grading and term work assessment should be done based on this scheme. The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work. Practical and Oral exam will be based on the entire syllabus.

Note:

Suggested List of Experiments is indicative. However, flexibility lies with individual course instructors to design and introduce new, innovative and challenging experiments, (limited to maximum 30% variation to the suggested list) from within the curriculum, so that the fundamentals and applications can be explored to give greater clarity to the students and they can be motivated to think differently.

Course Code	Course Name	Teaching Scheme (Hrs/week)			Credits Assigned			
		L	T	P	L	T	P	Total
2276111	Analog and Digital Communication	3	--	2	3	--	1	4
		Examination Scheme						
			IA1	IA2	ESE		Total	
		Theory	20	20	60		100	

Pre-requisite Course Codes	PCC 201X Digital Electronics**,** PCC 301 Signals and Systems, PCC 302 Electronic Devices, PCC402 Analog Electronics	
Course Outcomes	After the successful completion students should be able to	
	CO1	Explain fundamentals of communication systems, noise, and modulation techniques.
	CO2	Analyze AM, FM, and PM modulation and demodulation systems.
	CO3	Apply sampling, pulse modulation, and multiplexing techniques in communication systems.
	CO4	Explain source coding and channel coding methods for reliable communication.
	CO5	Analyze baseband and digital bandpass transmission techniques and their performance.
	CO6	Compare communication and modulation techniques for practical applications.

Module No.	Unit No.	Topics	Reference	Hrs.
Module 1	**1**	**Introduction to Communication Systems**	1,2,3,4,6	5 Hrs
	1.1	Electromagnetic Spectrum, Block diagram of analog and digital communication system, Need for modulation		
	1.2	Types of Noise, Signal-to-noise ratio, Noise factor, Noise Figure, Noise Temperature, Friis Formula		
Module 2	**2**	**Analog Modulation and Demodulation Techniques**		
	2.1	**Amplitude Modulation and Demodulation-** Mathematical and Graphical Representation AM wave, Voltage distribution and Power Calculations. Basic Concepts of DSBSC and SSB. AM Diode Detectors	1,2,3,4,6	10 Hrs
	2.2	**Frequency and Phase Modulation and Demodulation** - Mathematical Representation of FM wave, deviation ratio, bandwidth requirement, Narrowband and Wideband FM, Pre-emphasis and De-emphasis, Noise-triangle, FM Generation (Varactor Diode and Armstrong method) , FM Detectors- Foster-Seeley Detector, Phase Modulation, Applications and Comparison of AM, FM and PM		
	2.3	**Radio Receivers** – Characteristics of radio receivers, Super heterodyne AM and FM receivers, AGC		
Module 3	**3**	**Pulse Modulation and Multiplexing Techniques**		
	3.1	Sampling Theorem, Nyquist Criteria, Sampling Techniques, Aliasing error and Aperture effect	2,3,4,6,7	7 Hrs
	3.2	Generation and Detection of PAM, PWM and PPM. PCM Transmitter and Receiver. Concepts of Delta modulation (DM) and Adaptive Delta Modulation (ADM).		

	3.3	Need of Multiplexing, Block Diagram explanation of TDM and FDM Systems		
Module 4	**4**	**Source Coding and Channel Coding**	4,5,6,7	6 Hrs
	4.1	Basics of Information Theory, Entropy, Shannon Hartley theorem, Source Coding Techniques: Huffman coding and Shannon-Fano coding		
	4.2	Channel Coding for Error Detection and Correction - Linear Block Codes, Cyclic Codes and Convolutional Codes		
Module 5	**5**	**Baseband Transmission and Reception**	4,5,6,7	4 Hrs
	5.1	Block diagram of baseband Transmitter-Receiver system, Need of line codes, Properties, Types of line codes - RZ and NRZ Unipolar formats, RZ and NRZ Polar formats, RZ and NRZ Bipolar format (AMI format), Split phase Manchester format and Polar Quaternary formats		
	5.2	Inter Symbol Interference, Inter Channel Interference, Introduction to Matched Filter and Equalizer		
Module 6	**6**	**Bandpass Transmission and Reception**	4,5,6,7	7 Hrs
	6.1	Generation, Detection, Error probability, Power Spectral Density, Euclidean distance and Bandwidth of BASK, BFSK, BPSK		
	6.2	Generation, Detection, Error probability, Power Spectral Density, Euclidean distance and Bandwidth of QPSK, offset QPSK, M-ary PSK, 16-ary QASK and MSK.		
Total				**39**

Course Assessment:

Theory:

IA1: 20 Marks written examination for one hour

IA2: 20 Marks written examination for one hour

ESE: 60 Marks written examination for two hours

Recommended Books:

[1] Kennedy and Davis "Electronics Communication System", Tata McGraw Hill

[2] Wayne Tomasi, "Electronics Communication Systems" Pearson Education, 5th Edition.

[3] Herbert Taub, Donald L Schilling, Goutam Saha, "Principles of Communication Systems", Tata McGraw Hill, 3rd Edition.

[4] T. L. Singal, "Analog and Digital Communication," Tata Mc-Graw Hill, New Delhi, First Ed.

[5] Sklar B, and Ray P. K., "Digital Communication: Fundamentals and Applications," Pearson, Dorling Kindersley (India), Delhi, Second Edition

[6] Simon Haykin, "Communication System", John Wiley and Sons,4th Ed.

[7] B P Lathi, Zhi Ding, Modern Digital and Analog Communication Systems, Oxford University. 4th Ed.

[8] Beharouz A. Forouzan, "Data Communication and Networking", fourth edition

Online References:

1. Analog communication- https://swayam.gov.in/nd1_noc20_ee69/preview
2. Principles of Digital Communication- https://nptel.ac.in/courses/108/101/108101113/
3. Principles of Digital Communication- https://nptel.ac.in/courses/108/102/108102120/

<table>
<tr><th rowspan="1">Course Code</th><th>Course Name</th><th colspan="3">Teaching Scheme (Hrs/week)</th><th colspan="4">Credits Assigned</th></tr>
<tr><td rowspan="6">2276115</td><td rowspan="6">Analog and Digital Communication Lab</td><td>L</td><td>T</td><td>P</td><td>L</td><td>T</td><td>P</td><td>Total</td></tr>
<tr><td>--</td><td>--</td><td>2</td><td>--</td><td>--</td><td>1</td><td>1</td></tr>
<tr><td colspan="7">Examination Scheme</td></tr>
<tr><td colspan="2">Term work</td><td colspan="2">Oral and Practical Exam</td><td colspan="3">Total</td></tr>
<tr><td colspan="2">25</td><td colspan="2">25</td><td colspan="3">50</td></tr>
</table>

<table>
<tr><td>Pre-requisite Course Codes</td><td colspan="2"></td></tr>
<tr><td rowspan="4">Laboratory Objectives</td><td>1.</td><td>To implement analog, pulse, and digital modulation and demodulation techniques.</td></tr>
<tr><td>2</td><td>To study multiplexing, source coding, and channel coding techniques.</td></tr>
<tr><td>3.</td><td>To analyze various line coding methods used in communication systems.</td></tr>
<tr><td>4.</td><td>To use hardware and simulation tools for communication system analysis.</td></tr>
<tr><td rowspan="7">Laboratory Outcomes</td><td colspan="2">After the successful completion students should be able to</td></tr>
<tr><td>LO1</td><td>Implement various analog modulation and demodulation techniques.</td></tr>
<tr><td>LO 2</td><td>Implement pulse modulation, demodulation, and multiplexing techniques.</td></tr>
<tr><td>LO 3</td><td>Implement digital modulation and demodulation techniques.</td></tr>
<tr><td>LO 4</td><td>Analyze performance characteristics of communication systems and coding techniques.</td></tr>
<tr><td>LO 5</td><td>Analyze different line coding techniques used in digital communication.</td></tr>
<tr><td>LO 6</td><td>Simulate source coding and channel coding techniques for error control and data transmission.</td></tr>
</table>

Laboratory Experiments:

Sr. No.	Title of Experiment	Hardware / Software	Module	Reference
1	Generation and Detection of DSB-SC Signal	Hardware/Software	2	1,3,4
2	Generation and Detection of Frequency Modulation	Hardware/Software	2	1,3,4
3	Design and Implementation of Pre-emphasis and De-emphasis Circuit	Hardware/Software	2	1,3,6
4	Study of Superheterodyne Receiver Characteristics and AGC	Hardware/Software	3	1,2,3
5	Verification of Sampling Theorem	Hardware/Software	3	1,2,3
6	Generation and Detection of PAM Signals	Hardware/Software	3	1,2,3
7	Generation and Detection of PWM and PPM Signals	Hardware/Software	3	1,2,3,4
8	Generation and Detection of PCM Signals	Hardware/Software	3	1,2,3,4
9	Study of Delta Modulation and Adaptive Delta Modulation Techniques	Hardware/Software	3	1,2,4
10	Study of Time Division Multiplexing and De-multiplexing	Hardware/Software	3	1,2,4
11	Study of Frequency Division Multiplexing and De-multiplexing	Hardware/Software	4	4,5,7
12	Simulation of Shannon-Fano Coding and Calculation of Code Efficiency	Hardware/Software	4	4,5,7
13	Study of Huffman Coding and Calculation of Code Efficiency	Hardware/Software	4	4,5,7
14	Study of Linear Block Code and Analysis of Error Detection Capability	Hardware/Software	4	4,5,7
15	Study of Convolutional Code for Given Specifications	Hardware/Software	5	4,5,7
16	Observation and Comparison of Various Line Codes	Hardware/Software	5	5,6,7
17	Study of Matched Filter Response for a Given Input Signal	Hardware/Software	6	4,5,7
18	Study of Modulation and Demodulation of Binary ASK, FSK, and PSK	Hardware/Software	6	4,5,7
19	Study of Modulation and Demodulation of QPSK	Hardware/Software	6	4,5,7
20	Study of Modulation and Demodulation of MSK	Hardware/Software	6	4,5,7

Laboratory Assessment:

Assessment:

Term Work: Term Work shall consist of at least 10 to 12 practical's based on the above list. Also, Term work Journal must include at least 2 assignments.

Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)

Practical/ Oral Exam: An Practical and Oral examination will be held based on the above syllabus.

Term Work:

At least 10-12 experiments covering the entire syllabus should be set to have well predefined inference and conclusion. This must include **50%** Hardware and **50%** Simulation experiments. The experiments should be student centric and attempts should be made to make the experiments meaningful and interesting. Experiments must be graded from time to time. The grades should be converted into marks as per the Credit and Grading System manual and should be added and averaged. The grading and term work assessment should be done based on this scheme. The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work. Practical and Oral exam will be based on the entire syllabus.

Note:

Suggested List of Experiments is indicative. However, flexibility lies with individual course instructors to design and introduce new, innovative and challenging experiments, (limited to maximum 30% variation to the suggested list) from within the curriculum, so that the fundamentals and applications can be explored to give greater clarity to the students and they can be motivated to think differently.

Course Code	Course Name	Teaching Scheme (Hrs/week)			Credits Assigned			
2276112	Artificial Intelligence	L	T	P	L	T	P	Total
		3	--	--	3	--	-	3
		Examination Scheme						
			IA1	**IA2**	**ESE**		**Total**	
		Theory	20	20	60		100	

Pre-requisite Course Codes	Fundamentals of mathematics including probability and statistics, Understanding of data structures and algorithms	
Course Outcomes	After the successful completion students should be able to	
	CO1	Explain the fundamental concepts, scope, types, applications and current trends of Artificial Intelligence.
	CO2	Identify the characteristics of the environment and differentiate between various agent architectures
	CO3	Apply the most suitable search strategy to design problem solving agents.
	CO4	Represent a natural language description of statements in logic and apply the inference rules to design Knowledge Based agents.
	CO5	Apply probabilistic reasoning and Bayesian approaches to handle uncertainty in AI systems.
	CO6	Explain modern AI learning paradigms, Generative AI, LLMs, XAI and ethical aspects of AI systems.

Module No.	**Unit No.**	**Topics**	**Reference**	**Hrs.**
Module 1	**1**	**Introduction to Artificial Intelligence and Intelligent Agents**	1,2,3	5Hrs
	1.1	Definition, scope and history of Artificial Intelligence (AI), Different perspectives of AI, Scope of AI in modern computing and engineering.		
	1.2	Types of AI: Weak AI vs Strong AI, Narrow AI and General AI, Reactive machines, Super AI, Applications of AI, Current trends in AI.		
Module 2	**2**	**Intelligent Agents and Environment**	1,2,3,5,6	5 Hrs
	2.1	Definition of an intelligent agent, components of an agent, PEAS representation (Performance measure, Environment, Actuators and Sensors), Concept of rationality, types of agents: simple reflex, model-based, goal-based and utility-based agents, Learning Agent		
	2.2	Characteristics of environments: fully observable vs partially observable, deterministic vs stochastic, episodic vs sequential, static vs dynamic, discrete vs continuous, Competitive vs collaborative.		

Module 3	**3**	**Problem Solving & Search Techniques**	1,2,3,4,5	9Hrs
	3.1	State space representation: concept of problem solving in AI, definition of state space, representation using state space trees and graphs, examples: 8-puzzle and route finding. Performance measures of search algorithms: completeness, optimality, time complexity and space complexity		
	3.2	Uninformed search techniques: Breadth First Search (BFS), Depth First Search (DFS), Depth Limited Search (DLS), Iterative Deepening Search (IDS) and Uniform Cost Search (UCS). Informed search techniques: Greedy Best First Search, A* algorithm, heuristic functions and admissibility. Local search techniques: hill climbing and simulated annealing.		
	3.3	Game playing: minimax algorithm and alpha-beta pruning.		
Module 4	**4**	**Knowledge Representation & Reasoning**	4,5	8 Hrs
	4.1	Knowledge representation schemes, propositional logic (PL): basics of logic and statements, syntax and semantics, logical connectives, truth tables, conversion to conjunctive normal form (CNF). First Order Predicate Logic (FOPL): syntax and semantics, predicates, constants, variables, functions and quantifiers, representation of real-world knowledge.		
	4.2	Inference rules in FOPL, inference techniques: forward chaining, backward chaining and resolution in FOPL.		
Module 5	**5**	**Uncertainty & Probabilistic Reasoning**	3,4,5	5 Hrs
	5.1	Handling uncertain knowledge, random variables, joint, conditional and marginal probability, Bayes' theorem, concept of prior, likelihood and posterior probability.		
	5.2	Bayesian Belief Networks: Mathematical formulation and reasoning in belief networks.		
Module 6	**6**	**Modern AI Systems and Learning**	6,7,8,9	7 Hrs
	6.1	Concept of learning in AI, Overview of machine learning and deep learning, Concepts of Supervised, Unsupervised, Semi - Supervised Learning, Reinforcement Learning, Ensemble Learning.		
	6.2	Conversational AI: chatbots and virtual assistants, Generative AI and foundation models, Basics of prompt engineering, Overview of Large Language Models (LLMs)		
	6.3	Explainable AI (XAI): need for explainability, interpretability and transparency, Ethical issues, bias, privacy and limitations of modern AI systems		
			Total	**39**

Course Assessment:

Theory:

IA1: 20 Marks written examination for one hour
IA2: 20 Marks written examination for one hour

ESE: 60 Marks written examination for two hours

Recommended Books:

1. Stuart J. Russell and Peter Norvig, "Artificial Intelligence: A Modern Approach,Second Edition" Pearson Education.
2. Elaine Rich and Kevin Knight —Artificial Intelligence, Third Edition, Tata McGraw-Hill Education Pvt. Ltd., 2008.
3. George F Luger "Artificial Intelligence" Low Price Edition, Pearson Education., Fourth edition.
4. D. W. Patterson, Artificial Intelligence and Expert Systems, Prentice Hall.
5. Saroj Kaushik "Artificial Intelligence", Cengage Learning.
6. Deep Learning – Ian Goodfellow, Yoshua Bengio, Aaron Courville
   (Comprehensive deep learning reference, widely considered foundational)
7. Machine Learning for Absolute Beginners: A Plain English Introduction Book by Oliver. Theobald.
8. A Beginner's Guide to Generative AI: An Introductory Path to Diffusion Models, ChatGPT, and LLMs
9. Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI

Online References:

1. NPTEL – Artificial Intelligence by Prof. Deepak Khemani (IIT Madras)
   https://nptel.ac.in/courses/106106126.
2. NPTEL – Introduction to Machine Learning by Prof. Balaraman Ravindran (IIT Madras)
   https://nptel.ac.in/courses/106106139
3. Coursera – Machine Learning by Andrew Ng (Stanford University)
   https://www.coursera.org/learn/machine-learning

Course Code	Course Name	Teaching Scheme (Hrs/week)			Credits Assigned			
2276116	**Artificial Intelligence Lab**	L	T	P	L	T	P	Total
		--	--	2	--	--	1	1
		Examination Scheme						
		Term work		**Orals**			**Total**	
		25		25			50	

Pre-requisite Course Codes	**ESL 102 BEEE Lab, BSL2012 Semiconductor Physics Lab**	
Laboratory Objectives	1.	Develop practical understanding of intelligent agents, search techniques, knowledge representation, and reasoning methods used in Artificial Intelligence.
	2	Apply machine learning and probabilistic reasoning techniques for solving real-world AI problems.
	3.	Implement and evaluate AI algorithms using suitable programming tools and frameworks.
	4.	Explore modern AI applications including conversational AI, generative AI, and Large Language Models (LLMs).
Laboratory Outcomes	After the successful completion students should be able to	
	LO1	Analyze AI problems using PEAS description, task environments, and suitable intelligent agent architectures.
	LO2	Implement basic AI problem-solving techniques using uninformed and informed search algorithms.
	LO3	Represent knowledge using PROLOG and First Order Predicate Logic (FOPL) and perform logical inference.
	LO4	Apply probabilistic reasoning and Bayesian Networks for reasoning under uncertainty.
	LO 5	Use modern AI tools, LLMs and pretrained transformer models for generative AI applications.
	LO6	Analyze real-world AI systems with respect to functionality, performance and practical applications.

Laboratory Experiments:

Sr. No.	Title of experiment	Hardware /Software	Module	Reference
1	Write a simple program to provide the PEAS description and TASK Environment for a given AI problem and Identify suitable Agent Architecture for the problem	Logical	**1**	**1,2**
2	Design Vacuum Cleaner as reflex based and Model -Based Reflex Agent.	Python	**2**	**1,2**
3	Write basic PROLOG programs using facts and simple queries to understand syntax and execution.	PROLOG Software	**4**	**3**
4.	Implement any one of the Uninformed search techniques	Python/Virtual Lab	**3**	**1,2**
5.	Implement any one of the Informed search techniques	Python/Virtual Lab	**3**	**1,2**
6	Implement any one of the Local Search techniques. e.g. Hill Climbing, Simulated Annealing.	Python/Virtual Lab	**3**	**1,2**
7.	Convert the given statements in FOPL and draw inference from it using Resolution.	Software	**4**	**1,2**
8	Create a Bayesian Network for the given Problem Statement and draw inferences from it.	Software	**5**	**1,2**
9	Implement adversarial search using min-max algorithm.	Software	**3**	**1,2**
10.	Use an LLM interface/API to test different prompts	Software	**6**	**4**
11.	Generate text summaries, emails, or stories using pretrained transformer models	Software	**6**	**4**
12	Case study of any existing successful AI system	Software	**6**	

Laboratory Assessment:

Assessment:

Term Work: Term Work shall consist of at least 10 to 12 practicals' based on the above list. Also, Term work Journal must include at least 2 assignments.

Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)

Practical/ Oral Exam: An Oral examination will be held based on the above syllabus.

Recommended Books:

1. Stuart J. Russell and Peter Norvig, "Artificial Intelligence: A Modern Approach Second Edition" Pearson Education.
2. Elaine Rich and Kevin Knight ―Artificial Intelligence‖ Third Edition, Tata McGraw-Hill Education Pvt. Ltd., 2008.
3. Ivan Bratko "PROLOG Programming for Artificial Intelligence", Pearson Education,

   Third Edition.
4. A Beginner's Guide to Generative AI: An Introductory Path to Diffusion Models, ChatGPT, and LLMs

Term Work:

At least 10 experiments covering the entire syllabus of **PCL503** (Artificial intelligence Lab) should be set to have well predefined inference and conclusion. This must include **50%** Hardware and **50%** Simulation experiments. The experiments should be student centric and attempts should be made to make the experiments meaningful and interesting. Experiments must be graded from time to time. The grades should be converted into marks as per the Credit and Grading System manual and should be added and averaged. The grading and term work assessment should be done based on this scheme. The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work. Practical and Oral exam will be based on the entire syllabus.

Note:

Suggested List of Experiments is indicative. However, flexibility lies with individual course instructors to design and introduce new, innovative and challenging experiments, (limited to maximum 30% variation to the suggested list) from within the curriculum, so that the fundamentals and applications can be explored to give greater clarity to the students and they can be motivated to think differently.

<table>
<tr><th rowspan="2">Course Code</th><th rowspan="2">Course Name</th><th colspan="3">Teaching Scheme (Contact Hours)</th><th colspan="4">Credits Assigned</th></tr>
<tr><th>Theory</th><th>Practical</th><th>Tutorial</th><th>Theory</th><th>Practical</th><th>Tutorial</th><th>Total</th></tr>
<tr><td rowspan="4">22751141</td><td rowspan="4">Semiconductor Device Physics</td><td>03</td><td>--</td><td>--</td><td>03</td><td>--</td><td>--</td><td>03</td></tr>
<tr><td colspan="7">Examination Scheme</td></tr>
<tr><td>IA1</td><td>IA2</td><td>Total</td><td>ESE</td><td>Exam duration (Hrs.)</td><td colspan="2">Total</td></tr>
<tr><td>20</td><td>20</td><td>40</td><td>60</td><td>02</td><td colspan="2">100</td></tr>
</table>

Course Prerequisite:

- Engineering Physics, Electronic Devices and Circuit

Course Objectives:

1. Explain semiconductor crystal structures and carrier dynamics with relevance to material selection in devices.
2. Apply carrier statistics to understand doping and conductivity control in IC fabrication.
3. Relate carrier transport mechanisms to modern electronic and optoelectronic devices.
4. Interpret recombination and generation processes in solar cells and photodiodes.
5. Connect junction physics to capacitance effects in RF and high-frequency circuits.
6. Discuss advanced materials and their role in high-power and high-speed applications.

Course Outcomes:

After successful completion of the course, student will be able to:

1. Explain semiconductor crystal structures and carrier dynamics with relevance to material selection in devices.
2. Apply carrier statistics to understand doping and conductivity control in IC fabrication.
3. Relate carrier transport mechanisms to modern electronic and optoelectronic devices.
4. Interpret recombination and generation processes in solar cells and photodiodes.
5. Connect junction physics to capacitance effects in RF and high frequency circuits.
6. Discuss advanced materials and their role in high power and high speed applications.

Module No.	Unit No.	Topics	Hrs.
1.0		**Fundamentals of Semiconductor Materials**	**05**
	1.1	Crystal structure, bonding, energy bands, direct vs indirect bandgap: applications in LEDs (direct bandgap) and solar cells (indirect bandgap)	
2.0		**Carrier Statistics and Semiconductor Properties**	**07**
	2.1	**Carrier statistics:** Fermi level, intrinsic & extrinsic semiconductors, effective mass, applications in doping control for ICs, sensors, and MEMS devices.	
3.0		**Carrier Transport Mechanisms**	**07**
	3.1	**Carrier transport:** drift, diffusion, mobility, conductivity, applications in high speed transistors, diffusion in PN junctions, semiconductor sensors	
	3.2	**Classical Transport** Drift current: mobility and conductivity, Diffusion current and gradient-based transport, Einstein relation D/μ, Continuity equation (basic idea)	

	3.3	**Transport in Nanoscale Devices**: Velocity saturation, Ballistic transport, Scaling effects	
4.0		**Recombination and Generation Mechanisms**	**06**
	4.1	Recombination & generation: carrier lifetime, quasi-Fermi levels: applications in solar cell efficiency, photodiode response, LED performance	
5.0		**PN Junction Physics**	**06**
	5.1	Junction physics: depletion region, built in potential, capacitance effects, applications in RF varactors, high frequency circuits, voltage controlled oscillators	
6.0		**Advanced Semiconductor Devices and Materials**	**08**
	6.1	Advanced topics: high field effects, avalanche breakdown, hot carriers, emerging materials (GaN, SiC, nano devices), applications in power electronics, high frequency communication, nano electronics	
	6.2	MultiGate Structures, Nanowire Transistors, Quantum dots, Spintronics.	
Total			**39**

Internal Assessment (40-Marks):

Internal Assessment (IA) consists of two class tests of 20 marks each. IA-1 is to be conducted on approximately 40% of the syllabus and IA-2 will be based on remaining contents (approximately 40% syllabus but excluding contents covered in IA-I). Duration of each test shall be one hour.

End Semester Examination (60-Marks):

Weightage to each of the modules in end-semester examination will be proportional to number of respective lecture hours mentioned in the curriculum.

8. Question paper will comprise of **total 06** questions, each carrying **15 marks**.
9. **Question No: 01** will be **compulsory** and based on entire syllabus wherein 4 to 5 sub-questions will be asked.
10. Remaining questions will be mixed in nature and randomly selected from all the modules.
11. **Total 04 questions** need to be attempted.

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	**Practical**	**Tutorial**	**Theory**	**Practical**	**Tutorial**	**Total**
22751181	Semiconductor Device Physics Lab	--	02	--	--	01	--	01
		Examination Scheme						
		Term Work		PR / OR Exam		Total		
		25		25		50		

Course Objectives:

1. Develop practical skills in measuring and analyzing semiconductor properties such as resistivity, mobility, and conductivity.
2. Apply theoretical concepts (carrier dynamics, junction physics, recombination) through hands-on experiments and simulations.
3. Investigate device behavior in real-world applications like solar cells, photodiodes, and PN junctions.
4. Use simulations to explore advanced materials (GaN, SiC) and nanoscale devices (nanowire transistors, quantum dots).
5. Strengthen experimental methodology by performing systematic measurements, interpreting data, and validating semiconductor models.
6. Bridge theory and practice to enhance understanding of modern electronic and optoelectronic devices.

Course Outcomes:

After successful completion of the course, student will be able to:

1. Measure and analyze semiconductor properties such as resistivity, carrier type, and mobility using experimental methods (e.g., four-probe and Hall effect).
2. Evaluate device performance by studying I–V characteristics of solar cells, photodiodes, and PN junctions under different conditions.
3. Apply simulations to understand band structures, carrier distributions, and advanced materials like GaN, SiC, nanowires, and quantum dots.
4. Bridge theoretical knowledge with practical applications in modern semiconductor devices, optoelectronics, and nanoscale electronics.

Suggested List of Experiments:

Sr. No.	Title of Experiment	Hrs.
1.	Resistivity measurement of semiconductor samples (four probe method)	2
2.	Hall effect experiment: determination of carrier type and mobility	2
3.	Study of photoconductivity in semiconductors	2
4.	Temperature dependence of conductivity (activation energy)	2
5.	Capacitance–voltage characteristics of a PN junction	2
6.	I-V characteristics of solar cells under illumination	2
7.	Simulation of band structures and carrier distribution	2
8.	Observation of breakdown phenomena in semiconductor junctions	2
9.	Study of recombination lifetime using photodiode response	2

10.	Simulation of advanced materials (GaN, SiC)	2
11.	Resistivity measurement of semiconductor samples (four probe method)	2
12.	Hall effect experiment: determination of carrier type and mobility	2
13.	Nanowire Transistor Simulation: Analyze transport in nanoscale devices	2
14.	Quantum Dot Behavior (Simulation): Study discrete energy levels	2

Assessment:

Term Work: Term Work shall consist of at least 8 to 10 experiments. Teachers should refer to the suggested experiments and can design additional experiments to maintain better understanding and quality. Term work Journal must include at least 2 assignments.

Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)

Practical & Oral Exam: An Oral & Practical exam will be held based on the above syllabus.

Note:

Suggested List of Experiments is indicative. However, flexibility lies with individual course instructors to design and introduce new, innovative and challenging experiments, (limited to maximum 30% variation to the suggested list) from within the curriculum, so that the fundamentals and applications can be explored to give greater clarity to the students and they can be motivated to think differently.

Course Code	Course Name	Teaching Scheme (Hrs/week)			Credits Assigned			
22751142	**Machine Learning**	L	T	P	L	T	P	Total
		3	--	2	3	--	1	4
		Examination Scheme						
			IA1	**IA2**	**ESE**		**Total**	
		Theory	20	20	60		100	

Pre-requisite Course Codes	PC304 Data Structure and Algorithms, VSEC202 Python Programming, PC 401 Math	
Course Outcomes	After the successful completion students should be able to	
	CO1	Comprehend basics of machine learning
	CO2	Build mathematical foundation for machine learning
	CO3	Understand various machine learning models
	CO4	Select suitable machine learning models for a given problem
	CO5	Build neural network based models
	CO6	Apply dimensionality reduction techniques

Module No.	Unit No.	Topics	Reference	Hrs.
Module 1	**1**	**Introduction to Machine Learning**		6 Hrs
	1.1	Introduction to Machine Learning, Issues in Machine Learning, Application of Machine Learning, Steps of developing a Machine Learning Application	1,4,6,8	
	1.2	Supervised and Unsupervised Learning: Concepts of Classification, Clustering and Prediction, Training, Testing and validation dataset, cross validation, overfitting and under fitting of model		
	1.3	Performance Measures: Measuring Quality of model-Confusion Matrix, Accuracy, Recall, Precision, Specificity, F1 Score, RMSE, Bias -Variance trade-off		
Module 2	**2**	**Mathematical Foundation for ML**		5 Hrs
	2.1	System of Linear equations, Norms, Inner products, Length of Vector, Distance between vectors, Orthogonal vectors	2,7	
	2.2	Symmetric Positive Definite Matrices, Determinant, Trace, Eigenvalues and vectors, Orthogonal Projections, Diagonalization, SVD and its applications		

Module 3	**3**	**Linear models**	4,7,8	7 Hrs
	3.1	The least-squares method, Multivariate Linear Regression, Regularised Regression, Using Least-Squares Regression for classification		
	3.2	Support Vector Machine Constrained Optimization, Optimal decision boundary, Quadratic Programming, SVM for linear and nonlinear classification, Basics of Kernel trick.		
Module 4	**4**	**Clustering Algorithms**		6 Hrs
	4.1	Cluster Analysis: Basic Concepts; Partitioning Methods: K-Means, K Medoids; Hierarchical Methods: Agglomerative, Divisive, BIRCH; Density-Based Methods: DBSCAN, Hebbian Learning rule	3,11,12,9	
	4.2	What are outliers? Types, Challenges; Outlier Detection Methods		
Module 5	**5**	**Classification models**		11 Hrs
	5.1	Evolution of Neural Networks, Biological Neuron, Artificial Neural Networks, NN architecture, McCulloch-Pitts Model, Designing a simple network, Non-separable patterns, Perceptron model with Bias. Activation functions, Binary, Bipolar, continuous, Ramp. Limitations of Perceptron, Perceptron Learning Rule. Delta Learning Rule (LMS-Widrow Hoff), Multilayer perceptron network. Adjusting weights of hidden layers. Error back propagation algorithm Logistic regression	4,7,10,11, 12,9	
	5.2	Basic Concepts; Classification methods: Decision Tree Induction: Attribute Selection: Gini Index, Tree pruning, Multiclass Classification, K-Nearest Neighbors, Naive Bayes Classifier, Random Forest		
	5.3	Introduction of Ensemble methods, Bagging, Boosting, AdaBoost and Random forest.		
Module 6	**6**	**Dimensionality Reduction Techniques**		4 Hrs
	6.1	Curse of Dimensionality.	2,7,4	
	6.2	Feature Selection and Feature Extraction		
	6.3	Dimensionality Reduction Techniques, Principal Component Analysis, Linear Discriminant Analysis		
			Total	**39**

Course Assessment:

Theory:

IA1: 20 Marks written examination for one hour
IA2: 20 Marks written examination for one hour

ESE: 60 Marks written examination for two hours

Recommended Books:

1. Nathalie Japkowicz & Mohak Shah, “Evaluating Learning Algorithms: A Classification Perspective”, Cambridge.
4. Marc Peter Deisenroth, Aldo Faisal, Cheng Soon Ong, “Mathematics for machine learning”
5. Samir Roy and Chakraborty, “Introduction to soft computing”, Pearson Edition.
6. Ethem Alpaydın, “Introduction to Machine Learning”, MIT Press
7. Peter Flach, “Machine Learning”, Cambridge University Press
8. Tom M. Mitchell, “Machine Learning”, McGraw Hill
9. Kevin P. Murphy, “Machine Learning — A Probabilistic Perspective”, MIT Press
10. Stephen Marsland, “Machine Learning an Algorithmic Perspective”, CRC Press
11. Shai Shalev-Shwartz, Shai Ben-David, “Understanding Machine Learning”, Cambridge University Press
12. Peter Harrington, “Machine Learning in Action”, DreamTech Press
13. Han Kamber, Data Mining Concepts and Techniques‖, Morgan Kaufmann Publishers

Online References:

5. NPTEL: https://nptel.ac.in/courses/106106139
6. Supervised Machine Learning: Regression and Classification | Coursera
7. https://www.learndatasci.com/out/google-machine-learning-crash-course/

Course Code	Course Name	Teaching Scheme (Hrs/week)			Credits Assigned			
		L	T	P	L	T	P	Total
22751182	Machine learning Lab	--	--	2	--	--	1	1
		Examination Scheme						
		Term work		Oral and Practical			Total	
		25		25			50	

Pre-requisite Course Codes	PC304 Data Structure and Algorithms, VSEC202 Python Programming, PC 401 Math	
Laboratory Objectives	1.	To understand the implementation of basic Machine Learning algorithms using Python.
	2	To apply classification, regression, clustering, and dimensionality reduction techniques on datasets.
	3.	To analyze the performance of Machine Learning models using suitable evaluation metrics.

	4.	To develop problem-solving skills for real-world Machine Learning applications.
Laboratory Outcomes	After the successful completion students should be able to	
	LO 1	Implement supervised and unsupervised Machine Learning algorithms
	LO 2	Analyze Machine Learning models using suitable performance metrics and validation techniques.
	LO 3	Apply mathematical foundations and linear models for Machine Learning problems.
	LO 4	Implement classification, clustering, and neural network techniques for data analysis.
	LO 5	Apply dimensionality reduction and feature extraction techniques on datasets.
	LO 6	Develop Machine Learning solutions for real-world engineering applications using appropriate tools and libraries.

Laboratory Experiments:

Sr. No.	Title of Experiment	Module	Reference
1	Study and implementation of basic Python libraries for Machine Learning: NumPy, Pandas, Matplotlib and Scikit-learn	1	8,10
2	Implementation of data preprocessing techniques including handling missing values, normalization and dataset splitting	1	4,6,8
3	Perform supervised learning using a simple classification dataset and evaluate model performance	1	1,4,6
4	Perform unsupervised learning using clustering techniques on a sample dataset	1	4,8,11
5	Implementation of performance evaluation metrics using Confusion Matrix, Accuracy, Precision, Recall, Specificity and F1 Score	1	1,5,9
6	Study of overfitting and underfitting using training and testing datasets with cross validation	1	4,5,9
7	Implementation of vector and matrix operations using NumPy	2	2,10
8	Program to compute norms, inner products and distances between vectors	2	2
9	Implementation of Eigenvalues, Eigenvectors and matrix diagonalization	2	2

10	Demonstration of Singular Value Decomposition (SVD) and its applications in data compression	2	2,8
11	Implementation of Simple and Multivariate Linear Regression using least squares method	3	4,6,8
12	Implementation and comparison of Linear Regression with Regularized Regression techniques (Ridge and Lasso)	3	7,9
13	Implementation of Logistic Regression for classification problems	3	4,7,8
14	Implementation of Support Vector Machine for linear and nonlinear classification using kernel functions	3	7,9
15	Implementation of K-Means clustering algorithm on real-world datasets	4	8,11
16	Implementation of K-Medoids clustering techniques	4	11
17	Implementation of Hierarchical clustering using Agglomerative methods	4	11
18	Implementation of DBSCAN clustering and analysis of density-based clusters	4	11
19	Detection and visualization of outliers using statistical and distance-based methods	4	11
20	Implementation of McCulloch-Pitts neuron and Perceptron learning algorithm	5	3,8
21	Study and implementation of various activation functions in Neural Networks	5	3,7
22	Implementation of Multilayer Perceptron using Backpropagation algorithm	5	3,8
23	Implementation of Decision Tree classifier and analysis using Gini Index	5	5,11
24	Implementation of K-Nearest Neighbors (KNN) classifier	5	4,11
25	Implementation of Naive Bayes classifier for categorical and numerical datasets	5	4,7
26	Implementation and comparison of ensemble learning methods: Bagging, AdaBoost and Random Forest	5	5,7,9
27	Implementation of feature selection techniques for dimensionality reduction	6	4,7
28	Implementation of Principal Component Analysis (PCA) for feature extraction and visualization	6	2,7
29	Implementation of Linear Discriminant Analysis (LDA) for classification and dimensionality reduction	6	2,7
30	Develop and evaluate a complete Machine Learning model pipeline for a real-world dataset	6	4,7,10

Laboratory Assessment:

Assessment:

- Term Work: Term Work shall consist of at least 10 to 12 practical's based on the above list. Also, Term work Journal must include at least 2 assignments.
- Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)
- Practical/ Oral Exam: An practical and Oral examination will be held based on the entire syllabus.

Term Work:

At least 10-12 experiments covering the entire syllabus should be set to have well predefined inference and conclusion. The experiments should be student centric and attempts should be made to make the experiments meaningful and interesting. Experiments must be graded from time to time. The grades should be converted into marks as per the Credit and Grading System manual and should be added and averaged. The grading and term work assessment should be done based on this scheme. The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work. Practical and Oral exam will be based on the entire syllabus.

Note:

Suggested List of Experiments is indicative. However, flexibility lies with individual course instructors to design and introduce new, innovative and challenging experiments, (limited to maximum 30% variation to the suggested list) from within the curriculum, so that the fundamentals and applications can be explored to give greater clarity to the students and they can be motivated to think differently.

Course Code	Course Name	Teaching Scheme (Hrs/week)			Credits Assigned			
22751143	**Data Base Management Systems**	L	T	P	L	T	P	Total
		3	--	2	3	--	1	4
		Examination Scheme						
			IA1	**IA2**	**ESE**		**Total**	
		Theory	20	20	60		100	

Pre-requisite Course Codes		
Course Outcomes	After the successful completion students should be able to	
	CO1	Recognize the need of database management system
	CO2	Design ER and EER diagram for real life applications
	CO3	Construct relational models and write relational algebra queries.
	CO4	Formulate SQL queries.
	CO5	Apply the concept of normalization to relational database design.
	CO6	Describe the concept of transaction, concurrency and recovery

Module No.	Unit No.	Topics	Reference	Hrs.
Module 1	**1**	**Introduction Database Concepts**		4Hrs
	1	Introduction, Characteristics of databases, File system v/s Database system, View of Data: Data abstraction and data Independence, Data Models, Schemas and Instances, Three-Schema Architecture, Overall DBMS system architecture, Database Users and Administrator, Database Applications	1,2	
Module 2	**2**	**Entity–Relationship Data Model**		6Hrs
	2	The Entity-Relationship (ER) Model: Entity types: Weak and strong entity sets, Entity sets, Types of Attributes, Relationship constraints: Cardinality and Participation, Entity-Relationship Diagrams, Mapping ER model to a relational schema	1,2	

Module 3	**3**	**Relational Model and relational Algebra**	1,2	7 Hrs
	3.1	Introduction to the Relational Model, Codd's rules Relational schema and Concept of keys. Relational Algebra-operators, Relational Algebra Queries		
	3.2	Integrity constraints: key constraints, Domain Constraints, Referential integrity , check constraints		
Module 4	**4**	**Structured Query Language (SQL)**		8 Hrs
	4.1	Overview of SQL, Data Definition Commands, Data Manipulation commands, Data Control commands	1	
	4.2	Set and string operations, aggregate functions, group by, having, Views in SQL, Joins, Nested and Subqueries, Triggers.		
Module 5	**5**	**Relational-Database Design**		6Hrs
	5.	Concept of normalization, First Normal Form, Second Normal Form, Function Dependencies: Closure of Attribute sets, Algorithms for Decomposition: 3NF, Boyce-Codd Normal Form (BCNF).	1,3	
Module 6	**6**	**Transactions Management, Concurrency, and Recovery**		8Hrs
	6.1	Transaction concept, Transaction states, ACID properties, Transaction Control Commands, Concurrent Executions	1,2	
	6.2	Serializability:Conflict and View Concurrency Control: Lock-based, Timestamp-based protocols, Deadlock handling Recovery System: Log based recovery Database privacy and Security		
			Total	**39**

Course Assessment:

Theory:

IA1: 20 Marks written examination for one hour
IA2: 20 Marks written examination for one hour

ESE: 60 Marks written examination for two hours

Recommended Books:

[1] A. Silberschatz, H. F. Korth & S. Sudarshan, “Database System Concepts”, McGraw Hill, 7thEdition 2019.
[2] R. Elmasri & S. B. Navathe, “Fundamentals of Database Systems”, Addison Wesley, 7th Edition, 2016
[3] Raghu Ramakrishnan, “DatabaseManagement Systems”, Mcgraw-Hill, 4thEdition, 2018
[4] C.J.Date, A.Kannan, S.Swamynathan,” An Introduction to Database Systems”, Pearson, Eighth Edition, 2006.

Online References:

[1] NPTEL courses on DBMS: Introduction to Database systems

Course Code	Course Name	Teaching Scheme (Hrs/week)			Credits Assigned			
22751183	Data Base Management Systems Lab	L	T	P	L	T	P	Total
		--	--	2	--	--	1	1
		Examination Scheme						
		Term work		Orals			Total	
		25		25			50	

Pre-requisite Course Codes		
Laboratory Objectives	1.	To understand and design ER/EER models for real-world database applications.
	2.	To develop skills in mapping conceptual models to relational schemas.
	3.	To gain hands-on experience in SQL (DDL, DML, DCL, TCL) and query formulation.
	4.	To apply concepts of normalization, transactions, and concurrency control in database systems.
Laboratory Outcomes	After the successful completion students should be able to	
	LO1	Identify real-world problems and design appropriate ER/EER

		diagrams.
	LO2	Convert ER models into relational schema with proper keys and constraints.
	LO3	Create and manage databases using SQL commands.
	LO4	Write complex SQL queries using joins, nested queries, and aggregate functions.
	LO 5	Apply normalization techniques to design efficient and consistent databases.
	LO 6	Implement transaction management and concurrency control mechanisms in database systems.

Laboratory Experiments:

Sr. No.	Title of experiment	Hardware /Software	Module	Reference
1.	**Identify the case study and detailed statement of the problem. Design an Entity-Relationship(ER) / Extended Entity-Relationship (EER) Model.**	Software	1	
2.	**Mapping ER/EER to Relational schema model.**	Software	2	
3.	**Create a database using Data Definition Language (DDL) and apply integrity constraints for the specified System.**	Software	2	
4.	**Apply DML Commands for the specified system.**	Software	3	
5.	**To perform Simple queries, string manipulation operations and aggregate functions.**	Software	3	
6	**To perform Nested and Complex queries**	Software	3	
7	**To implement Views and Join operations.**	Software	4	
8.	**To perform DCL and TCL commands.**	Software	4	
9.	**To implement function and trigger.**	Software	4	
10.	**To demonstrate Database connectivity**	Software	5	

Laboratory Assessment:

Assessment:

Term Work: Term Work shall consist of at least 10 to 11 practicals' based on the above list. Also, Term work Journal must include at least 2 assignments.

Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)

Practical/ Oral Exam: An Oral examination will be held based on the above syllabus.

Recommended Books:

[1] Patrick Dalton,"Microsoft SQL Server Black Book".

[2] Patel J, "DBMS Lab Manual", eBookIt.com, 2012.

Term Work:

At least 10 experiments covering the entire syllabus of PEL 501 (**Data Base Management Systems** Lab) should be set to have well predefined inference and conclusion. The experiments should be student centric and attempts should be made to make the experiments meaningful and interesting. Experiments must be graded from time to time. The grades should be converted into marks as per the Credit and Grading System manual and should be added and averaged. The grading and term work assessment should be done based on this scheme. The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work. Practical and Oral exam will be based on the entire syllabus.

Note:

Suggested List of Experiments is indicative. However, flexibility lies with individual course instructors to design and introduce new, innovative and challenging experiments, (limited to maximum 30% variation to the suggested list) from within the curriculum, so that the fundamentals and applications can be explored to give greater clarity to the students and they can be motivated to think differently.

Course Code	Course Name	Teaching Scheme (Hrs/week)			Credits Assigned			
22751144	Data Compression	L	T	P	L	T	P	Total
		3	--	1	3	--	1	4
		Examination Scheme						
		Theory	**IA1**	**IA2**	**ESE**		**Total**	
			20	20	60		100	
		Practical	**Term Work**		**Practical Exam**		**Total**	
			25		25		50	

Pre-requisite Course Codes	**PC301- Signals and Systems, PC304- Analysis of Algorithms, PC401 – Maths, PC403- Discrete Structures and Automata Theory,**	
Course Outcomes	After the successful completion students should be able to	
	CO1	Explain fundamentals of information theory, redundancy, and compression models.
	CO2	Apply statistical and dictionary-based lossless compression algorithms.
	CO3	Analyze transform and quantization techniques used in lossy compression.
	CO4	Explain image, audio, and video compression standards used in industry.
	CO5	Evaluate secure and efficient compression techniques for cloud, multimedia, and IoT applications.
	CO6	Implement and compare compression algorithms using modern software tools.

Module No.	Unit No.	Topics	Reference	Hrs.
Module 1	**1**	**Fundamentals of Data Compression and Information Theory**		7 Hrs
	1.1	Introduction to Data Compression, Need for Compression, Applications in Multimedia, Cloud, AI, IoT and Cyber Security	1, 2, 3	
	1.2	Lossless and Lossy Compression, Performance Measures: Compression Ratio, Redundancy, Entropy, Bit Rate and Distortion		
	1.3	Information Theory Fundamentals, Entropy,Mutual Information, Source Coding Theorem, Binary Symmetric Channels		
	1.4	Mathematical Models: Physical Models, Probability Models, Markov Models		
	1.5	Coding Concepts: Prefix Codes, Uniquely Decodable Codes, Kraft Inequality		
Module 2	**2**	**Statistical and Entropy-Based Compression Techniques**		8 Hrs
	2.1	Run Length Encoding (RLE), Shannon-Fano Coding	1, 2, 3, 4	
	2.2	Huffman Coding: Binary Huffman Coding, Adaptive Huffman Coding, Minimum Variance Huffman Coding, Extended Huffman code		
	2.3	Golomb Coding and Rice Coding		
	2.4	Arithmetic Coding: Encoding and Decoding Process, Binary Arithmetic Coding		
	2.5	Comparison of Huffman and Arithmetic Coding, Applications in Text and Image Compression		
Module 3	**3**	**Dictionary-Based and Predictive Compression Techniques**	1, 2, 3, 4	7 Hrs
	3.1	Dictionary-Based Compression: Static and Adaptive Dictionaries		
	3.2	LZ Family Algorithms: LZ77, LZ78, LZW		
	3.3	Burrows-Wheeler Transform (BWT) and Move-to-Front Coding		
	3.4	Prediction by Partial Matching (PPM), Context-Based Compression		
	3.5	Dynamic Markov Compression, Applications in GIF, PNG and Web Compression		
Module 4	**4**	**Image Compression and Modern Image Standards**		8 Hrs
	4.1	Fundamentals of Digital Images and Image Redundancies	1, 2, 3, 4	
	4.2	Transform Coding: Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT)		
	4.3	Scalar Quantization and Vector Quantization		
	4.4	JPEG Compression: Baseline JPEG and JPEG-LS		
	4.5	JPEG2000, WebP and HEIF Image Formats		
	4.6	Applications in Medical Imaging, Satellite Imaging and Mobile Platforms		
Module 5	**5**	**Audio and Video Compression Standards**		7 Hrs
	5.1	Basics of Digital Audio and Psychoacoustic Models	1, 2, 3, 4, 5	
	5.2	Audio Compression Techniques: MP3, AAC and FLAC Compression		
	5.3	Fundamentals of Digital Video, Spatial and Temporal Redundancy		
	5.4	Motion Estimation and Motion Compensation		
	5.5	Video Compression Standards: MPEG-1, MPEG-2, MPEG-4, H.264/AVC, H.265/HEVC and AV1		
	5.6	Applications in Video Streaming, OTT Platforms and Video Conferencing		

Module 6	**6**	**Secure Compression, Emerging Trends and Applications**		8 Hrs
	6.1	Compression and Encryption Relationship, Secure Compressed Data Transmission	2, 3, 4, 5, 6	
	6.2	Compression in Cloud Storage, IoT Devices and Edge Computing		
	6.3	Error Resilience and Compression in Cyber Security Applications		
	6.4	AI-Assisted Compression and Neural Data Compression Concepts		
	6.5	Green Computing and Energy-Efficient Compression		
	6.6	Case Studies: Google Brotli, Zstandard (Zstd), Secure Multimedia Streaming		
			Total	**45**

Course Assessment:

Theory:

IA1: 20 Marks written examination for one hour
IA2: 20 Marks written examination for one hour

ESE: 60 Marks written examination for two hours

Recommended Books:

[1] Khalid Sayood, *Introduction to Data Compression*, Morgan Kaufmann.
[2] David Salomon and Giovanni Motta, *Handbook of Data Compression*, Springe.
[3] Mark Nelson and Jean-Loup Gailly, *The Data Compression Book*, BPB Publication.
[4] Ian H. Witten, Alistair Moffat, Timothy Bell, *Managing Gigabytes*, Morgan Kaufmann.
[5] Rafael C. Gonzalez and Richard E. Woods, *Digital Image Processing*, Pearson.
[6] Yao Wang, Joemelakha and Alan Bovik, *Video Processing and Communications*, Pearson.

Online References:

1. NPTEL, "Data Compression", Available at: NPTEL Data Compression Course
2. MIT OpenCourseWare, "Information Theory and Data Compression", Available at: MIT OpenCourseWare - Information Theory and Data Compression
3. Stanford University, "Digital Signal Processing and Compression Concepts", Available at: Stanford Online Courses
4. Google Developers, "WebP Compression Technology", Available at: Google WebP Documentation
5. TensorFlow Compression Library, "Machine Learning Based Compression", Available at: TensorFlow Compression Library
6. Mozilla Developer Network (MDN), "Compression Techniques for Web Applications", Available at: MDN Compression Guide
7. IETF, "Brotli Compressed Data Format", Available at: IETF Brotli Specification

	Course Code	Course Name	Teaching Scheme (Hrs/week)			Credits Assigned			
	22751184	Data Compression Lab	L	T	P	L	T	P	Total
			--	--	2	--	--	1	1
			Examination Scheme						
			Term work		Orals			Total	
			25		25			50	

Pre-requisite Course Codes	**ESL 102 BEEE Lab, BSL2012 Semiconductor Physics Lab**	
Laboratory Objectives	1.	To provide hands-on experience in implementing various data compression algorithms for text, image, audio, and video data
	2	To understand and analyze the performance of lossless and lossy compression techniques.
	3.	To introduce modern software tools and programming libraries used for multimedia compression applications.
	4.	To develop practical skills for implementing secure and efficient compression techniques used in cloud and multimedia systems.
Laboratory Outcomes	After the successful completion students should be able to	
	LO1	Implement basic lossless compression algorithms such as RLE, Huffman Coding, and Arithmetic Coding.
	LO2	Analyze and compare dictionary-based compression techniques including LZ77 and LZW algorithms.
	LO3	Implement image compression techniques using JPEG and transform coding methods.
	LO4	Evaluate audio and video compression standards using modern multimedia tools

	LO 5	Analyze compression efficiency using parameters such as compression ratio, bitrate, and redundancy.
	LO6	Develop applications related to secure and multimedia data compression using software tools and programming libraries.

Laboratory Experiments:

Sr. No.	Title of experiment	Hardware / Software	Module	Reference
1.	**Study of Compression Ratio and Performance Metrics for Different File Types**	**Software**	**1**	**1, 2**
2.	**Implementation of Run Length Encoding (RLE) for Text Compression**	**Software**	**2**	**1, 2**
3.	**Implementation of Shannon Fano and Huffman Coding for Text Compression**	**Software**	**2**	**1, 2, 3**
4.	**Comparative Analysis of Huffman and Arithmetic Coding Techniques**	**Software**	**2**	**1, 2, 3**
5.	**Implementation of Arithmetic Coding for Data Compression**	**Software**	**2**	**1, 2**
6	**Implementation of LZ77 Compression Algorithm**	**Software**	**3**	**1, 2**
7	**Implementation of LZ78 and LZW Compression Algorithm**	**Software**	**3**	**1, 2, 3**
8.	**Study and Implementation of Burrows-Wheeler Transform (BWT)**	**Software**	**3**	**1, 2**
9.	**JPEG Image Compression and Decompression using DCT**	**Software**	**4**	**1, 2, 5**
10.	**Comparative Analysis of JPEG and JPEG2000 Compression Standards**	**Software**	**4**	**1, 2, 5**
11.	**Implementation of Scalar and Vector Quantization Techniques**	**Software**	**4**	**1, 2**
12.	**Audio Compression using MP3 and AAC Formats**	**Software**	**5**	**1, 2, 6**
13	**Video Compression using MPEG and H.264 Standards**	**Software**	**5**	**1, 2, 6**
14	**Secure File Compression using Compression and Encryption Techniques**	**Software**	**6**	**2, 3**
15	**Mini Project: AI-Based Multimedia Compression and Performance Analysis**	**Software**	**6**	**3, 5, 6**

Laboratory Assessment:

Assessment:

Term Work: Term Work shall consist of at least 10 to 12 practicals based on the above list. Also, Term work Journal must include at least 2 assignments or one mini project/case study related to cybersecurity or cryptography applications.

Term Work Marks:

25 Marks (Total marks) =
15 Marks (Experiment)
5 Marks (Assignments/Mini Project)
5 Marks (Attendance)

Practical/ Oral Exam:

An Oral examination will be conducted based on the entire syllabus and laboratory experiments.

Recommended Software / Tools

1. MATLAB
2. Python (NumPy, OpenCV, Pillow)
3. FFmpeg
4. Wireshark
5. GZIP / ZIP Utilities
6. TensorFlow/PyTorch (for AI-based compression concepts)

Recommended Books:

[1] Khalid Sayood, *Introduction to Data Compression*, Morgan Kaufmann.
[2] David Salomon and Giovanni Motta, *Handbook of Data Compression*, Springe.
[3] Mark Nelson and Jean-Loup Gailly, *The Data Compression Book*, BPB Publication.
[4] Ian H. Witten, Alistair Moffat, Timothy Bell, *Managing Gigabytes*, Morgan Kaufmann.
[5] Rafael C. Gonzalez and Richard E. Woods, *Digital Image Processing*, Pearson.
[6] Yao Wang, Joemelakha and Alan Bovik, *Video Processing and Communications*, Pearson.

Term Work:

At least 10 experiments covering the entire syllabus of PEL 501 (Data Compression Lab) should be set to have well predefined inference and conclusion. This must include **50%** Hardware and **50%** Simulation experiments. The experiments should be student centric and attempts should be made to make the experiments meaningful and interesting. Experiments must be graded from time to time. The grades should be converted into marks as per the Credit and Grading System manual and should be added and averaged. The grading and term work assessment should be done based on this scheme. The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work. Practical and Oral exam will be based on the entire syllabus.

Note:

Suggested List of Experiments is indicative. However, flexibility lies with individual course instructors to design and introduce new, innovative and challenging experiments, (limited to maximum 30% variation to the suggested list) from within the curriculum, so that the fundamentals and applications can be explored to give greater clarity to the students and they can be motivated to think differently.

Course Code	Course Name	Teaching Scheme (Hrs/week)			Credits Assigned			
		L	T	P	L	T	P	Total
22761131	**ASIC Design & Verification**	3	--	--	3	--	--	3
		Examination Scheme						
			IA1	**IA2**	**ESE**		**Total**	
		Theory	20	20	60		100	

Pre-requisite Course Codes	**PCC2014 Digital Electronics** **PC 302 Electronic Devices** **PC 402 Analog Electronics** **PC 503 Semiconductor Devices and Characteristics**	
Course Objectives	1. To introduce the learner System Verilog concepts for verification. 2. To provide understanding of System Verilog and SVA for verification, and understand the improvements in verification efficiency. 3. To introduce the learner advanced verification features such as practical use of classes, randomization, checking and coverage. 4. To highlight the significance of verification in VLSI industry.	
Course Outcomes	After the successful completion students should be able to	
	CO1	Understand basics of Verilog Hardware Description Language and its programming
	CO2	Demonstrate an understanding of programmable devices and verification methodologies.
	CO3	Exploit new constructs in System Verilog.
	CO4	Understand the features of an object-oriented language: classes, object, varisbles
	CO5	Summarize ASIC verification techniques such as Randomization, assertions, coverage etc.
	CO6	Create layered test benches for digital designs in system Verilog.

Module No.	Unit No.	Topics	Refe renc e	Hrs.
Module 1	**1**	**Introduction to Verilog HDL**	1,2	5Hrs
	1.1	Need for Hardware Description Languages (HDL), Overview of Verilog and its applications in digital system design, Difference between Verilog and conventional programming languages, Design flow using Verilog (Design → Simulation → Synthesis)		
	1.2	Module declaration and syntax, Input, Output, and In-out ports, Data types, wire, reg. Modeling Styles in Verilog: Gate-level modeling, Dataflow modeling, Behavioral modeling. Procedural Blocks: initial block, always block, Sensitivity list		
	1.3	Introduction to Testbench: Concept of simulation, writing a simple testbench, applying input stimulus, Observing outputs		
Module 2	**2**	**Programmable Devices and Verification Basics**	1,3	7Hrs
	2.1	**Programmable Devices**: Different types of Integrated Circuits-CPLD, FPGA, ASIC, SoC (System-on-Chip), SiP (System-in-Package), MCM (Multi-Chip Module), SoP (System-on-Package), Choices based on application and cost, Architecture of FPGA, CPLD (Xilinx and Altera family devices), Difference between ASIC, FPGA and CPLD, ASIC flow and overview of types of tools used in each stage of lifecycle		

	2.2	**Verification Basics**: Introduction, Verification Process, Verification Plan, Verification Methodology options, Basic Testbench Functionality, Directed Testing, Constrained-Random Stimulus, Functional Coverage, Testbench Components, Layered Testbench, Technology challenges test, Verification languages, Verification IP reuse, Verification approaches.		
Module 3	**3**	**Data types, Procedural statements, Connecting the Test bench and Design**	1,3	7Hrs
	3.1	Data Types: Built-in Data Types, Logic Data type, Fixed-Size Arrays (Packed and Unpacked arrays), Dynamic Arrays, Queues, associative array, array methods – Reduction, Locator & ordering, Creating New Types with typedef, Creating User-Defined Structures, Enumerated Types, Constants, Strings, Expression width.		
	3.2	**Procedural statements:** Procedural Statements, Tasks, Functions, and Void Functions, routine arguments, returning from a routine, Time values.		
	3.3	**Connecting the Test bench and Design:** Separating the test-bench and design, The Interface construct, Grouping Signals in an Interface using Modports, Creating Interface Monitor, Stimulus timing with Clocking Block, Test-bench design Race Condition, Program Block, Connecting it all together, Top level Scope, Program-Module interactions.		
Module 4	**4**	**Basic Object -Oriented Programming**	2,3	6Hrs
	4.1	OOP: Class, Creating new objects, Where to Define a Class, OOP Terminology, Understanding Dynamic objects, Object Deallocation, using objects, Static vs Global Variables, Class methods, Defining methods outside class, Scoping rules, Using one class inside another, Understanding Dynamic objects, Copying objects, public vs. local, Building a test-bench		
Module 5	**5**	**Randomization and Inter-process Communication**	2,3	7Hrs
	5.1	**Randomization:** Randomization in system Verilog, Constraint details, Solution probabilities, Controlling multiple constraint blocks, Valid constraints, In-line constraints, The pre-randomize and post-randomize functions, Random number functions, Constraints tips and techniques		
	5.2	**Threads and Inter-process Communication:** Working with threads, disabling threads, inter-process communication, Events, Semaphores, Mailboxes, building a test-bench with threads and IPC.		
Module 6	**6**	**System Verilog Assertions and Functional Cover**	2, 3	7Hrs
	6.1	**System Verilog Assertions**: Types of Assertions and examples, Immediate Assertions, Concurrent Assertions, SVA Property and Sequences, Implication (Overlapped & Non-Overlapped) Operator and Repetition Operator, System Verilog Assertion built-in methods ($rose, $fell, $stable, $past)		
	6.2	**Functional Coverage**: Coverage Types, Functional Coverage Strategies, Simple Functional Coverage Example, anatomy of a cover group, triggering a cover group, data sampling, cross coverage, generic cover groups, Coverage Options, Parameterized Cover Groups, Analysing Coverage Data, Measuring Coverage Statistics During Simulation.		
			Total	**39**

Course Assessment:

Theory:

IA1: 20 Marks written examination for one hour
IA2: 20 Marks written examination for one hour
ESE: 60 Marks written examination for two hours

Recommended Books:

[1]. Chris Spear, “System Verilog for Verification: A guide to learning the testbench language features”, Springer, 3rd Edition.
[2]. J. Bhaskar “A Verilog HDL Primer” 3rd Edition Star Galaxy Publishing 2018.
[3]. Janick Bergeron, “Writing Testbenches Using System Verilog”, Springer 2006.
[4]. Stuart Sutherland, Simon Davidmann, and Peter Flake, “System Verilog for Design: A guide to using system verilog for hardware design and modeling”, Springer, 2nd Edition.
[5]. Ben Cohen, Srinivasan Venkataramanan, Ajeetha Kumari and Lisa Piper, “System Verilog Assertions Handbook”, Vhdl Cohen Publishing, 3rd edition
[6]. S Prakash Rashinkar, Peter Paterson and Leena Singh, “System on Chip Verification Methodologies and Techniques”, Kluwer Academic, 1st Edition.
[7]. System Verilog Language Reference manual
[8]. Samir Palnitkar, “Verilog HDL: A guide to Digital Design and Synthesis” second edition, Pearson – IEEE 1364-2001 compliant.
[9]. Spartan and Virtex family user manuals from Xilinx
[10]. Verilog Language Reference manual

Online References:

NPTEL courses on Introduction to ASIC design flow

Course Code	Course Name	Teaching Scheme (Hrs/week)			Credits Assigned			
		L	T	P	L	T	P	Total
22761171	**ASIC Design & Verification Lab**	--	--	2	--	--	1	1
		Examination Scheme						
		Term work		**Practical & Orals**			**Total**	
		25		25			50	

Pre-requisite Course Codes		**PCC2014 Digital Electronics Lab**
Laboratory Objectives	1.	To introduce the learner System Verilog concepts for verification.
	2	To provide understanding of System Verilog and SVA for verification, and understand the improvements in verification efficiency.
	3.	To introduce the learner advanced verification features such as practical use of classes, randomization, checking and coverage.
	4.	To highlight the significance of verification in VLSI industry.
Laboratory Outcomes	After the successful completion students should be able to	
	LO1	Create test plan and test cases to verify any digital design.
	LO2	Apply the advanced verification techniques like Randomization on set of inputs.
	LO3	Create a transaction class and apply object -oriented programming for Verification.
	LO4	Carry out simulation of designs using System Verilog hardware verification language.
	LO5	Develop a complete Layered Test-bench for any digital design

Laboratory Experiments:

Sr. No.	Title of experiment	Hardware /Software	Module	Reference
1.	Write Verilog code for 4:1 MUX using all Verilog modeling styles and simulate the same.	**Software**	**1**	**1, 2, 3, 4**
2.	Write Verilog code and test-bench for D flip flop and 4 bit counter and simulate the same.	**Software**	**1**	**1, 2, 3, 4**
3.	Create a test plan and self-checking test-bench for the ALU.	**Software**	**2**	**1, 2, 3, 4**
4.	Create dynamic arrays, associative arrays, and queues using System Verilog.	**Software**	**2**	**1, 2, 3, 4**
5.	Write test bench using dynamic arrays, associative arrays with System Verilog to test a synchronous 8-bit x64K (512kBit) RAM.	**Software**	**2**	**1, 2, 3, 4**
6	Create an Interface for a Memory Design. Use Modport to assign direction to signal.	**Software**	**3**	**1, 2, 3, 4**
7	Create class and its objects and perform deep copy and shallow copy.	**Software**	**3**	**1, 2, 3, 4**
8.	Create an Interface for a Memory Design. (without modport)	**Software**	**3**	**1, 2, 3, 4**
9.	To understand and create Virtual interface and use it in a class.	**Software**	**4**	**1, 2, 3, 4**
10.	Given design specifications, draw waveform and write SVA expressions.	**Software**	**5**	**1, 2, 3, 4**
11.	To understand Constraint randomisation	**Software**	**5**	**1, 2, 3, 4**
12.	Create IPCs like events, mailbox and semaphores to interact between threads.	**Software**	**5**	**1, 2, 3, 4**
13.	Find coverage by writing cover groups for a design.	**Software**	**6**	**1, 2, 3, 4**
14.	Implementation of parallel processes using Fork Join/ join_any / join_none statement.	**Software**	**6**	**1, 2, 3, 4**
15.	Create a layered test-bench for a simple design like Adder or Memory module (Case study)	**Software**	**-**	**1, 2, 3, 4**

Recommended Books:

[1]. J. Bhaskar "A Verilog HDL Primer" 3rd Edition Star Galaxy Publishing 2018.

[2]. Janick Bergeron, "Writing Testbenches Using System Verilog", Springer 2006.

[3]. Verilog Language Reference manual

[4]. Spartan and Virtex family user manuals from Xilinx.

Laboratory Assessment:

Assessment:

Term Work: Term Work shall consist of at least 10 to 12 practicals' based on the above list. Also, Term work Journal must include at least 2 assignments.
Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)
Practical/ Oral Exam: An Oral examination will be held based on the above syllabus.

Term Work:

At least 10 experiments covering the entire syllabus of PCEL 602 (ASIC Design & Verification Lab) should be set to have well predefined inference and conclusion. The experiments should be student centric and attempts should be made to make the experiments meaningful and interesting. Experiments must be graded from time to time. The grades should be converted into marks as per the Credit and Grading System manual and should be added and averaged. The grading and term work assessment should be done based on this scheme. The final certification and acceptance of term work ensure satisfactory performance of laboratory work and minimum passing marks in term work. Practical and Oral exam will be based on the entire syllabus.

Note:

Suggested List of Experiments is indicative. However, flexibility lies with individual course instructors to design and introduce new, innovative and challenging experiments, (limited to maximum 30% variation to the suggested list) from within the curriculum, so that the fundamentals and applications can be explored to give greater clarity to the students and they can be motivated to think differently.

Course Code	Course Name	Teaching Scheme (Hrs/week)			Credits Assigned			
22761132	**Natural Language Processing**	L	T	P	L	T	P	Total
		2	--	--	2	--	1	3
		Examination Scheme						
			IA1	**IA2**	**ESE**		**Total**	
		Theory	20	20	60		100	

Pre-requisite Course Codes	VSE 401 Creative Coding with python,	
Course Outcomes	After the successful completion students should be able to	
	CO1	Apply foundational NLP and deep learning concepts, along with basic text-processing techniques, to analyze language data and develop simple intelligent NLP application
	CO2	Apply finite-state techniques and N-gram models to analyze morphology and evaluate language models effectively.
	CO3	Analyze POS tagging methods and parsing techniques to interpret and construct accurate syntactic structures of natural language text.
	CO4	Interpret semantic and discourse techniques to understand text meaning and resolve references.
	CO5	Describe key LLM architectures and evaluation methods used in modern NLP systems.
	CO6	Identify major NLP applications and modern LLMs used for real-world language understanding tasks.

Module No.	Unit No.	Topics	Reference	Hrs.
Module 1	**1**	**Introduction to Deep Learning for NLP**		10 Hrs
	1.1	Biological neuron, Mc-Culloch Pitts Neuron, Perceptron Learning rule, Delta learning rule, Multilayer Perceptron: Linearly separable, linearly non-separable classes. Deep Networks: Fundamentals, Three Classes of Deep Learning, basic Terminologies of Deep Learning.	1	
	1.2	Training, Optimization and Regularization of Deep Neural Network Training Feed forward DNN Multi Layered Feed Forward Neural Network, Learning Factors, Activation functions, Loss functions.		
	1.3	CNN: Convolution Operation, Basic structure of a convolutional neural network: Padding, strides, pooling, fully connected layers, interleaving between layers. Training a convolutional network: Convolution/backpropagation as matrix multiplication.		
	1.4	Recurrent Neural Networks (RNN): Sequence Learning Problem, Unfolding Computational graphs, Recurrent Neural Network, Bidirectional RNN, Backpropagation Through Time (BTT), Vanishing and Exploding Gradients, Truncated BTT.		
Module 2	**2**	**Introduction to NLP and Text Processing**		5 Hrs
	2.1	What is Natural Language Processing (NLP)? History of NLP, Generic NLP System, Levels of NLP, Ambiguity and Layers of NLP. Challenges of NLP, Processing Indian Languages, Applications of NLP.	2,3	
	2.2	Basic Text Processing: Tokenization, Stopword Removal, Stemming and Lemmatization, Regular Expressions		
Module 3	**3**	**Word Level Analysis**		5 Hrs
	3.1	Survey of English Morphology, Inflectional Morphology, Derivational Morphology.	2,3	
	3.2	Morphological Models: Dictionary lookup, finite state morphology; Morphological parsing with FST (Finite State Transducer); Lexicon free FST Porter Stemmer algorithm		

	3.3	Language models- N-gram models and smoothing techniques: Laplace smoothing, Good-Turing Discounting, Evaluating language models- perplexity, cross-entropy		
Module 4	**4**	**Syntax Analysis**	2, 3, 4	8 Hrs
	4.1	Part-of-speech Tagging: Illustration of Ambiguity in POS, Table Look-up-based and Rule-based POS, Statistical POS Tagging: Hidden Markov Models (HMM) for POS tagging, Viterbi decoding algorithm; Chunking		
	4.2	Context-Free Grammars (CFG) and parse trees; Linguistics of Parsing		
	4.3	Algorithmics of Parsing, Constituency Parsing: Rule Based (Top-Down, Bottom-Up, CYK)		
	4.4	Statistical Parsing, Dependency Parsing and Neural Parsing		
Module 5	**5**	**Semantic and Discourse Processing**		6 Hrs
	5.1	Introduction, Meaning Representation; Lexical Semantics; Study of Various language dictionaries like WordNet, Lexical Relations	3, 5, 6	
	5.2	Word Embeddings: Word2Vec, CBOW, Skip-gram, GloVe		
	5.3	Word Sense Disambiguation (WSD) methods: Lesk algorithm Document representation: TF-IDF, Doc2Vec		
	5.4	Discourse Structure, Cohesion, Discourse Coherence & Structure, Reference resolution, Reference phenomena, Anaphora resolution using Hobbs algorithm		
Module 6	**6**	**Modern NLP with Transformers and Applications**		5 Hrs
	6.1	Introduction to Large Language Models (LLM), Fundamental components of the Transformer model	4, 5, 6	
	6.2	Transformer Architecture and Self-Attention Mechanism, Encoder–Decoder architecture, Introduction to pre-training and fine-tuning paradigms, Pretrained models (BERT, GPT, T5) and fine-tuning, Positional Encoding. Evaluation metrics (BLEU, ROUGE, F1)		
	6.3	Named Entity Recognition (NER), Question Answering System, Machine Translation.		
Total				**39**

<u>Course Assessment:</u>

Theory:
<u>IA1:</u> 20 Marks written examination for one hour
<u>IA2:</u> 20 Marks written examination for one hour
<u>ESE:</u> 60 Marks written examination for two hours

Recommended Books:

[1] **Charu C. Aggarwal**, Neural Networks and Deep Learning: A Textbook, 2nd Edition, Springer, 2023.
[2] Daniel Jurafsky, James H. and Martin, Speech and Language Processing, Third Edition, Prentice Hall, 2026.
[3] Christopher D.Manning and HinrichSchutze, Foundations of Statistical Natural Language Processing, MIT Press, 1999.
[4] Pushpak Bhattacharyya and Aditya Joshi, Natural Language Processing, Wiley India Pvt. Ltd., Edition: 2023
[5] Dr. Tanmoy Chakraborty, “Introduction to Large Language Models: Generative AI for Text”, Wiley (Wiley India). December 2024.
[6] Palash Goyal, Sumit Pandey, Karan Jain. Deep Learning for Natural Language Processing-2018.
[7] Joseph Babcock, Raghav Bali, “Generative AI with Python and TensorFlow”, Packt Publishing 2021.

Online References:

1. NPTEL Course”Natural Language Processing”: https://onlinecourses.nptel.ac.in/noc23_cs45/preview
2. NPTEL Swayam “Natural Language Processing” : https://archive.nptel.ac.in/courses/106/106/106106211/

<table>
<tr><td rowspan="2">Course Code</td><td rowspan="2">Course Name</td><td colspan="3">Teaching Scheme (Hrs/week)</td><td colspan="4">Credits Assigned</td></tr>
<tr><td>L</td><td>T</td><td>P</td><td>L</td><td>T</td><td>P</td><td>Total</td></tr>
<tr><td rowspan="4">22761172</td><td rowspan="4">Natural Language Processing Lab</td><td>--</td><td>--</td><td>2</td><td>--</td><td>--</td><td>1</td><td>1</td></tr>
<tr><td colspan="7">Examination Scheme</td></tr>
<tr><td colspan="2">Term work</td><td colspan="3">Orals</td><td colspan="2">Total</td></tr>
<tr><td colspan="2">25</td><td colspan="3">25</td><td colspan="2">50</td></tr>
</table>

<table>
<tr><td>Pre-requisite Course Codes</td><td colspan="2">VSE 401 Creative Coding with python</td></tr>
<tr><td rowspan="6">Laboratory Objectives</td><td>1.</td><td>To understand deep learning models for classification and sequence learning problems.</td></tr>
<tr><td>2</td><td>To familiarize with linguistic analysis through morphology, N-grams, and chunking.</td></tr>
<tr><td>3.</td><td>To build statistical analysis for N-grams and parsing.</td></tr>
<tr><td>4.</td><td>To understand semantic analysis and document similarity techniques</td></tr>
<tr><td>5.</td><td>To learn NLP systems using standard classification techniques.</td></tr>
<tr><td>6.</td><td>To understand large language models for understanding architectural variations.</td></tr>
<tr><td rowspan="7">Laboratory Outcomes</td><td colspan="2">After the successful completion students should be able to</td></tr>
<tr><td>LO1</td><td>Implement and analyze deep learning models for classification and sequence learning problems.</td></tr>
<tr><td>LO2</td><td>. Apply various text processing techniques for real-world NLP tasks.</td></tr>
<tr><td>LO3</td><td>Analyze linguistic structures using morphological processing, N-grams, and chunking techniques</td></tr>
<tr><td>LO4</td><td>Develop statistical and neural language models such as POS tagging, HMMs, and Pasers.</td></tr>
<tr><td>LO 5</td><td>Use TF-IDF and cosine similarity to extract semantics and document distances.</td></tr>
<tr><td>LO 6</td><td>Analyze and compare the performance, outputs, and capabilities of modern NLP models, including NER, BERT, and large language models.</td></tr>
</table>

Laboratory Experiments:

Sr. No.	Title of experiment	Hardware /Software	Mod ule	Refere nce
1.	**Study of Activation Functions used in Deep Learning. ii.Implementation of Feed Forward Neural Network using Python**	**Software**	**1**	**1**
2.	**Basic Text Preprocessing Techniques**	**Software**	**2**	**2,3**
3.	**Perform morphological analysis on input text to extract root words and morphemes, and generate new word forms using inflectional and derivational rules.**	**Software**	**2**	**2,3**

4.	**Extraction of Email Addresses and Indian Mobile Numbers Using Regular Expressions**	**Software**	**2**	**3**
5.	**Perform chunking (shallow parsing) on text to extract phrases like noun phrases (NP) and verb phrases (VP)**	**Software**	**3**	**4**
6	**Study of Part-of-Speech Taggers and POS Tagging**	**Software**	**3**	**4**
7	**Perform Word Sense Disambiguation Using Lesk Algorithm**	**Software**	**4**	**4**
8.	**Word Embeddings Using Word2Vec and GloVe**	**Software**	**4**	**4,5**
9.	**Demonstrate TF-IDF and Document Similarity Analysis**	**Software**	**4**	**4,5**
10.	**Study of Transformer Architecture and Self-Attention Mechanism**	**Software**	**5**	**4,5**
11.	**Text Classification Using BERT Fine-Tuning**	**Software**	**5**	**4**
12.	**Named Entity Recognition (NER) on Text Data**	**Software**	**6**	**5**
13.	**Exploration of Modern NLP Models (GPT-4, Llama-3, Claude-3, Mistral, Gemini)**	**Software**	**6**	**5**

Laboratory Assessment:

Assessment:

Term Work: Term Work shall consist of at least 10 to 12 practicals' based on the above list. Also, Term work Journal must include at least 2 assignments.
Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)
Practical/ Oral Exam: An Oral examination will be held based on the above syllabus.

Recommended Books:

[1] Charu C. Aggarwal, Neural Networks and Deep Learning: A Textbook, Second Edition, Springer, 2023.
[2] Daniel Jurafsky, James H. and Martin, Speech and Language Processing, Third Edition, Prentice Hall, 2026.
[3] Christopher D.Manning and HinrichSchutze, Foundations of Statistical Natural Language Processing, MIT Press, 1999.
[4] Pushpak Bhattacharyya and Aditya Joshi, Natural Language Processing, Wiley India Pvt. Ltd., Edition: 2023
[5] Dr. Tanmoy Chakraborty, "Introduction to Large Language Models: Generative AI for Text", Wiley (Wiley India). December 2024.

Term Work:

At least 10 experiments covering the entire syllabus of PCL 302 (Electronic Devices and circuits Lab) should be set to have well predefined inference and conclusion. This must include **50%** Hardware and **50%** Simulation experiments. The experiments should be student centric and attempts should be made to make the experiments meaningful and interesting. Experiments must be graded from time to time. The grades should be converted into marks as per the Credit and Grading System manual and should be added and averaged. The grading and term work assessment should be done based on this scheme. The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work. Practical and Oral exam will be based on the entire syllabus.

Note:
Suggested List of Experiments is indicative. However, flexibility lies with individual course instructors to design and introduce new, innovative and challenging experiments, (limited to maximum 30% variation to the suggested list) from within the curriculum, so that the fundamentals and applications can be explored to give greater clarity to the students and they can be motivated to think differently.

Course Code	Course Name	Teaching Scheme (Hrs/week)			Credits Assigned			
22761133	**Advanced Database Management Systems**	L	T	P	L	T	P	Total
		3	--	--	3	--	--	3
		Examination Scheme						
			IA1	IA2	ESE	Total		
		Theory	20	20	60	100		

Pre-requisite Course Codes	PC 304 Analysis of Algorithm PC 402 Discrete Structure and Automata Theory PE 501 – Database Management Systems	
Course Outcomes	After the successful completion students should be able to	
	CO1	Explain query processing steps and apply optimization strategies to improve query execution in DBMS.
	CO2	Apply database security mechanisms DAC, MAC, and RBAC — and compare their access control models.
	CO3	Apply indexing structures (B-Trees, B+ Trees) and hashing techniques to organize and retrieve data efficiently
	CO4	Explain the architecture and classify parallelism types in parallel database systems
	CO5	Apply distributed database design principles including fragmentation and analyze query optimization in distributed environments.
	CO6	Classify and explain emerging database technologies such as NoSQL, graph, temporal, spatial, and cloud databases with their use cases.

Module No.	Unit No.	Topics	Reference	Hrs.
Module 1	**1**	**Query Processing and Optimization**	**1,2**	**7 Hrs**
	1.1	Query Processing Overview: Introduction, Query Processing in DBMS, Steps of Query Processing, Measures of Query Cost, Selection Operation, Sorting, Join Operation, Evaluation of Expressions.		
	1.2	Query Optimization: Overview and Goals of Query Optimization, Approaches of Query Optimization, Transformation of Relational Expressions, Estimating Statistics of Expression Results, Choice of Evaluation Plans.		
Module 2	**2**	**Indexing and Hashing**	**1,2**	**7 Hrs**
	2.1	Indexing Techniques: Basic Concepts of Indexing, Ordered Indices: Single-Level and Multi-Level Indexes, Indexed Sequential Access Method (ISAM), B-Tree and		

		B+ Tree Structure, Format of a Node, Search, Insert and Delete Operations in B+ Trees.		
	2.2	Hashing Techniques: Static and Dynamic Hashing, Multiple-Key Access and Bitmap Indexes, Comparison of Ordered Indexing and Hashing, Index Definition in SQL.		
Module 3	**3**	**Database Security and Advanced Access Protocols**	**1,3**	**5 Hrs**
	3.1	Database Security: Database Security Overview, Discretionary Access Control (DAC), Mandatory Access Control (MAC), Role-Based Access Control (RBAC), Database Auditing and Encryption Techniques.		
	3.2	Advanced Access Protocols: Remote Database Access Protocols, Introduction to Mobile Databases and their Characteristics.		
Module 4	**4**	**Parallel Databases**	**1,2**	**6 Hrs**
	4.1	Parallel Database Systems: Introduction to Parallel Databases, Architectures for Parallel Systems, I/O Parallelism and Data Partitioning Techniques, Interquery, Intraquery and Intraoperation Parallelism.		
	4.2	Parallel Database Operations: Parallelizing Database Operations — Bulk Loading, Scanning, Sorting and Joins, Design Challenges in Parallel Databases.		
Module 5	**5**	**Distributed Databases**	**2,3**	**8 Hrs**
	5.1	Introduction and Design Issues: Introduction to Distributed Database Systems, Distributed Data Processing, Advantages of Distributed Databases, Design Issues in Distributed DBMS, Distributed DBMS Architecture, Homogeneous and Heterogeneous Distributed DBMS.		
	5.2	Distributed Database Design and Query Processing: Top-Down Design Process, Fragmentation — Horizontal, Vertical and Mixed Fragmentation, Correctness Rules of Fragmentation, Replicated and Non-Replicated Allocation. Query Processing Overview, Objectives of Query Processing, Communication Cost, Layers of Query Processing, Query Optimization in Distributed Databases.		
Module 6	**6**	**Emerging Trends in Databases**	**2,3**	**6 Hrs**
	6.1	NoSQL and Graph Databases: Introduction to NoSQL Databases, Need of NoSQL, CAP Theorem, Types of NoSQL Databases, Introduction to MongoDB Architecture and CRUD Operations. Graph Databases: Introduction, Features, Transactions, Consistency, Availability, Querying, Case Study of Neo4j.		
	6.2	Advanced and Emerging Database Technologies: Temporal Databases — Concepts, Time Representation, Time Dimension, Incorporating Time in Relational Databases. Spatial Databases: Introduction, Data Types, Models, Operators and Queries. Introduction to Big Data, Introduction to Cloud Databases and DBaaS.		
Total	**39 Hrs**			

Course Assessment — Theory

IA1: 20 Marks written examination for one hour

IA2: 20 Marks written examination for one hour

ESE: 60 Marks written examination for two hours

Recommended Books

[1] A. Silberschatz, H. F. Korth & S. Sudarshan, "Database System Concepts", McGraw Hill, 7th Edition, 2019. — Modules 1, 2, 3, 4

[2] R. Elmasri & S. B. Navathe, "Fundamentals of Database Systems", Addison Wesley, 7th Edition, 2016. — Modules 1, 3, 5, 6 (Temporal & Spatial)

[3] Raghu Ramakrishnan & Johannes Gehrke, "Database Management Systems", McGraw-Hill, 3rd Edition, 2003. — Modules 1, 2, 3

[4] M. Tamer Ozsu & Patrick Valduriez, "Principles of Distributed Database Systems", Springer, 3rd Edition, 2011. — Module 5

[5] Kristina Chodorow, "MongoDB: The Definitive Guide", O'Reilly Media, 3rd Edition, 2019. — Module 6 (NoSQL/MongoDB)

[6] Ian Robinson, Jim Webber & Emil Eifrem, "Graph Databases: New Opportunities for Connected Data", O'Reilly Media, 2nd Edition, 2015. — Module 6 (Graph Databases/Neo4j)

Online References

[1] NPTEL courses on DBMS: https://nptel.ac.in

[2] MongoDB Documentation: https://www.mongodb.com/docs/

[3] Neo4j Documentation: https://neo4j.com/docs/

Laboratory Course

<table>
<tr><th>Course Code</th><th>Course Name</th><th colspan="3">Teaching Scheme (Hrs/week)</th><th colspan="4">Credits Assigned</th></tr>
<tr><td rowspan="6">22761173</td><td rowspan="6">Advanced Database Management Systems Lab</td><td>L</td><td>T</td><td>P</td><td>L</td><td>T</td><td>P</td><td>Total</td></tr>
<tr><td>--</td><td>--</td><td>2</td><td>--</td><td>--</td><td>1</td><td>1</td></tr>
<tr><td colspan="7">Examination Scheme</td></tr>
<tr><td colspan="3">Term Work</td><td colspan="2">Orals</td><td colspan="2">Total</td></tr>
<tr><td colspan="3">25</td><td colspan="2">25</td><td colspan="2">50</td></tr>
</table>

<table>
<tr><td>Pre-requisite Course Codes</td><td colspan="2">PEL 501 – Database Management Systems Lab</td></tr>
<tr><td>Laboratory Objectives</td><td colspan="2">1. To analyze and implement query processing and optimization techniques.
2. To understand and apply database security models and access control mechanisms.
3. To implement B+ Tree indexing and hashing techniques for efficient data retrieval.
4. To explore parallel and distributed database operations.
5. To get hands-on experience with NoSQL databases (MongoDB) and graph databases (Neo4j).</td></tr>
<tr><td>Laboratory Outcomes</td><td colspan="2">After the successful completion students should be able to</td></tr>
<tr><td></td><td>LO1</td><td>Trace and compare query execution plans and apply optimization techniques to reduce query cost. [L3, L4]</td></tr>
</table>

	LO2	Apply and differentiate database security models — DAC, MAC, and RBAC — using SQL commands. [L3, L4]
	LO3	Apply B+ Tree indexing and hashing techniques to implement efficient data storage and retrieval. [L3]
	LO4	Explain data partitioning and apply distributed fragmentation strategies in database design. [L2, L3]
	LO5	Apply NoSQL (MongoDB) and graph database (Neo4j) operations for data management tasks. [L3]
	LO6	Explain temporal and spatial database concepts and apply them through structured queries. [L2, L3]

Laboratory Experiments

Sr. No.	Title of Experiment	Hardware/Software	Module	Reference
1.	Study and trace the steps of query processing: parsing, optimization, and execution using EXPLAIN / EXPLAIN ANALYZE in SQL.	Software	1	1, 2
2.	Apply relational algebra transformation rules for query optimization and evaluate query cost estimates.	Software	1	1, 2
3.	Implement B+ Tree structure: perform search, insert, and delete operations and trace node formats.	Software	2	1, 2
4.	Apply static and dynamic hashing techniques; compare performance with B+ Tree indexed access.	Software	2	1, 2
5.	Implement Discretionary Access Control (DAC) and Role-Based Access Control (RBAC) using SQL GRANT and REVOKE.	Software	3	1, 3
6.	Apply database auditing: log and monitor DML operations on sensitive tables using triggers and audit trails.	Software	3	1, 3
7.	Simulate I/O parallelism using data partitioning techniques: range partitioning, hash partitioning, and round-robin partitioning.	Software	4	1, 2
8.	Apply distributed database design: implement horizontal, vertical, and mixed fragmentation with allocation strategies.	Software	5	2, 3
9.	Apply distributed query processing: evaluate communication cost and layers of query optimization in distributed DBMS.	Software	5	2, 3
10.	Apply MongoDB CRUD operations, aggregation pipelines, and index creation on a document collection.	Software	6	2, 3
11.	Apply graph database operations using Neo4j: create nodes and relationships, and execute Cypher queries for graph traversal.	Software	6	2, 3
12.	Apply temporal database concepts: write queries using time dimensions and bi-temporal data management.	Software	6	2, 3

Laboratory Assessment

Term Work: Term Work shall consist of at least 10 to 12 practicals based on the above list. Term work Journal must include at least 2 assignments.

Term Work Marks: 25 Marks (Total) = 15 Marks (Experiments) + 5 Marks (Assignments) + 5 Marks (Attendance)

Practical/Oral Exam: An Oral examination will be held based on the above syllabus.

Recommended Books — Laboratory

[1] A. Silberschatz, H. F. Korth & S. Sudarshan, "Database System Concepts", McGraw Hill, 7th Edition, 2019. — Exps 1, 2, 3, 4, 7

[2] M. Tamer Ozsu & Patrick Valduriez, "Principles of Distributed Database Systems", Springer, 3rd Edition, 2011. — Exps 8, 9

[3] Kristina Chodorow, "MongoDB: The Definitive Guide", O'Reilly Media, 3rd Edition, 2019. — Exp 10

[4] Ian Robinson, Jim Webber & Emil Eifrem, "Graph Databases", O'Reilly Media, 2nd Edition, 2015. — Exp 11

[5] R. Elmasri & S. B. Navathe, "Fundamentals of Database Systems", Addison Wesley, 7th Edition, 2016. — Exps 5, 6, 12

Term Work

At least 10 experiments covering the entire syllabus of PEL 602 (Advanced Database Management Systems Lab) should be set with well-defined inference and conclusion. The experiments should be student-centric and attempts should be made to make them meaningful and interesting. Experiments must be graded from time to time. The grades should be converted into marks as per the Credit and Grading System manual. The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work. Practical and Oral exam will be based on the entire syllabus.

Note

Suggested List of Experiments is indicative. However, flexibility lies with individual course instructors to design and introduce new, innovative and challenging experiments (limited to maximum 30% variation to the suggested list) from within the curriculum, so that the fundamentals and applications can be explored to give greater clarity to the students and they can be motivated to think differently.

Course Code	Course Name	Teaching Scheme (Hrs/week)			Credits Assigned			
22761134	Network Security	L	T	P	L	T	P	Total
		3	--	2	3	--	1	4
		Examination Scheme						
			IA1	IA2	ESE		Total	
		Theory	20	20	60		100	

Pre-requisite Course Codes	Computer Network, Cloud Computing	
Course Outcomes	After the successful completion students should be able to	
	CO1	Explain fundamental concepts of network security and the security goals
	CO2	Apply cryptographic techniques for securing information and communication.
	CO3	Identify various network attacks, and evaluate their impact on systems and users.
	CO4	Interpret and analyze the working of secure communication protocols and mechanisms
	CO5	Implement and evaluate network security mechanisms applied to strengthen organizational defenses, and to reduce risk factors.
	CO6	Assess security challenges in cloud, wireless and web environments

Module No.	Unit No.	Topics	Reference	Hrs.
Module 1	**1**	**Introduction**		4Hrs
	1	Introduction, Security goals, CIA triad (Confidentiality, Integrity, Availability), Threats, vulnerabilities, attacks, Examples of passive vs active attacks, vulnerability analysis in network models, Security policies and risk assessment	1,2	
Module 2	**2**	**Cryptography Essentials**		6Hrs
	2	Symmetric encryption concepts (AES, DES), Asymmetric encryption (RSA), Hashing (SHA), digital signatures, Key Management, Key exchange (Diffie–Hellman), Certificates and PKI, VPN fundamentals	1,2	
Module 3	**3**	**Network Attacks & Defenses**	1,2	8Hrs
	3.1	Malwares: Virus, Worms, Trojans, Ransomware, DoS/DDoS attacks, Defense Mechanisms against DDoS attacks, Man-in-the-Middle (MITM), SYN flood attacks, Smurf attacks, ARP spoofing, DNS poisoning, Packet sniffing, Phishing & social engineering		
	3.2	Cyber crimes and Cyber Security: Classifications of cyber crimes, Botnets, Mobile phones and security Implications, Attacks on mobile devices – Viruses, Mishing, Vishing, Smishing		

Module 4	**4**	**Network Security Mechanisms**		7Hrs
	4.1	Authentication and access control: Password security, Multifactor authentication, Firewall – types and uses, Intrusion Detection/Prevention Systems (IDS/IPS), Access control Lists (ACLs), Network segmentation, Zero Trust concept, Honeypots and Honeynets	1	
	4.2	Cybercrimes: Legal perspective, Indian ITA 2000 and its amendments, Global perspective on cybercrimes		
Module 5	**5**	**Secure Network Protocols**		6Hrs
	5.1	Working and handshake overview of SSL/TLS, HTTPS vs HTTP, Secure email, Overview of S/MIME and PGP, Secure DNS, IPSec, SSL and VPN concepts	1,3	
	5.2	Security Monitoring: Log Analysis, Threat Intelligence		
Module 6	**6**	**Wireless & Web Security**		8Hrs
	6.1	Web vulnerabilities: SQL Injection, Cross Site scripting (XSS), Cross Site Request Forgery (CSRF), Wi-Fi security (WEP, WPA, WPA2, WPA3),	1,2	
	6.2	Secure Coding Practices, API security basics Introduction to Cloud security basics, risks, threats to cloud infrastructure, Cloud service provider risks Security in IoT		
			Total	**39**

<u>Course Assessment:</u>

Theory:

<u>IA1:</u> 20 Marks written examination for one hour
<u>IA2:</u> 20 Marks written examination for one hour

<u>ESE:</u> 60 Marks written examination for two hours

Recommended Books:

[1] William Stallings, Network Security Essentials: Applications and Standards, Pearson Education, 6th Edition, 2017.
[2] Atul Kahate, Cryptography and Network Security, McGraw Hill Education, 4th Edition, 2019.
[3] Behrouz A. Forouzan, Cryptography and Network Security, McGraw Hill Education, 1st Edition, 2007.
[4] Nina Godbole and Sunit Belapure, Cyber Security: Understanding Cyber Crimes, Computer Forensics and Legal Perspectives, Wiley India Pvt. Ltd., 1st Edition, 2011.
[5] Ronald L. Krutz and Russell Dean Vines, Cloud Security: A Comprehensive Guide to Secure Cloud Computing, Wiley, 1st Edition, 2010.

Online References:

[1] NPTEL courses on Network Security

<table>
<tr><th rowspan="2">Course Code</th><th rowspan="2">Course Name</th><th colspan="3">Teaching Scheme (Hrs/week)</th><th colspan="4">Credits Assigned</th></tr>
<tr><td>L</td><td>T</td><td>P</td><td>L</td><td>T</td><td>P</td><td>Total</td></tr>
<tr><td rowspan="5">22761174</td><td rowspan="5">Network Security Lab</td><td>--</td><td>--</td><td>2</td><td>--</td><td>--</td><td>1</td><td>1</td></tr>
<tr><td colspan="7">Examination Scheme</td></tr>
<tr><td colspan="2">Term work</td><td colspan="3">Orals</td><td colspan="2">Total</td></tr>
<tr><td colspan="2">25</td><td colspan="3">25</td><td colspan="2">50</td></tr>
</table>

Pre-requisite Course Codes		
Laboratory Objectives	1.	To analyze Packets using Wireshark
	2.	To implement Basic Encryption & Hashing
	3.	Perform Password Security Experiments
	4.	Perform Vulnerability Scanning
Laboratory Outcomes	After the successful completion students should be able to	
	LO1	Use network monitoring and packet analysis tools to examine network communication and troubleshoot basic networking and security issues.
	LO2	Implement and test cryptographic techniques and secure key exchange using appropriate tools
	LO3	Analyze common network and web security attacks including ARP spoofing and SQL Injection, and identify suitable detection and mitigation mechanisms.
	LO4	Configure and evaluate security mechanisms

Laboratory Experiments:

Sr. No.	Title of experiment	Hardware /Software	Module	Reference
1.	**Study of basic network configuration commands (ping, ipconfig, tracert, netstat)**	**Software**	**1**	
2.	**Packet capturing and analysis using Wireshark**	**Software**	**2**	
3.	**Implement AES and RSA encryption/decryption in Python/Java.**	**Software**	**2**	
4.	**Demonstrate Diffie–Hellman key exchange using a coding exercise**	**Software**	**3**	
5.	**Configure a firewall (iptables or pfSense) and/or set up Snort IDS to detect suspicious traffic.**	**Software**	**3**	
6	**Detection and Analysis of ARP Spoofing Attack using Wireshark**	**Software**	**3**	

7	**Password strength analysis and authentication mechanisms**	**Software**	**4**	
8.	**Secure file transfer using SFTP/SSH**	**Software**	**4**	
9.	**Demonstrate SQL Injection and Detection**	**Software**	**4**	
10.	**Digital signature generation and verification**	**Software**	**5**	

Laboratory Assessment:

Assessment:
Term Work: Term Work shall consist of at least 8 practical based on the above list. Also, Term work Journal must include at least 2 assignments.
Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)
Practical/ Oral Exam: An Oral examination will be held based on the above syllabus.

Recommended Books:

Term Work:

At least 8 experiments covering the entire syllabus of PEL should be set to have well predefined inference and conclusion. The experiments should be student centric and attempts should be made to make the experiments meaningful and interesting. Experiments must be graded from time to time. The grades should be converted into marks as per the Credit and Grading System manual and should be added and averaged. The grading and term work assessment should be done based on this scheme. The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work. Practical and Oral exam will be based on the entire syllabus.

Note:

Suggested List of Experiments is indicative. However, flexibility lies with individual course instructors to design and introduce new, innovative and challenging experiments, (limited to maximum 30% variation to the suggested list) from within the curriculum, so that the fundamentals and applications can be explored to give greater clarity to the students and they can be motivated to think differently.

Course Code	Course Name	Teaching Scheme (Hrs/week)			Credits Assigned			
		L	T	P	L	T	P	Total
		3	--	--	3	--	--	3
22761141	**MEMS & Automation**	**Examination Scheme**						
			IA1	**IA2**	**ESE**		**Total**	
		Theory	20	20	60		100	

Pre-requisite Course Codes	PC 302 – Electronics Device PC 303 – Digital System Design PC 402 – Electronics Circuit & Design PC 403 – Physics of Semiconductor Devices
Course Objective	• To introduce semiconductor manufacturing environment, wafer preparation, and crystalline properties used in VLSI technology. • To explain semiconductor fabrication processes involved in integrated circuit manufacturing and emphasis on CMOS based integrated circuit fabrication flow, and layout design principles. • To provide knowledge of micromachining techniques and selecting suitable material for MEMS fabrication. • To familiarize students with modern semiconductor industry practices, for characterization of IC/MEMS, its sustainable packaging approaches, reliability, testing techniques and real-world engineering applications.
Course Outcomes: After successful completion of the course, student will be able to	
CO1	Differentiate IC and MEMS microfabrication technologies based on materials, scaling, structure, function, and applications.
CO2	Illustrate semiconductor and MEMS microfabrication processes and interpret the effect of fabrication parameters on process characteristics. .
CO3	Apply fabrication process sequences, layout mask concepts, and microfabrication technologies used in integrated microelectronic devices.
CO4	Describe MEMS layout mask concepts, material selection, and micromachining process flows for MEMS devices.
CO5	Illustrate the characteristics, packaging, and reliability aspects of semiconductor and MEMS devices based on industry-standard and sustainable engineering practices.

Module No.	Unit No.	Topics	Ref.	Hrs.
1		**Introduction to Microfabrication**		06
	1.1	**Overview of microfabrication and its role in VLSI/MEMS**: Moore's Law, Scaling effects (electrical, mechanical, thermal, etc.), Historical evolution, applications in ICs, sensors, actuators, VLSI Technology Vs MEMS Technology, ICs Vs MEMS Devices	1,3	
	1.2	**Materials for microfabrication its Properties:** Materials like (but not limited to) Si, SiO_2, SiN, SU8, PMMA, emerging materials like Graphene, Carbon Nano Tube	3	
2		**Substrate Preparation and Thin Film Deposition**		06
	2.1	**Substrate Preparation:** Crystal growth (Czochralski), wafer slicing/polishing	1,2	
	2.2	**Doping:** Thermal diffusion and Ion Implantation, annealing	1,2	
	2.3	**Thin film Deposition:** Evaporation, Sputtering and Chemical Vapor Deposition, Epitaxy, Thermal oxidation	1,2	
3		**Photolithography, and Patterning**		06
	3.1	**Introduction to Lithography:** Cleanroom classifications, contamination control, and safety, Photoreactive materials, photolithography process, mask preparation, advanced lithography types: Electron beam, Ion beam and X-ray lithography	1,2	
	3.2	**Etching:** Wet etching, Dry etching: Plasma/RIE, DRIE, ion milling	1,2	
4		**Process Integration**		09
	4.1	**Fabrication process sequence for:** integrated resistors, capacitors, and inductors; standard bipolar transistors and advanced bipolar devices; MOSFETs with emphasis on CMOS and memory devices; high-performance MESFETs and monolithic microwave integrated circuits (MMICs).	2	
	4.1	**CMOS IC Process Flow:** N well, P-well and Twin tub, Design rules, Layout of MOS based circuits (gates and combinational logic), Buried and Butting Contact	1,2	
	4.2	**Device Isolation and Metallization:** Junction and oxide isolation, LOCOS, trench isolation, Schottky contacts, Ohmic contacts, Metallization	1,2	
	4.4	MEMS fabrication using bulk, surface, and LIGA micromachining techniques with representative device examples.	2,3	
5		**Semiconductor and MEMS Device Characterization**		06
	5.1	**Semiconductor material characterization:** Resistivity, carrier concentration, mobility, sheet resistance, oxide thickness and crystal defects	1	
	5.2	**MEMS device characterization:** Device dimensions, piezoresistance, TCR, stiffness, adhesion, vibration and resonant frequency	3	
6		**IC and MEMS Packaging**		06
	6.1	**Introduction to semiconductor packaging:** Package functions, and package types, Package Types: DIP, QFP, BGA, CSP and Flip-chip packaging	4,5	

	6.2	**IC and MEMS packaging processes:** Die attach, wire bonding, wafer bonding, dicing and encapsulation	4,5	
	6.3	**Packaging materials and properties:** Molding compounds, bonding wires, solders, ceramics and heat spreaders	4,5	
	6.4	MEMS packaging requirements, reliability issues and packaging challenges	4,5	
			Total	**39**

Course Assessment:

Theory:

IA1: 20 Marks written examination for one hour
IA2: 20 Marks written examination for one hour

ESE: 60 Marks written examination for two hours

Text Books:

1. James D. Plummer, Michael D. Deal and Peter B. Griffin, "Silicon VLSI Technology", Pearson Education, First Indian Edition, 2000.

2. Simon M. Sze and Ming-Kwei Lee, "Semiconductor Devices: Physics and Technology", Wiley India, Third Edition, 2012.

3. James J. Allen, "Micro Electro Mechanical System Design", CRC Press, e-book Edition, 2005.

4. William J. Greig, "Integrated Circuit Packaging, Assembly and Interconnections", Springer, First Edition, 2007.

5. Andrea Chen and Randy Hsiao-Yu Lo, "Semiconductor Packaging: Materials, Interaction and Reliability", CRC Press.2012.

Reference Books:

1. Gary S. May and Simon M. Sze, "Fundamentals of Semiconductor Fabrication", Wiley India, First Edition, 2011

2. Tai-Ran Hsu, "MEMS and Microsystems: Design, Manufacture, and Applications", McGraw Hill Education, First Edition, 2017

Online References:

1. **https://nptel.ac.in/courses/117106093**

2. **https://nptel.ac.in/courses/117105082**

Term Work:

To strengthen the course, a compulsory visit to microfabrication laboratory at reputed institutes like IIT may be planned. Moreover, a mini case study or literature review on semiconductor manufacturing, integrated IC–MEMS systems, MEMS applications, packaging technologies, or device reliability may be included to enhance industry awareness and real-world understanding. Additionally, one lecture on sustainable semiconductor manufacturing and advanced packaging trends may be incorporated to address environmental and societal considerations in modern semiconductor industries.

<table>
<tr><th rowspan="2">Course Code</th><th rowspan="2">Course Name</th><th colspan="3">Teaching Scheme (Hrs/week)</th><th colspan="4">Credits Assigned</th></tr>
<tr><td>L</td><td>T</td><td>P</td><td>L</td><td>T</td><td>P</td><td>Total</td></tr>
<tr><td rowspan="5">22761181</td><td rowspan="5">MEMS & Automation Lab</td><td>--</td><td>--</td><td>2</td><td>--</td><td>--</td><td>1</td><td>1</td></tr>
<tr><td colspan="7">Examination Scheme</td></tr>
<tr><td></td><td>IA1</td><td>IA2</td><td>Term Work</td><td>Practical & Oral</td><td colspan="2">ESE Total</td></tr>
<tr><td>Lab</td><td>--</td><td>--</td><td>25</td><td>--</td><td colspan="2">25</td></tr>
</table>

<table>
<tr><td>Pre-requisite Course Codes</td><td>PC 302 – Electronics Device
PC 303 – Digital System Design
PC 402 – Electronics Circuit & Design
PC 403 – Physics of Semiconductor Devices</td></tr>
<tr><td>Course Objective</td><td>● To introduce semiconductor and MEMS microfabrication technologies, materials, scaling concepts, and process integration techniques used in integrated circuits and microsystems.
● To explain substrate preparation, doping, thin-film deposition, lithography, etching, and micromachining processes used in semiconductor and MEMS fabrication.
● To describe fabrication process flows, layout mask concepts, and microfabrication technologies used for passive and active integrated microelectronic devices.
● To explain MEMS fabrication using micromachining techniques, material selection, and representative MEMS device structures.
● To introduce semiconductor and MEMS device characterization methods, packaging technologies, reliability issues, and sustainable engineering practices relevant to microfabrication industries.</td></tr>
<tr><td colspan="2">Course Outcomes: After successful completion of the course, student will be able to</td></tr>
<tr><td>CO1</td><td>Simulate semiconductor and MEMS microfabrication processes and analyze the effect of process parameters on fabrication characteristics.</td></tr>
<tr><td>CO2</td><td>Perform characterization of semiconductor materials and micromachined structures to determine its important parameters and compare with theoretical parameters.</td></tr>
<tr><td>CO3</td><td>Design and verify layout masks and fabrication process flows for microelectronic devices using CAD/TCAD tools.</td></tr>
<tr><td>CO4</td><td>Analyze IC and MEMS packaging techniques and evaluate packaging effects on device performance, thermal behavior, and reliability using simulation and literature studies.</td></tr>
</table>

Suggested List of Experiments:

Sr. No.	Title
1	To simulate and analyze thermal oxidation of silicon using the Deal–Grove model and study the effect of process parameters on oxide growth characteristics.
2	To simulate dopant diffusion in semiconductor substrates and analyze the influence of diffusion parameters on impurity concentration profiles.
3	To determine electrical properties of semiconductor materials using virtual Hall Effect and Four Probe experimental setups.
4	**Simulation for CMOS IC Technology:** ● To design, simulate, and verify CMOS based combinational digital circuit layouts using industry-standard VLSI CAD tools. ● To design, simulate, and verify CMOS based sequential digital circuits using industry-standard VLSI CAD tools and evaluate their timing and layout characteristics. ● To design, simulate, and analyze CMOS arithmetic circuits such as adders and ALUs using industry-standard VLSI CAD tools.
5	**Simulation for Bipolar IC Technology:** ● To simulate and analyze semiconductor device characteristics such as PN junctions using industry-standard TCAD tools. ● To design, simulate, and evaluate MOS structures and MOSFET devices using TCAD tools and study their electrical characteristics. ● To simulate and analyze bipolar junction transistor structures using TCAD tools and evaluate their device performance parameters. ● To simulate and analyze semiconductor fabrication process steps such as oxidation, diffusion, and ion implantation using TCAD tools. ● To analyze short channel effects and scaling behavior in CMOS devices using TCAD simulation tools
6	**Simulation for Passive Components in IC Technology:** ● To design, simulate, and analyze integrated passive components such as resistors, capacitors, and interconnects using industry-standard VLSI CAD tools. ● To extract and evaluate parasitic resistance, capacitance, and inductance effects in VLSI interconnects and study their impact on circuit performance using simulation tools. ● To design and analyze MOS-based capacitors and diffusion/poly resistors and evaluate their electrical characteristics using industry-standard VLSI CAD tools.
7	To simulate MEMS fabrication process steps using industry-standard TCAD tools, design mask layouts, and analyze the resulting device structure.

8	To model and analyze the mechanical behavior of MEMS cantilever structures using the open source simulation tool (SUGAR)
9	To study and analyze IC and MEMS packaging techniques and evaluate the effect of packaging on device performance and reliability using FEM Multiphysics simulation tools.

(Course teachers can design their own experiments based on the prescribed syllabus)

Link for online resources:

1. **https://nanohub.org/**
2. **https://www.vlab.co.in/**
3. **www.microwind.com**

Note:

Suggested List of Experiments is indicative. However, flexibility lies with the individual course instructor designing and introducing new, innovative and challenging experiments, (limited to maximum 30% variation to the suggested list) from within the curriculum, so that the fundamentals and applications can be explored to give greater clarity to the students, and they can be motivated to think differently.

Term Work:

At least 8 experiments covering the entire syllabus must be given "Batch Wise". The experiments can be conducted with the help of a simulation tool, virtual lab and FEM tool. The teacher should refer to the suggested list of experiments and can design additional experiments to acquire practical design skills. The experiments should be student centric and attempts should be made to make experiments more meaningful, interesting and innovative.
Term work assessment must be based on the overall performance of the student with every experiment and assignments graded from time to time. The grades will be converted to marks as per "Credit and Grading System" manual and should be added and averaged. Based on the above scheme grading and term work assessment should be done.

Course Code	Course Name	Teaching Scheme (Hrs/week)			Credits Assigned			
22761142	**Signals and Image Processing**	L	T	P	L	T	P	Total
		2	--	--	2	--	1	3
		Examination Scheme						
			IA1	**IA2**	**ESE**		**Total**	
		Theory	20	20	60		100	

Pre-requisite Course Codes	**PC301: Signals & System (Mathematics III)**	
Course Outcomes	After the successful completion students should be able to	
	CO1	Apply fundamental DSP principles to represent, analyze, and process discrete-time signals and systems
	CO2	Apply frequency-domain analysis techniques for efficient spectral analysis and processing of discrete-time signals.
	CO3	Apply image enhancement techniques in spatial and frequency domains to improve image quality.
	CO4	Analyze and apply image segmentation methods to extract meaningful regions from images.
	CO5	Apply morphological operations and image representation techniques for machine vision applications.
	CO6	Evaluate and apply appropriate image compression techniques to reduce image redundancy and optimize storage and transmission efficiency

Module No.	**Unit No.**	**Topics**	**Reference**	**Hrs.**
Module 1	**1**	**Digital Signal Processing Fundamentals**		7Hrs
	1.1	Introduction to Digital Signal Processing, Sampling and Reconstruction, Standard DT Signals, Concept of Digital Frequency, Representation of DT signal using Standard DT Signals,Operation on Signals(shifting, reversal, scaling, addition, multiplication), Classification of Discrete-Time Signals and Systems	1,2	
	1.2	Linear Convolution formulation for 1-D signal (without mathematical proof), Circular Convolution , Linear convolution using Circular Convolution. Auto and Cross Correlation,Concept of LTI system, Output of DT system using Time Domain Linear Convolution		
Module 2	**2**	**Discrete Fourier Transform and Fast Fourier Transform**		7Hrs
	2.1	Introduction to DFT, Properties of DFT without mathematical proof, Convolution of long sequences, Introduction to 2-D DFT.	1,2	
	2.2	Need of FFT, Radix-2 DIT-FFT algorithm, DIT-FFT Flow graph		

		for N=4 and 8, Inverse FFT algorithm.		
Module 3	**3**	**Fundamentals of Digital Image Processing**	3,4	8 Hrs
	3.1	**Fundamentals of Image Processing :** Introduction and Applications of Image Processing, Image Formation, Sampling and Quantization ,Basic Relationships between Pixels		
	3.2	**Intensity Transformations & Spatial Domain Enhancement :** Gray Level Transformations: Digital Negative, Log Transformation, Power-law Transformation, Contrast Stretching		
	3.3	**Histogram Processing:** Histogram Equalization, Histogram Matching		
	3.4	**Spatial Filtering & Frequency Domain Basics**: Smoothing Filters, Sharpening Filters, High-Boost Filters, Introduction to Frequency Domain Filtering (LPF, HPF – concept only)		
Module 4	**4**	**Image Segmentation**		6 Hrs
	4.1	**Thresholding Techniques** :Global and Adaptive Thresholding, Basics of Otsu's Method.	3,4	
	4.2	**Edge Detection:** Roberts, Prewitt, Sobel Operators, Laplacian and Canny edge detection.		
	4.3	**Region-Based Segmentation:** Region Growing ,Splitting and Merging, Watershed algorithm.		
Module 5	**5**	**Binary Image Processing and Image Representation**		7Hrs
	5.1	**Binary Image Processing:** Morphology**,** Erosion, Dilation, Opening, Closing, Hit-or-Miss Transformation, Boundary extraction , Hole filling, Thinning and Thickening	3, 4	
	5.2	**Image Representation:** Machine vision, Chain Codes, Shape Number, Polygon Approximation, Introduction to Texture, Co-occurrence matrix		
Module 6	**6**	**Image Compression**		4Hrs
	6.1	**Fundamentals:** Need of compression, Redundancy and Types, Compression Ratio, Compression Methods.	3,4	
	6.2	**Lossless Compression**: Run-Length Encoding, Huffman Coding, LZW Coding		
	6.3	**Lossy Compression:** Lossy Predictive Coding, Transform Coding, Improved Gray Scale Quantization.		
			Total	**39**

Course Assessment:

Theory:

IA1: 20 Marks written examination for one hour
IA2: 20 Marks written examination for one hour

ESE: 60 Marks written examination for two hours

Recommended Books:

[1] John G. Proakis and D.G. Manolakis Introduction to Digital signal processing Pearson Fourth edition, 2015

[2] A. NagoorKani Digital Signal Processing McGraw Hill Publications 2nd edition

[3] Rafel C. Gonzalez and Richard E. Woods, _Digital Image Processing‘, Pearson Education Asia, 3rd Edition, 2009.

[4] Anil K. Jain, _Fundamentals and Digital Image Processing‘, Prentice Hall of India Private Ltd, 3rd Edition.

[5] S. Jayaraman, E. Esakkirajan and T. Veerkumar, ―Digital Image Processing‖ TataMcGraw Hill Education Private Ltd, 2009

Online References:

1. Image Processing Tutorials: https://scikit-image.org/
2. MATLAB Image Processing Toolbox: https://www.mathworks.com/
3. NPTEL courses on Digital signal Processing https://onlinecourses.nptel.ac.in/noc19_ee50/course

<table>
<tr><th rowspan="2">Course Code</th><th rowspan="2">Course Name</th><th colspan="3">Teaching Scheme (Hrs/week)</th><th colspan="4">Credits Assigned</th></tr>
<tr><td>L</td><td>T</td><td>P</td><td>L</td><td>T</td><td>P</td><td>Total</td></tr>
<tr><td rowspan="5">22761182</td><td rowspan="5">Signals and Image Processing Lab</td><td>--</td><td>--</td><td>2</td><td>--</td><td>--</td><td>1</td><td>1</td></tr>
<tr><th colspan="7">Examination Scheme</th></tr>
<tr><th colspan="2">Term work</th><th colspan="3">Orals</th><th colspan="2">Total</th></tr>
<tr><td colspan="2">25</td><td colspan="3">25</td><td colspan="2">50</td></tr>
</table>

<table>
<tr><td>Pre-requisite Course Codes</td><td colspan="2">PC301: Signals & System (Mathematics III)</td></tr>
<tr><td rowspan="3">Laboratory Objectives</td><td>1.</td><td>To develop a strong foundation in digital signal processing concepts including discrete-time signals, systems, sampling, reconstruction, signal operations, convolution, correlation, and LTI system analysis.</td></tr>
<tr><td>2</td><td>To understand and apply frequency-domain analysis techniques such as DFT, FFT, and inverse FFT for efficient spectral analysis and processing of discrete-time signals.</td></tr>
<tr><td>3.</td><td>To understand fundamental concepts and techniques of image segmentation and morphological operations</td></tr>
<tr><td></td><td>4.</td><td>To analyze and implement lossless compression techniques such as Run-Length Encoding, Huffman Coding, and LZW Coding.</td></tr>
</table>

Laboratory Outcomes	After the successful completion students should be able to	
	LO1	To generate basic discrete-time signals such as unit impulse, unit step, ramp, exponential, and sinusoidal sequences using programming tools.
	LO2	To implement and compute linear, circular convolution, Discrete Fourier Transform and Inverse Discrete Fourier Transform for small sequences (4-point and 8-point).
	LO3	Develop programs for image enhancement in spatial and frequency domains.
	LO4	Implement segmentation algorithms for object detection and image analysis tasks.
	LO 5	Perform binary image processing using morphological operations for noise removal and shape enhancement
	LO6	Apply image compression techniques to an image to achieve efficient image storage.

Laboratory Experiments:

Sr. No.	Title of experiment (Suggested List of Experiments)	Hardware /Software	Module	Reference
1.	To Perform a program for generation of basic signals.	**Software**	**1**	**3**
2.	To perform linear and circular convolution of discrete time sequences.	**Software**	**1**	**3**
3.	To perform 4-point/8-point DFT and IDFT of the discrete time sequence and sketch the magnitude and phase spectrum.	**Software**	**2**	**3**
4.	Write a program to read and display images using a library (e.g., OpenCV/Matplotlib). Perform basic operations: cropping, resizing, and flipping images. To enhance image contrast using gray-level transformations and histogram equalization.	**Software**	**3**	**1,2**
5.	Write a program to implement spatial filtering techniques to an image for noise reduction and edge enhancement.	**Software**	**3**	**1,2**

6	Write a program to Implement edge detection algorithms: Sobel, Prewitt, and Canny.	**Software**	**4**	**1,2**
7	Write a program to Perform morphological operations: erosion, dilation, opening, and closing.	**Software**	**5**	**1,2**
8.	Write a program to Implement Fourier Transform to enhance images in the frequency domain.	**Software**	**2**	**1,2**
9.	Write a program to compress an image using a standard technique (e.g., Run-Length Encoding or JPEG).	**Software**	**6**	**1,2**
10.	Write a program to Perform color image processing: separate and manipulate RGB channels.	**Software**	**3**	**1,2**
11.	Design and implement any application of Image Processing (e.g., license plate detection or histogram-based image matching application etc.).	**Software**	**3, 4, 5, 6**	**1,2**

Laboratory Assessment:

Assessment:

Term Work: Term Work shall consist of at least 10 to 12 practicals' based on the above list. Also, Term work Journal must include at least 2 assignments.

Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)

Practical/ Oral Exam: An Oral examination will be held based on the above syllabus.

Recommended Books:

[1] Digital Image Processing Using MATLAB, Rafael C. Gonzalez, Richard E. Woods, Steven L. Eddins, Tata Mc Graw Hill Pvt. Ltd. Third Edition, 2011.

[2] Digital Image Processing and Analysis: Application with MATLAB and CVIP tools, SE Umbaugh, Taylor & Francis/CRC Press, 3rd Edition, 2018

[3] Digital Signal Processing Using MATLAB: A Problem Solving Companion by Vinay K. Ingle and John G. Proakis

Term Work:

At least 10 experiments covering the entire syllabus of PCL 302 (Electronic Devices and circuits Lab) should be set to have well predefined inference and conclusion. This must include **50%** Hardware and **50%** Simulation experiments. The experiments should be student centric and attempts should be made to make the experiments meaningful and interesting. Experiments must be graded from time to time. The grades should be converted into marks as per the Credit and Grading System manual and should be added and averaged. The grading and term work assessment should be done based on this scheme. The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work. Practical and Oral exam will be based on the entire syllabus.

Note:

Suggested List of Experiments is indicative. However, flexibility lies with individual course instructors to design and introduce new, innovative and challenging experiments, (limited to maximum 30% variation to the suggested list) from within the curriculum, so that the fundamentals and applications can be explored to give greater clarity to the students and they can be motivated to think differently.

<table>
<tr><th rowspan="2">Course Code</th><th rowspan="2">Course Name</th><th colspan="3">Teaching Scheme (Hrs/week)</th><th colspan="4">Credits Assigned</th></tr>
<tr><td>L</td><td>T</td><td>P</td><td>L</td><td>T</td><td>P</td><td>Total</td></tr>
<tr><td rowspan="7">22761143</td><td rowspan="7">Cryptography</td><td>3</td><td>--</td><td>1</td><td>3</td><td>--</td><td>1</td><td>4</td></tr>
<tr><td colspan="7">Examination Scheme</td></tr>
<tr><td rowspan="2">Theory</td><td>IA1</td><td>IA2</td><td colspan="2">ESE</td><td colspan="2">Total</td></tr>
<tr><td>20</td><td>20</td><td colspan="2">60</td><td colspan="2">100</td></tr>
<tr><td rowspan="2">Practical</td><td colspan="2">Term Work</td><td colspan="2">Practical Exam</td><td colspan="2">Total</td></tr>
<tr><td colspan="2">25</td><td colspan="2">25</td><td colspan="2">50</td></tr>
</table>

<table>
<tr><td>Pre-requisite Course Codes</td><td colspan="2">PC401-Maths, PC502 - Computer Network</td></tr>
<tr><td></td><td colspan="2"></td></tr>
<tr><td rowspan="7">Course Outcomes</td><td colspan="2">After the successful completion students should be able to</td></tr>
<tr><td>CO1</td><td>Analyze security concepts and cryptographic fundamentals.</td></tr>
<tr><td>CO2</td><td>Apply symmetric encryption techniques for secure communication.</td></tr>
<tr><td>CO3</td><td>Analyze public key cryptographic algorithms and key exchange methods.</td></tr>
<tr><td>CO4</td><td>Implement hashing, authentication, and digital signature techniques.</td></tr>
<tr><td>CO5</td><td>Analyze secure communication protocols and PKI mechanisms.</td></tr>
<tr><td>CO6</td><td>Evaluate emerging cryptographic technologies for cybersecurity applications.</td></tr>
</table>

Module No.	Unit No.	Topics	Reference	Hrs.
Module 1	**1**	**Introduction to Cryptography and Security Fundamentals**		7 Hrs
	1.1	Information Security Fundamentals, CIA Triad, Security Threats, Vulnerabilities and Attacks, Security Services and Security Mechanisms, Cryptography Terminology: Plaintext, Ciphertext, Encryption, Decryption, Keys	1, 2, 3	
	1.2	Classical Encryption Techniques: Caesar Cipher, Playfair Cipher, Hill Cipher, Vigenère Cipher, Transposition Cipher, Cryptanalysis Techniques		
	1.3	Steganography and Introduction to Modern Cryptography		
Module 2	**2**	**Symmetric Key Cryptography and Block Ciphers**		8 Hrs
	2.1	Symmetric Key Cryptography Principles, Stream and Block Ciphers, Feistel Structure, Confusion and Diffusion	1, 2, 3, 4	
	2.2	Data Encryption Standard (DES), Triple DES (3DES), Differential and Linear Cryptanalysis		
	2.3	Advanced Encryption Standard (AES): Structure, Key Expansion, Encryption and Decryption Process		
	2.4	Modes of Operation: ECB, CBC, CFB, OFB, CTR Modes, Introduction to Lightweight Cryptography		
Module 3	**3**	**Number Theory and Public Key Cryptography**	1, 2, 3, 4	8 Hrs
	3.1	Prime Numbers, Euclidean and Extended Euclidean Algorithms, Modular Arithmetic, Fermat's and Euler's Theorem, Chinese Remainder Theorem		
	3.2	Public Key Cryptography Concepts, RSA Algorithm, Diffie-Hellman Key Exchange, ElGamal Cryptosystem		
	3.3	Elliptic Curve Cryptography (ECC), Introduction to Post-Quantum Cryptography, Man-in-the-Middle Attack		
Module 4	**4**	**Hash Functions, Authentication and Digital Signatures**		7 Hrs
	4.1	Cryptographic Hash Functions, Properties of Hash Functions, MD5, SHA-1, SHA-2, SHA-3	1, 2, 3, 4	
	4.2	Message Authentication Codes (MAC), HMAC, CMAC		
	4.3	Digital Signatures, Digital Signature Standard (DSS), RSA-based Digital Signature, Authentication Protocols		
	4.4	Multi-Factor Authentication (MFA), Password Security and Salting Techniques		
Module 5	**5**	**Network Security and Public Key Infrastructure**		7 Hrs
	5.1	Public Key Infrastructure (PKI), X.509 Certificates, Certificate Authorities, Key Management and Distribution	1, 2, 4, 5	
	5.2	Kerberos Authentication Protocol, SSL and TLS Protocols		
	5.3	IP Security (IPSec), Virtual Private Networks (VPNs), Secure Email Systems: PGP and S/MIME		
	5.4	Firewalls and Intrusion Detection Systems (IDS)		
Module 6	**6**	**Emerging Trends and Applications in Cryptography**		8 Hrs
	6.1	Blockchain and Cryptocurrency Security, Cryptography in Cloud Computing	2, 3, 5, 6	
	6.2	Secure APIs, Zero Trust Security Model, Cryptography in IoT and Edge Devices		

	6.3	Quantum Computing Threats and Introduction to Post-Quantum Cryptography		
	6.4	AI in Cybersecurity, Secure Coding Practices, Case Studies on Security Breaches, Ethical and Legal Aspects of Cryptography		
			Total	**39**

Course Assessment:

Theory:

IA1: 20 Marks written examination for one hour
IA2: 20 Marks written examination for one hour

ESE: 60 Marks written examination for two hours

Recommended Books:

[1] William Stallings, “Cryptography and Network Security: Principles and Practice”, Pearson Education.
[2] Behrouz A. Forouzan, “Cryptography and Network Security”, McGraw Hill Education.
[3] Jonathan Katz and Yehuda Lindell, “Introduction to Modern Cryptography”, CRC Press.
[4] Bruce Schneier, “Applied Cryptography”, Wiley India.
[5] Wenbo Mao, “Modern Cryptography: Theory and Practice”, Pearson Education.
[6] Mark Stamp, “Information Security: Principles and Practice”, Wiley India.

Online References:

1. NPTEL – Cryptography and Network Security by IIT Kharagpur
2. Coursera – Applied Cryptography Specialization
3. MIT OpenCourseWare – Cryptography and Computer Security
4. Stanford Online – Computer and Network Security
5. Cybrary – Cryptography Fundamentals

Course Code	Course Name	Teaching Scheme (Hrs/week)			Credits Assigned			
22761183	Cryptography Lab	L	T	P	L	T	P	Total
		--	--	2	--	--	1	1
		Examination Scheme						
		Term work		Orals			Total	
		25		25			50	

Pre-requisite Course Codes	**PCL403 Discrete Structures and Automata Theory Lab, VSE401 Creative Coding in Python, PCL502 Computer Network Lab**	
Laboratory Objectives	1.	To provide hands-on experience in implementing cryptographic algorithms and security techniques.
	2	To understand encryption, decryption, hashing, and authentication mechanisms through practical experiments
	3.	To analyze secure communication protocols and cryptographic security models
	4.	To introduce modern cryptographic tools and software used in cybersecurity applications.
	5.	To develop practical skills for designing secure systems and applications using cryptographic techniques.
Laboratory Outcomes	**After the successful completion students should be able to**	
	LO1	Implement classical and modern cryptographic algorithms for secure communication.
	LO2	Analyze symmetric and asymmetric encryption techniques using software tools.
	LO3	Implement hashing, authentication, and digital signature mechanisms.

	LO4	Analyze secure communication protocols such as SSL/TLS and IPSec.
	LO 5	Develop secure applications using cryptographic and cybersecurity tools.

Laboratory Experiments:

Sr. No.	Title of experiment	Hardware / Software	Module	Reference
1.	Implementation of Caesar Cipher and Monoalphabetic Cipher	Software	1	1, 2
2.	Implementation of Playfair and Vigenère vCipher Techniques	Software	1	1, 2
3.	Implementation of Hill Cipher and Transposition Cipher	Software	1	1, 3
4.	Study and Performance Analysis of DES Algorithm	Software	2	1, 2
5.	Implementation of Triple DES (3DES) Encryption and Decryption	Software	2	1, 2
6	AES Encryption and Decryption for Secure File Transfer	Software	2	1, 2, 3
7	Comparative Analysis of ECB, CBC, CFB, OFB, and CTR Modes of Operation	Software	2	1, 3
8.	Implementation of RSA Algorithm for Secure Communication	Software	3	1, 2, 3
9.	Implementation of Diffie-Hellman Key Exchange Algorithm	Software	3	1, 2
10.	Study and Implementation of Elliptic Curve Cryptography (ECC)	Software	3	2, 3
11.	Generation and Verification of Digital Signature using RSA/DSA	Software	4	1, 2, 3
12.	Implementation of SHA-256 and SHA-3 Hashing Algorithms	Software	4	1, 2
13	Development of Password Authentication System using Hashing and Salting	Software	4	1, 3
14	SSL/TLS Packet Analysis using Wireshark	Software	5	1, 2
15	Study of Firewall Configuration and Intrusion Detection System (IDS)	Hardware/ Software	5	2, 3
16	Mini Project: Blockchain-based Secure Transaction Demonstration	Software	6	3, 4
17	Study of Cryptography in Cloud Security using Secure API Communication	Software	6	2, 4

Laboratory Assessment:

Assessment:

Term Work: Term Work shall consist of at least 10 to 12 practicals based on the above list. Also, Term work Journal must include at least 2 assignments or one mini project/case study related to cybersecurity or cryptography applications.

Term Work Marks:

25 Marks (Total marks) =
15 Marks (Experiment)
5 Marks (Assignments/Mini Project)
5 Marks (Attendance)

Practical/ Oral Exam:

An Oral examination will be conducted based on the entire syllabus and laboratory experiments.

Recommended Software / Tools

1. Python with Cryptography Libraries
2. OpenSSL Toolkit
3. Wireshark Network Analyzer
4. Kali Linux Tools
5. CrypTool
6. GPG (GNU Privacy Guard)

Recommended Books:

[1] David A Bell, "Laboratory Manual for Electronic Devices and Circuits", 4th edition, PHI, [1] William Stallings, "Cryptography and Network Security: Principles and Practice", Pearson Education.
[2] Behrouz A. Forouzan, "Cryptography and Network Security", McGraw Hill Education.
[3] Bruce Schneier, "Applied Cryptography", Wiley India.
[4] Jonathan Katz and Yehuda Lindell, "Introduction to Modern Cryptography", CRC Press.

Term Work:

At least 10 experiments covering the entire syllabus of PEL 602 (Cryptography Lab) should be set to have well predefined inference and conclusion. This must include **50%** Hardware and **50%** Simulation experiments. The experiments should be student centric and attempts should be made to make the experiments meaningful and interesting. Experiments must be graded from time to time. The grades should be converted into marks as per the Credit and Grading System manual and should be added and averaged. The grading and term work assessment should be done based on this scheme. The final certification and acceptance of term work ensures satisfactory performance of laboratory work and minimum passing marks in term work. Practical and Oral exam will be based on the entire syllabus.

Note:

Suggested List of Experiments is indicative. However, flexibility lies with individual course instructors to design and introduce new, innovative and challenging experiments, (limited to maximum 30% variation to the suggested list) from within the curriculum, so that the fundamentals and applications can be explored to give greater clarity to the students and they can be motivated to think differently.

<table>
<tr><th rowspan="2">Course Code</th><th rowspan="2">Course Name</th><th colspan="3">Teaching Scheme (Hrs/week)</th><th colspan="4">Credits Assigned</th></tr>
<tr><td>L</td><td>T</td><td>P</td><td>L</td><td>T</td><td>P</td><td>Total</td></tr>
<tr><td rowspan="5">22761144</td><td rowspan="5">Computer Communication Network</td><td>3</td><td>--</td><td>1</td><td>3</td><td>--</td><td>1</td><td>4</td></tr>
<tr><td colspan="7">Examination Scheme</td></tr>
<tr><td></td><td>IA1</td><td>IA2</td><td colspan="2">ESE</td><td colspan="2">Total</td></tr>
<tr><td>Theory</td><td>20</td><td>20</td><td colspan="2">60</td><td colspan="2">100</td></tr>
</table>

<table>
<tr><td>Pre-requisite Course Codes</td><td colspan="2">PC 502-Computer Network</td></tr>
<tr><td></td><td colspan="2"></td></tr>
<tr><td rowspan="7">Course Outcomes</td><td colspan="2">After successful completion students should be able to</td></tr>
<tr><td>CO1</td><td>Apply Internet architecture concepts for network performance analysis.</td></tr>
<tr><td>CO2</td><td>Analyze application and transport layer protocols for reliable communication.</td></tr>
<tr><td>CO3</td><td>Apply network layer concepts to analyze routing, forwarding, IP addressing, subnetting, and Software Defined Networking mechanisms.</td></tr>
<tr><td>CO4</td><td>Apply LAN and data center networking concepts to demonstrate Ethernet operations, VLAN configuration, MAC addressing, and switching mechanisms.</td></tr>
<tr><td>CO5</td><td>Analyze multimedia networking techniques for streaming, CDN, QoS, and real-time communication systems.</td></tr>
<tr><td>CO6</td><td>Analyze wireless networks and security mechanisms for secure communication and mobility management.</td></tr>
</table>

Module No.	Unit No.	Topics	Reference	Hrs.
Module 1	**1**	**Internet Architecture and Network Performance Analysis**		7Hrs
	1.1	Internet structure: Network edge, access networks, core Packet switching vs circuit switching (comparative analysis), Delay analysis: Processing, queuing, transmission, propagation, Throughput and bottleneck link analysis,	1,2,3	
	1.2	Protocol layering and service models, Encapsulation, segmentation, and reassembly, End-to-End principle and protocol design issues, Introduction to performance evaluation metrics		
Module 2	**2**	**Application & Transport Layer Services**		7Hrs
	2.1	Application architectures: Client-server, Peer-to-Peer (P2P), hybrid-Web protocols: HTTP (persistent vs non-persistent, HTTP/2 overview) DNS: Hierarchy, caching, records	1,3,6	
	2.2	Email protocols: SMTP, POP3, IMAP, Socket programming concepts, Transport layer services: Multiplexing and demultiplexing, Reliable data transfer: Stop-and-wait, pipelined protocols		
Module 3	**3**	**Network Layer – Routing, Forwarding & SDN**	1,2, 4	7Hrs
	3.1	Router architecture: Input/output ports, switching fabric, forwarding vs routing, IP addressing: CIDR, subnetting, IPv6 basics		
	3.2	Routing algorithms: Distance vector and link state Routing protocols: RIP, OSPF, BGP, Network Address Translation (NAT)		
	3.3	Introduction to Software Defined Networking, Data plane vs control plane separation		
Module 4	**4**	**LANs & Data Center Networking**		4Hrs
	4.1	Ethernet: Frame structure and switching, VLAN basics, MAC addressing and ARP, Switching in LANs, Data Center Networking: Architecture and requirements, Traffic patterns and load balancing	2,3,5	

Module 5	**5**	**Multimedia Networking & Content Distribution**	1,4,5	6Hrs
	5.1	Multimedia networking applications, Streaming stored and live media ,DASH (Dynamic Adaptive Streaming over HTTP)		
	5.2	, Content Distribution Networks (CDN): Architecture and operation, Quality of Service (QoS) for multimedia: Delay, jitter, bandwidth requirements, Real-time transport: RTP/RTCP concepts, Internet telephony basics (VoIP)		
Module 6	**6**	**Wireless Networks & Network Security**	2,3,4	8Hrs
	6.1	Wireless networks: Wi-Fi (IEEE 802.11 architecture), Cellular networks: 3G/4G architecture basics, Mobility management: Handoff mechanisms, Wireless link characteristics and challenges		
	6.2	Network security fundamentals: Cryptography basics (symmetric/asymmetric) , Secure communication: SSL/TLS concepts, Authentication and integrity, Firewalls and intrusion detection systems, Security challenges in wireless networks		
			Total	**39**

Course Assessment:

Theory:

IA1: 20 Marks written examination for one hour
IA2: 20 Marks written examination for one hour

ESE: 60 Marks written examination for two hours

Recommended Books:

[1] James F. Kurose, Keith W. Ross, “Computer Networking: A Top-Down Approach”, Pearson Education, Latest Edition.
[2] Andrew S. Tanenbaum, David J. Wetherall, “Computer Networks”, Pearson Education, 5th Edition.
[3] Behrouz A. Forouzan, “Data Communications and Networking”, McGraw Hill Education, 5th Edition.
[4] William Stallings, “Data and Computer Communications”, Pearson Education, 10th Edition.
[5] Larry L. Peterson, Bruce S. Davie, “Computer Networks: A Systems Approach”, Morgan Kaufmann Publishers, Latest Edition.
[6] W. Richard Stevens, “TCP/IP Illustrated, Volume 1: The Protocols”, Addison-Wesley, Latest Edition.

Online References:

NPTEL courses on Computer Communication Network: Computer Networks and Internet Protocol by Prof. Soumya Kanti Ghosh Prof. Sandip Chakraborty

<table>
<tr><th rowspan="2">Course Code</th><th rowspan="2">Course Name</th><th colspan="3">Teaching Scheme (Hrs/week)</th><th colspan="4">Credits Assigned</th></tr>
<tr><td>L</td><td>T</td><td>P</td><td>L</td><td>T</td><td>P</td><td>Total</td></tr>
<tr><td rowspan="5">22761184</td><td rowspan="5">Computer Communication Network Lab</td><td>--</td><td>--</td><td>2</td><td>--</td><td>--</td><td>1</td><td>1</td></tr>
<tr><td colspan="7">Examination Scheme</td></tr>
<tr><td colspan="2">Term work</td><td colspan="3">Orals</td><td colspan="2">Total</td></tr>
<tr><td colspan="2">25</td><td colspan="3">25</td><td colspan="2">50</td></tr>
</table>

Pre-requisite Course Codes	**PCL 502: Computer Network lab**	
Laboratory Objectives	**1**	To understand and analyze computer networking concepts through simulation and packet analysis tools.
	2	To implement application, transport, routing, and switching protocols using modern networking tools and programming techniques.
	3	To study network performance, secure communication, wireless networking, and Software Defined Networking concepts through practical experimentation.
	4	To develop hands-on skills in configuring, troubleshooting, and evaluating network architectures and communication mechanisms.
Laboratory Outcomes	After the successful completion students should be able to	
	LO1	Analyze network communication and performance metrics using packet analysis and simulation tools such as Wireshark and Packet Tracer.
	LO2	Implement client-server communication, socket programming, DNS, NAT, VLAN, and routing protocols in simulated network environments.

	LO3	Demonstrate Software Defined Networking, multimedia streaming, and wireless communication concepts using SDN and network simulation tools.
	LO4	Evaluate secure communication mechanisms, mobility management, and cellular/wireless network behavior through practical experimentation and traffic analysis.
	LO5	Interpret network performance and security characteristics by comparing protocol behaviours, analysing traffic patterns, and examining secure communication processes.

Laboratory Experiments:

Sr.NO	**Title of experiment**	**Hardware /Software**	**Module**	**Reference**
1.	To analyze the performance evaluation metrics in packet fragmentation using Wireshark	software	1	1, 3
2.	To perform Socket Programming using TCP and UDP in python	software	2	1, 4
3.	To implement Client -server architecture using HTTP/FTP using cisco packet tracer	Software	2	1, 2
4.	To create DNS resolution using cisco packet tracer	Software		1, 2
5.	To Configure OSPF and compare with RIP performance using Packet tracer/GNS3	Software	3	2, 3
6.	To Implement static and dynamic NAT and verify translation using Packet tracer	Software	3	2, 3
7.	To Create VLANs and demonstrate inter-VLAN communication using Packet Tracer	Software	4	2, 3

8.	To simulate client-server communication in SDN environment	Software	4	1, 2
9.	Study adaptive bitrate streaming using DASH.js + browser / YouTube stats	Software	4	1, 3
10.	To understand and analyze hand (handover) mechanisms in wirel networks using ns-3 or Mininet-WiFi.	Software	5	2, 3
11.	To analyze 4G cellular network traffic using Wireshark through a mobile hotspot connection.	Software	6	1, 3
12.	To understand secure communication mechanisms by analyzing encrypted network traffic using Wireshark (SSL/TLS Handshake Analysis)	Software	6	1, 4

Laboratory Assessment:

Assessment:

Term Work: Term Work shall consist of at least 10 to 12 practicals' based on the above list. Also, Term work Journal must include at least 2 assignments.

Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)

Practical/ Oral Exam: An Oral examination will be held based on the above syllabus.

Recommended Books:

[1] J. F. Kurose and K. W. Ross, Computer Networking: A Top-Down Approach, 7th ed. Boston, MA, USA: Pearson, 2017.

[2] A. S. Tanenbaum and D. J. Wetherall, Computer Networks, 5th ed. Upper Saddle River, NJ, USA: Pearson, 2011.

[3] W. Stallings, Data and Computer Communications, 10th ed. Boston, MA, USA: Pearson, 2013.

[4] W. R. Stevens, TCP/IP Illustrated, Volume 1: The Protocols. Reading, MA, USA: Addison-Wesley, 1994.

Term Work:

At least 10 to 12 experiments covering the entire syllabus of Computer Communication Network Lab should be conducted with well-defined objectives, observations, inference, and conclusions. The practical should include a balanced combination of configuration-based, programming-based, packet analysis, and simulation-based experiments using tools such as Cisco Packet Tracer, Wireshark, GNS3, Mininet, ns-3, and Python programming.

The experiments should be student-centric, focusing on hands-on implementation of networking concepts such as routing, switching, socket programming, network performance analysis, SDN, multimedia networking, wireless communication, and network security. Attempts should be made to design experiments that enhance analytical thinking, troubleshooting skills, and understanding of real-world communication systems.

Experiments must be graded periodically based on performance, understanding, implementation, observations, and viva. The grades should be converted into marks as per the Credit and Grading System manual and should be added and averaged. The grading and term work assessment should be carried out according to this scheme.

The final certification and acceptance of term work ensure satisfactory performance of laboratory work and minimum passing marks in term work. Practical and Oral examination will be based on the entire syllabus.

Note:

The suggested list of experiments is indicative. However, flexibility lies with individual course instructors to design and introduce new, innovative, and challenging experiments (limited to maximum 30% variation from the suggested list) within the scope of the curriculum. This will help students explore networking fundamentals and applications with greater clarity and encourage them to develop problem-solving and practical implementation skills in modern computer communication networks.

Vertical – 2

(MDM Courses offered by BoS -Electronics Engineering to other branches)

Course Code	Course Name	Teaching Scheme (Hours/week)			Credits Assigned			
		L	**P**	**T**	L	**P**	**T**	**Total**
2275211	Embedded Systems and RTOS	03	--	--	03	01	--	04
		Examination Scheme						
			IA1	IA2	ESE		Total	
		Theory	20	20	60		100	

Pre-requisite Course Codes	1. Digital Electronics 2. Basics of Microcontrollers	
After the successful completion of the course, students should be able to:		
Course Outcomes	CO1	Identify and describe various characteristic features and applications of embedded systems.
	CO2	To acquaint with fundamentals of embedded C programming.
	CO3	Analyse various processors in embedded systems and study ARM Cortex processor.
	CO4	To study various analog and digital interfaces used in embedded systems
	CO5	Compare GPOS and RTOS and investigate the concepts of RTOS.
	CO6	Based on life-cycle for embedded system design and test a system for different requirements, keeping oneself aware of ethics and environmental issues.

Module No.	Unit No.	Contents	Hrs.
1		**Introduction to Embedded Systems and Fundamentals of Embedded Programming**	**06**
	1.1	Definition, Characteristics, Embedded systems vs General Computing Systems, Classification, applications	
	1.2	Design metrics of Embedded system and challenges in optimization of metrics.	
	1.3	Embedded C: Need, data Types (bit, byte) ,Modifiers (static and volatile), bit operations, unsigned numbers representation, ISR (exception handling)	
2		**Embedded Processors**	**10**
	2.1	Features of Embedded cores- µC, ASIC, ASSP, SoC, FPGA, RISC and CISC cores. Types of memories.	
	2.2	Case Study: ARM Cortex-M3 Features, Architecture, Programmer's model, Special Registers, Operating Modes and States, MPU, Memory map and NVIC.	
	2.3	Low power: - Need and techniques. Case study of Low Power modes in Cortex-M3.	
	2.4	Compare A,R and M series	
	2.5	Hardware Features of ATMEGA 328P (Arduino Uno/Nano)	
3		**Interfacing Analog and Digital Devices**	**05**
	3.1	Interfacing various input (analog and digital) devices.	
	3.2	Interfacing various output (analog and digital) devices.	
	3.3	Signal Conditioning: Amplifiers, filters (LPF and notch filter)	
4		**Embedded Communication Interfaces**	**04**
		Comparative study of serial communication Interfaces :	
	4.1	4.1Wired: UART, SPI, I2C, CAN bus	
	4.2	4.2 Wireless: USB (v2.0), Bluetooth, Zig-Bee, RFID	
5		**Real Time Operating Systems**	**06**
	5.1	5.l Real-time Operating system:- Need of RTOS in Embedded system software and comparison with GPOS, Task, Task states, Multi-tasking, Task scheduling, and algorithms-Preemptive SJF, Round-Robin, Priority, Rate Monotonic Scheduling (graphical representation) , Earliest Deadline First. (Numerical on all scheduling techniques to estimate Turnaround time and schedulability).	
	5.2	Inter-process communication: Message queues, Mailbox, Event timers.	
	5.3	Task synchronization: Need, (issues:- Deadlock, Race condition, live Lock) solutions using Mutex, Semaphores. Shared data problem, Priority inversion and Work-around.	
6		**System Integration and Design**	08
	6.1	Embedded Product Design Life-Cycle (EDLC)	
	6.2	i) Specification requirements(choice of components), ii) Hardware architecture (Block Diagram) iii) Software architecture (FSM/flowchart) **iv)** Testing and Debugging Methodology **:**: Hardware testing tools, Emulator, Software Testing tools: simulator, debugger. White-Box and Black-Box testing.	
	6.3	Example applications: automatic vending machine, smart cards, agriculture, environment etc.	
		Total	**39**

Text Books:

1.Dr. K.V. K. K. Prasad, "Embedded Real Time System: Concepts, Design and Programming", Dreamtech, New Delhi, Edition 2014.

2.Rajkamal, “Embedded Systems: Architecture, Programming and Design”, McGraw Hill Education (India) Private Limited, New Delhi, 2015, Edition 3rd.
3.Sriram Iyer, Pankaj Gupta,“ Embedded Real Time Systems Programming”, Tata McGraw Hill Publishing Company ltd., 2003.
4. Joseph Yiu,“The Definitive guide to ARM CORTEX-M3 & CORTEX-M4 Processors”, Elsevier, 2014, 3rd Edition.
5.www.freertos.org

Reference Books:

1.David Simon,“An Embedded Software Primer”, Pearson, 2009.
2.Jonathan W. Valvano, “Embedded Microcomputer Systems – Real Time Interfacing”, Publisher - Cengage Learning, 2012 Edition 3rd.
3.AndrewSloss, Domnic Symes, Chris Wright,“ ARM System Developers Guide Designing and Optimising System Software”, Elsevier, 2004
4.FrankVahid, Tony Givargis,“Embedded System Design – A Unified Hardware/Software Introduction”, John Wiley & Sons Inc., 2002.
5.Shibu K V, “Introduction to Embedded Systems”, Tata McGraw Hill Education Private Limited, New Delhi, 2009.

Internal Assessment (20-Marks):

Internal Assessment (IA) consists of two class tests of 20 marks each. IA-1 is to be conducted on completion of approximately 40% of the syllabus, and IA-2 will be based on the remaining contents (approximately 40% syllabus, but excluding contents covered in IA-1). The duration of each test shall be one hour. The average of the two tests will be considered as the IA marks.

End Semester Examination (60-Marks):

Weightage to each of the modules in end-semester examination will be proportional to number of respective lecture hours mentioned in the curriculum.

1. Question paper will comprise of **total 06 questions**, each **carrying 20 marks**.
2. **Question No: 01** will be **compulsory** and based on entire syllabus wherein 4 to 5 sub- questions will be asked.
3. Remaining questions will be mixed in nature and randomly selected from all the modules.
4. Weightage of each module will be proportional to number of respective lecture hours as mentioned in the syllabus.
5.**Total 04 questions** need to be solved.

<table>
<tr><th rowspan="2">Course Code</th><th rowspan="2">Course Name</th><th colspan="3">Teaching Scheme (Hours/week)</th><th colspan="4">Credits Assigned</th></tr>
<tr><th>L</th><th>P</th><th>T</th><th>L</th><th>P</th><th>T</th><th>Total</th></tr>
<tr><td rowspan="4">2275212</td><td rowspan="4">Embedded Systems and RTOS Lab</td><td>--</td><td>02</td><td>--</td><td>--</td><td>01</td><td>--</td><td>01</td></tr>
<tr><td colspan="7">Examination Scheme</td></tr>
<tr><td colspan="2">Term Work</td><td colspan="2">PR / OR Exam</td><td colspan="3">Total</td></tr>
<tr><td colspan="2">25</td><td colspan="2">25</td><td colspan="3">50</td></tr>
</table>

Prerequisite: 1. Basics of Microcontroller programming
2. C programming

Lab Objectives: To design and write efficient code for single-tasking and multi-tasking embedded systems

Lab Outcomes:

After successful completion of the course students will be able to:

- Interface various sensors and actuators to embedded cores.
- Write code using RTOS for multi-tasking Embedded systems
- Design applications using different embedded cores

Important Note:

1. Students must perform the experiments using Hardware.
2. Experiments must include a minimum of 3 experiments using Free RTOS

	Suggested List of Practicals /Tutorials	**Ref.**
1	Interfacing button (switch) controlled LEDs with any embedded core.	1,2,7
2	Interfacing of Temperature sensor with any embedded core.	1,2,7
3	Interfacing of LCD/ Seven segment display with any embedded core.	1,2,7
4	Interfacing of Ultrasonic/Humidity sensor with any embedded core.	1,2,4
5	Interfacing of a relay with any embedded core.	1,2,4
6	Interfacing of a DC motor (speed and Direction control) with any embedded core.	2,10
7	Interfacing of a stepper motor (to move by a particular angle) with any embedded core.	2,10,4
8	Implement power management in any embedded core of your choice	2,10
9	Implement the I2C communication to connect to any hardware	2,10
1	Porting of FreeRTOS to Arduino/STM32.	4
1	Write a Program to Create Multiple Tasks and understand the Multitasking capabilities of RTOS(FreeRTOS).	2
1	Write a Program to illustrate the Queue Management Features of FreeRTOS.	3
1	Write a Program to illustrate the Event Management Features of FreeRTOS.	4
1	Write a Program to illustrate the use of Binary and Counting Semaphore for Task Synchronisation using FreeRTOS.	5
1	Build a Multitasking Real-Time Applications using the above IPC Mechanisms (Message Queue, EventGroup, Semaphores) with FreeRTOS on Arduino/STM32.	6

Course Code	Course Name	Teaching Scheme (Hrs/week)			Credits Assigned			
		L	T	P	L	T	P	Total
2276211	Internet of Things (IoT)	3	--	2	2	--	2	4
		Examination Scheme						
			IA1	IA2	ESE		Total	
		Theory	20	20	60		100	

Pre-requisite Course Codes	(PC302) Electronic Devices, (MD401) Microprocessor and Microcontrollers, (MD501) Embedded System and RTOS (PC502) Computer Networks	
Course Outcomes	After successful completion students should be able to	
	CO1	Apply IoT system architecture concepts for real-world applications
	CO2	Integrate sensors and actuators with embedded IoT systems
	CO3	Analyse IoT communication models and wireless technologies.
	CO4	Apply concepts of Embedded and RTOS-based applications for IoT devices
	CO5	Analyze IoT data using cloud platforms and AI techniques
	CO6	Evaluate challenges in IoT Security and applications through case studies in IIoT, Healthcare IoT and Smart Grid systems

Module No.	Unit No.	Topics	Refe rence	Hrs.
Module 1	**1**	**IoT Architecture & Design**		6Hrs
	1.1	Introduction to IoT: Definition, Characteristics, Evolution, IoT vs M2M, IoT Architecture: 3-layer, 5-layer, Cloud-Fog-Edge architecture, End-to-End IoT System Architecture	1, 2, 3, 4, 5, 7	
	1.2	IoT Design Methodology: Requirement analysis, system design, prototyping, IoT Applications: Smart homes, smart cities, healthcare, agriculture		
Module 2	**2**	**Sensors, Actuators & Interfacing**		6Hrs
	2.1	Sensors: Types (Temperature, Humidity, Motion, Gas, Biomedical), Actuators: Relays, Motors, Solenoids, Sensor parameters, Interfacing challenges.	1, 2, 6, 7	
	2.2	Signal Conditioning basics (link to Analog Electronics) Interfacing Techniques: GPIO, ADC, DAC, PWM Communication Interfaces: I2C, SPI, UART		
Module 3	**3**	**Embedded Systems & RTOS for IoT**	2, 3, 4, 6	8 Hrs
	3.1	IoT Embedded Platforms: Arduino, ESP32, Raspberry Pi Embedded System Architecture for IoT nodes		
	3.2	Real-Time Systems: Concepts, Introduction to multitasking in IoT systems, scheduling, interrupts, RTOS basics: Tasks, queues, semaphores, Case Study: Free RTOS-based IoT system.		
Module 4	**4**	**IoT Communication & Networking**	1, 2, 3, 7	7 Hrs
	4.1	IoT Communication Models (D2D, D2C, D2G), Protocols: MQTT, CoAP, HTTP/REST, Protocol comparison		
	4.2	Wireless Technologies: WiFi, BLE, Zigbee, LoRa (overview)		
	4.3	IPv6 and addressing in IoT, Data transmission and reliability		
Module 5	**5**	**Cloud, Data Analytics & AI in IoT**		6Hrs

	5.1	IoT Cloud Platforms: AWS IoT / Azure IoT / Thing Speak Data Pipeline Flow	2, 5, 6, 7, 10	
	5.2	Data Storage: Structured vs unstructured, NoSQL basics		
	5.3	Data Visualization: Dashboards, Introduction to AI in IoT: Predictive analytics, anomaly detection, Edge vs Cloud processing		
Module 6	**6**	**IoT Security & Applications**		6Hrs
	6.1	Security Challenges in IoT, Ethics, Prototype to Reality.	3, 8, 9, 10	
	6.2	Authentication, Encryption basics, Privacy and data protection, Multi-layer IoT Security Architecture		
	6.3	Industrial IoT (IIoT) and Smart Manufacturing		
	6.4	Case Study-based IoT System Analysis: Healthcare IoT, Smart Grid		
Total				**39**

Course Assessment:

Theory:

IA1: 20 Marks written examination for one hour
IA2: 20 Marks written examination for one hour

ESE: 60 Marks written examination for two hours

Recommended Books:

1. Arshdeep Bahga, Vijay Madisetti, “Internet of Things: A Hands-on Approach”,Universities Press (India) Pvt. Ltd.,1st Edition, 2015
2. Raj Kamal, “Internet of Things”, McGraw Hill Education, 1st Edition, 2017.
3. Raj Kamal, “Internet of Things: Architecture and Design Principles”, McGraw Hill Education, 2nd Edition, 2022.
4. Adrian McEwen, Hakim Cassimally, “Designing the Internet of Things”, John Wiley & Sons, 1st Edition, 2013.
5. Samuel Greengard, “The Internet of Things”, The MIT Press, Revised and Updated Edition, 2021.
6. Raj Kamal, “Embedded Systems: SoC, IoT, AI and Real-Time Systems”, McGraw Hill Education, 4th Edition, 2020.
7. S. Misra, A. Mukherjee, and A. Roy, 2020. Introduction to IoT. Cambridge University Press.
8. S. Misra, C. Roy, and A. Mukherjee, 2020. *Introduction to Industrial Internet of Things and Industry 4.0.* CRC Press.
9. Ovidiu Vermesan, Peter Friess, “Internet of Things Applications - From Research and Innovation to Market Deployment”1st Edition ,First Published 2014,River Publishers DOIhttps://doi.org/10.1201/9781003338628,Pages364,eBook ISBN9781003338628
10. Ovidiu Vermesan, Joël Bacquet, “Internet of Things – The Call of the Edge-Everything Intelligent Everywhere” ISBN 9788770221962,392 Pages,Published August 31, 2020 by River Publishers.

Online References
NPTEL courses :

1. Prof. Sudip Misra, (IIT Kharagpur): Introduction To Internet Of Things By https://onlinecourses.nptel.ac.in/e-learning/preview/noc26_cs37
2. Prof. Sudip Misra, (IIT Kharagpur): Introduction To Industry 4.0 And Industrial Internet Of Things https://onlinecourses.nptel.ac.in/e-learning/preview/noc26_cs38

Course Code	Course Name	Teaching Scheme (Hrs/week)			Credits Assigned			
		L	T	P	L	T	P	Total
2276212	Internet of Things (IoT)	--	--	2	--	--	1	1
		Examination Scheme						
		Term work	Orals			Total		
		25	25			50		

Pre-requisite Course Codes	**(PC302) Electronic Devices, (MD401) Microprocessor and Microcontrollers, (MD501) Embedded System and RTOS**	
Laboratory Objectives	1.	To provide hands-on experience in interfacing sensors and actuators with embedded IoT platforms for real-time data acquisition and control.
	2	To enable students to design and implement IoT communication using standard protocols and wireless technologies
	3.	To develop skills in integrating IoT devices with cloud platforms for data storage, visualization, and basic analytics.
	4.	To design and implement end-to-end IoT applications by integrating hardware, embedded systems, networking, and software components
Laboratory Outcomes	After successful completion students should be able to	
	LO1	Interface sensors and acquire real-time data
	LO2	Implement communication using IoT protocols
	LO3	Develop embedded IoT applications
	LO4	Connect IoT devices to cloud platforms
	LO 5	Design and demonstrate mini IoT project

Laboratory Experiments:

Sr. No.	Title of Experiment	Hardware / Software	Module	Reference
1	Interfacing of Temperature and Humidity Sensor (DHT11/DHT22) with Arduino/ESP32 and display output	Hardware	2	1,2,3
2	Interfacing of Motion/IR Sensor and controlling an LED/Relay (Actuator Control)	Hardware	2	1,2,3
3	Study and implementation of communication interfaces (I2C/SPI) using sensors (OLED/RTC)	Hardware	2	3,5
4	Basic WiFi connectivity using ESP32 and sending sensor data to serial monitor/web server	Hardware/Software	4	3,5
5	Implementation of MQTT protocol using public broker (publish/subscribe model)	Software	4	3,4,5
6	HTTP/REST-based data transmission from IoT	Software	4	3,4,5

	device to cloud/server			
7	Cloud integration using ThingSpeak/Firebase for data logging and visualization	Software	5	3,4,5
8	RTOS-based multi-tasking IoT system using FreeRTOS (sensor + communication tasks)	Software	3	3,4,5
9	IoT Security implementation: Basic encryption/authentication for secure communication	Software	6	3,4,5
10	Mini Project: Design and implementation of IoT application (Smart Energy / Irrigation / Health Monitoring)	Hardware/Software	1–6	1,2,3,4,5
11	Case Study: Smart Home Automation System using IoT (Control of lights/fans via mobile/web interface with real-time monitoring)	Hardware/Software	1–6	1,2,3,4,5
12	Case Study: Smart Healthcare Monitoring System (Measurement of parameters like temperature/heart rate and cloud-based alert system)	Hardware/Software	1–6	1,2,3,4,5

Note:

Laboratory Assessment:
Assessment:

- Term Work: Term Work shall consist of at least 8 Practical's and 01 (group project) *, 01(Case Study) *(*-Suggested work by group of 3 to 5 students). Also, Term work Journal must include at least 2 assignments.
- Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)
- Practical/ Oral Exam: An Oral examination will be held based on the above syllabus.

Recommended Books:

1. David A. Bell, "Laboratory Manual for Electronic Devices and Circuits", Prentice Hall of India (PHI), 4th Edition, 2001.
2. Muhammad H. Rashid, "SPICE for Circuits and Electronics Using PSPICE", Prentice Hall of India (PHI), 2nd Edition, 1995.
3. Arshdeep Bahga, Vijay Madisetti, "Internet of Things: A Hands-on Approach", Universities Press (India) Pvt. Ltd., 1st Edition, 2015.
4. Simon Monk, "Programming Arduino: Getting Started with Sketches", McGraw Hill Education, 2nd Edition, 2016.
5. Dogan Ibrahim, "Internet of Things Projects with ESP32", Packt Publishing, 1st Edition, 2020.

Term Work:

At least 10 experiments covering the entire syllabus of MD 601 Internet of Things should be set to have well predefined inference and conclusion. The experiments should be student centric and attempts should be made to make the experiments meaningful and interesting. Experiments must be graded from time to time. The grades should be converted into marks as per the Credit and Grading System manual and should be added and averaged. The grading and term work assessment should be done based on this scheme. The final certification and acceptance of term work ensure satisfactory performance of laboratory work and minimum passing marks in term work. Practical and Oral exam will be based on the entire syllabus.

Note:

Suggested List of Experiments is indicative. However, flexibility lies with individual course instructors to design and introducing new, innovative and challenging experiments, from within the curriculum, so that the fundamentals and applications can be explored to give greater clarity to the students and they can be motivated to think differently.

Vertical – 3

<table>
<tr><th rowspan="2">Course Code</th><th rowspan="2">Course Name</th><th colspan="3">Teaching Scheme (Hrs/week)</th><th colspan="4">Credits Assigned</th></tr>
<tr><td>L</td><td>T</td><td>P</td><td>L</td><td>T</td><td>P</td><td>Total</td></tr>
<tr><td rowspan="7">2275311</td><td rowspan="7">Industrial Automation</td><td>2</td><td>--</td><td>--</td><td>2</td><td>--</td><td>--</td><td>2</td></tr>
<tr><td colspan="7">Examination Scheme</td></tr>
<tr><td rowspan="2">Theory</td><td>IA1</td><td>IA2</td><td colspan="2">ESE</td><td colspan="2">Total</td></tr>
<tr><td>--</td><td>--</td><td colspan="2">--</td><td colspan="2">--</td></tr>
<tr><td rowspan="2">Practical</td><td colspan="2">Term Work</td><td colspan="2">Practical Exam</td><td colspan="2">Total</td></tr>
<tr><td colspan="2">50</td><td colspan="2">--</td><td colspan="2">50</td></tr>
</table>

<table>
<tr><td>Pre-requisite Course Codes</td><td colspan="2">(PC302) Electronic Devices,
(PC501) Control System</td></tr>
<tr><td rowspan="5">Course Outcomes</td><td colspan="2">After successful completion students should be able to</td></tr>
<tr><td>CO1</td><td>Analyze industrial automation architecture and hierarchy.</td></tr>
<tr><td>CO2</td><td>Identify, select, and justify appropriate sensors and actuators as per industry standards.</td></tr>
<tr><td>CO3</td><td>Develop and implement basic PLC-based control logic for industrial processes</td></tr>
<tr><td>CO4</td><td>Evaluate industrial communication protocols and SCADA systems</td></tr>
<tr><td rowspan="2"></td><td>CO5</td><td>Apply control system concepts such as PID control in energy-optimized industrial automation systems</td></tr>
<tr><td>CO6</td><td>Analyze real-world industrial processes and propose automation solutions considering ethical practices, environmental sustainability, and Industry 4.0 technologies.</td></tr>
</table>

<table>
<tr><th>Module No.</th><th>Unit No.</th><th>Topics</th><th>Reference</th><th>Hrs.</th></tr>
<tr><td>Module 1</td><td>1</td><td>Introduction to Industrial Automation</td><td></td><td rowspan="4">6Hrs</td></tr>
<tr><td rowspan="3"></td><td>1.1</td><td>Concept of Automation, Types (Fixed, Programmable, Flexible)</td><td rowspan="3">1,2</td></tr>
<tr><td>1.2</td><td>Automation hierarchy (Field level, Control level, Supervisory level) Industrial Automation components overview</td></tr>
<tr><td>1.3</td><td>Industry 4.0 and smart manufacturing</td></tr>
<tr><td rowspan="3">Module 2</td><td>2</td><td>Sensors and Actuators in Automation</td><td></td><td rowspan="3">6Hrs</td></tr>
<tr><td>2.1</td><td>Industrial Sensors: Proximity, Temperature, Pressure, Level, Actuators: Relays, Solenoids, Motors (AC/DC, Stepper, Servo), Sensor characteristics: accuracy, resolution, hysteresis, Industrial interfacing (4–20 mA, 0–10V signals)</td><td rowspan="2">1,2</td></tr>
<tr><td>2.2</td><td>Signal conditioning, Selection criteria for sensors and actuators, safety standards (IEC, ISO basics),Fail-safe design concepts Industrial safety & sensor reliability</td></tr>
</table>

Module 3	**3**	**Programmable Logic Controllers (PLC)**	1,2 6	8 Hrs
	3.1	Introduction to PLC: Architecture, working principle , PLC hardware: CPU, I/O modules, PLC I/O addressing basics		
	3.2	Programming languages: Ladder diagram, basics Timer, Counter, and basic logic implementation, Industrial case studies such as (but not limited to) Conveyor system, Tank filling system, Traffic light system		
Module 4	**4**	**SCADA, Industrial IoT & Data Systems**		7 Hrs
	4.1	Need for industrial communication, real-time requirements, latency and reliability		
	4.2	Industrial protocols: Modbus, Profibus, CAN, Industrial Ethernet (overview), IoT-based communication: MQTT, OPC-UA (introduction),	1,2 ,3,4	
	4.3	Industrial IoT (IIoT): Edge devices, gateways, cloud integration Protocol comparison table (Modbus vs CAN vs Ethernet) One data flow example (sensor → PLC → SCADA → cloud)		
	4.4	SCADA systems: Architecture, components, real-time monitoring, Data acquisition systems and Industrial Data Flow Architecture.		
Module 5	**5**	**Advanced Control Systems & Intelligent Automation**		6Hrs
	5.1	Open-loop and closed-loop control systems, PID control: Tuning methods and industrial significance, PID tuning concept (Ziegler-Nichols basic idea)	1.2,3, 4,5	
	5.2	Industrial control strategies: Adaptive and supervisory control (overview)		
	5.3	Distributed Control Systems (DCS): Architecture and applications, Introduction to AI/ML in industrial automation such as (but not limited to) predictive control, anomaly detection.		
Module 6	**6**	**Smart Manufacturing & Industrial Applications**		6Hrs
		• Automated assembly line • Smart factory architecture • Digital twins and cyber-physical systems • Automation in manufacturing systems, power plant monitoring, Chemical Plant Automation, Smart Irrigation System,	1,2,3, 4,5,6	
			Total	**39**

<u>Course Assessment:</u>

Term Work:
To strengthen the course, a compulsory mini case study on automation failure or AI-based predictive maintenance can be included to highlight real-world relevance and ethical considerations. A small project or experiment, such as sensor-based data logging using a microcontroller/FPGA or a PLC automation demo, can improve practical exposure and industry connection.

Recommended Books:

1. Frank D. Petruzella, "Programmable Logic Controllers", McGraw Hill Education, 5th Edition, 2016.

2. B. R. Mehta, Y. J. Reddy, "Industrial Process Automation Systems: Design and Implementation", Elsevier, 1st Edition, 2014.

3. Krishna Kant, "Computer-Based Industrial Control", PHI Learning, 2nd Edition, 2010.

4. Chanchal Dey, Sunit Kumar Sen, "Industrial Automation Technologies", CRC Press / Taylor & Francis, 1st Edition, 2020.

5. K. S. Manoj, "Industrial Automation with SCADA: Concepts, Communications and Security", Notion Press, 1st Edition, 2019.

6. Sanjay B. Katariya, "Industrial Automation Solutions for PLC, SCADA, Drive and Field Instruments", Notion Press, 1st Edition, 2020.

Online References:

1. **https://onlinecourses.nptel.ac.in/noc22_me59/preview**
2. **https://nptel.ac.in/courses/108105063**

List of Experiments:

Sr. No.	Title of Experiment / Mini Project	Hardware / Software	Module	Reference
1	Simulation of Industrial Automation Hierarchy using Smart Factory Blocks	Draw.io, Factory I/O Community Edition	Module 1	1, 2
2	Virtual Study of Industrial Sensors and Actuators	Tinkercad Circuits	Module 2	1, 2
3	PLC Ladder Logic Simulation for Start-Stop Motor Control	OpenPLC Editor	Module 3	1, 6
4	Conveyor Belt Automation Simulation using Ladder Diagram	OpenPLC, Factory I/O Community Edition	Module 3	1, 2
5	Traffic Light Controller using PLC Simulation	OpenPLC	Module 3	1, 2
6	MQTT Communication Demo between Virtual Devices	Node-RED, Mosquitto MQTT	Module 4	3, 4
7	Mini SCADA Dashboard for Temperature Monitoring	Node-RED, ScadaBR	Module 4	4, 5
8	PID Control Simulation for DC Motor Speed Control	MATLAB Online Trial or Scilab	Module 5	2, 5
9	Predictive Maintenance using Sample Sensor Dataset	Python, Google Colab	Module 5	4, 5
10	Smart Irrigation Logic Simulation using Virtual Sensors	Tinkercad Circuits, Node-RED	Module 6	2, 4

General Instructions for Students:

Students are expected to perform the experiments/projects in self-learning mode using freely available open-source simulation tools and online resources. Before implementation, students should study the basic concepts related to the assigned module through lecture notes, tutorials, or reference videos. Each student should document the objective, procedure, simulation setup, observations, and conclusion with appropriate screenshots/results. The experiments are designed to be completed at home within approximately 2 hours and focus on conceptual understanding, simulation skills, and practical exposure to industrial automation technologies.

Vertical - 4

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Practical	Tutorial	Theory	Practical	Tutorial	Total
2276411	Mini Project - II	---	04	---	---	02	---	02

Course Code	Course Name	Theory					Term Work	Practical / Oral	Total
		Internal Assessment			End Semester Exam	Exam Duration (in Hrs)			
		IAT-I	IAT-II	IAT-I + IAT-II					
2286411	Mini Project - II	---	---	---	---	---	50	---	50

Course Objectives:

Sr. No.	Course Objectives
1	To enable students to identify real-world problems through need-based surveys, literature review, and structured requirement analysis.
2	To develop skills in formulating clear problem statements and project objectives based on feasibility, innovation, and societal relevance.
3	To train students in designing and planning project execution using tools such as Gantt charts, PERT/CPM, and structured methodologies.
4	To provide hands-on experience in developing and implementing functional prototypes or models using appropriate engineering tools and domain-specific technologies.
5	To enhance teamwork, leadership, and project management skills through collaborative group work, progress tracking, and review presentations.
6	To cultivate professional competencies in documentation, technical communication, validation, and evaluation of project outcomes with sustainability and continuous learning focus.

Course Outcomes: On successful completion of the course, the learner will be able to:

Sr. No.	Course Outcomes	Cognitive Levels of Attainment as per Bloom's Taxonomy
1	Formulate a real-world problem by conducting need-based surveys and demonstrating professional responsibility in problem selection.	L2, L3, L4
2	Implement appropriate engineering methods to develop a functional solution while collaboratively exhibiting ethical teamwork behaviour.	L2, L3, L4, L5, L6
3	Demonstrate effective leadership or collaboration by actively coordinating project tasks and responding constructively to group dynamics.	L2, L3, L4
4	Validate experimental results through analytical techniques and justify findings with disciplined technical documentation.	L2, L3, L4, L5, L6

5	Evaluate the societal and environmental impact of the developed solution while applying sustainable engineering practices.	L3, L4, L5
6	Execute project activities using standard engineering tools while exhibiting commitment towards continuous self-directed learning.	L2, L3, L4, L5, L6

Guidelines for Mini Project - II:

1	Students shall form a group of 3 to 4 students, while forming a group shall not be allowed less than three or more than four students, as it is a group activity.
2	Interdisciplinary project is also allowed.
3	Students should do survey and identify needs, which shall be converted into problem statement for project in consultation with faculty supervisor/head of department/internal committee of faculties.
4	Students shall submit implementation plan in the form of Gantt/PERT/CPM chart, which will cover weekly activity of project.
5	A logbook to be prepared by each group, wherein group can record weekly work progress, guide/supervisor can verify and record notes/comments.
6	Faculty supervisor may give inputs to students during project activity; however, focus shall be on self-learning.
7	Students in a group shall understand problem effectively, propose multiple solution and select best possible solution in consultation with guide/ supervisor.
8	Students shall convert the best solution into working model using various components of their domain areas and demonstrate.
9	The solution to be validated with proper justification and report to be compiled in standard format of University of Mumbai.
10	With the focus on the self-learning, innovation, addressing societal problems and entrepreneurship quality development within the students through the project.

Term Work:

The review/ progress monitoring committee shall be constituted by head of departments of each institute. The progress of project to be evaluated on continuous basis, minimum two reviews in the semester.			
In continuous assessment focus shall also be on each individual student, assessment based on individual’s contribution in group activity, their understanding and response to questions.			
Distribution of Term work marks shall be as below:			**Marks**
1	Marks awarded by guide/supervisor based on logbook		5
2	Marks awarded by review committee (Average of Review 1 & Review 2)		40
Project Review - 1			
1	Identification of Problem		4
2	Requirement analysis and Feasibility of the proposed work		4
3	Literature Review		4
4	Objectives of the proposed work		4
5	Methodology of the proposed work		4
	Total Marks		**20**
Project Review - 2			
	a	Planning of project work and team structure	4
	b	Design Methodology	4
	c	Conceptual and Technical Demonstration	4
	d	Presentation: Oral delivery, contact with audience, slides, timing	4
	e	Quality of answers	4
	Total Marks		**20**
	Quality of Project Report		**5**

Online References:

Sr. No.	URL
1	https://www.geeksforgeeks.org/project-idea-college-network/?ref=ml_lbp
2	https://www.simplilearn.com/tutorials/artificial-intelligence-tutorial/ai-project-ideas
3	https://roadmap.sh/backend/project-ideas
4	https://webflow.com/blog/website-ideas
5	https://gist.github.com/MWins/41c6fec2122dd47fdfaca31924647499
6	https://www.projectpro.io/article/artificial-intelligence-project-ideas/461
7	https://github.com/The-Cool-Coders/Project-Ideas-And-Resources
8	https://nevonprojects.com/project-ideas/software-project-ideas/
9	https://roadmap.sh/projects

Vertical – 5

IKS

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2275511	**Indian Knowledge System**	**2**	**2**	**--**	**2**	**2**	**--**	**2**

Course Code	Course Name	Theory					Term Work	Practical / Oral	Total
		Internal Assessment			End Semester Exam	Exam Duration (in Hrs)			
		IAT-I	IAT-II	IAT-I + IAT-II					
2285511	**Indian Knowledge System**	---	---	---	---	---	50	---	50

Rationale:

The proposed IKS modules contextualize core engineering principles across Computer, IT, EXTC, Electronics, and Electrical engineering—within scientifically analyzable Indian knowledge traditions. By mapping indigenous practices to modern technical frameworks, students critically examine foundational engineering concepts through interdisciplinary and systems-based perspectives. This approach strengthens conceptual understanding, promotes analytical reinterpretation of traditional knowledge using contemporary tools, and aligns engineering education with culturally rooted yet technically rigorous inquiry.

Prerequisite: -

Course Objectives:

1. Provide foundational knowledge of Indian acoustic practices.
2. To Understand ancient Indian use of renewable energy
3. Introduce Sanskrit grammar in simple language, with special attention to its rule-based structure.
4. Learners should be able to explain key contributions of ancient Indian mathematics in simple language
5. Understand how Indian traditions of logic, language, knowledge organization, design thinking, governance, sustainability, and wellbeing can support modern IT solutions.
6. Understand the basic structure of Sanskrit grammar and relate selected grammatical ideas to computational modules used in information technology.

Course Outcomes:

On successful completion of the course learner will be able to:

1. Analyze traditional Indian acoustic practices and interpret their relevance to frequency analysis.
2. Apply the traditional knowledge to modern energy solutions and sustainable engineering. 3. Apply Indian Knowledge Systems in linguistics, Sanskrit-based NLP, traditional number systems, measurement practices, and ancient rainfall prediction techniques using calendars, nakshatras, and natural indicators.
4. Apply basic modular arithmetic, pattern recognition, and place-value ideas to simple IT examples such as coding, data representation, and checksums.
5. Apply basic concepts of Indian Knowledge Systems such as logic, language, knowledge organization, ethics, sustainability, and responsible decision-making to design simple technology-based solutions like apps, chatbots, databases, or awareness tools.
6. Applybasic Sanskrit grammar concepts and Pāṇinian rule-based principles to understand formal grammar, algorithms, and simple computational language processing applications in IT.

DETAILED SYLLABUS:

Sr. No	Name of Module	Detailed Content	Hours	CO Mapping
0	Prerequisite	-	-	-
I	Acoustic Science in Vedic Chanting and Indian Music	I.1 Concept of Nāda (Sound) and Shruti • Physical interpretation: frequency, pitch, harmonic content • Shruti system (22 microtones) and frequency discrimination I.2 Resonance and Architectural Acoustics • Temple architecture and echo control • Natural acoustic amplification • Relation to reverberation time and standing waves I.3 Structured Sound Encoding in Vedic Recitation • Pada, Krama, Ghana patha recitation methods • Built-in redundancy → error detection and correction • Parallel to digital communication redundancy schemes **Self-Study:** Study of sounds and analysis in time and frequency domains.	5	CO1

II	**Traditio**	1. Introduction to IKS in renewable energy
		2. Ancient Indian use of renewable energy
		    a. Solar Energy: Solar Architecture, Solar Heating Systems
		    b. Wind Energy: Ancient Windmills, Wind-Powered Water Lifting
		    c. Water (Hydro) Energy: Ancient Watermills, Irrigation Systems
		    d. Bioenergy: Biomass Fuel, Bioenergy Production
		3. Integration of ancient Traditional knowledge of renewable energy with Modern Technologies.
		
		Students shall study the following to understand the use of ancient technologies in Renewable Energy:
		• Gujarat Windmills
		• Watermills in the Indus Valley
		• Konark Sun Temple
			5	**CO2**

III		Linguistics: ● Components of language ● Panini's work on Sanskrit grammar ● Role of Sanskrit in Natural Language Processing Number System and Units of Measurement ● Number system in India- Historical Evidence ● Decimal System ● Unique Approaches to represent Numbers-Katapayadi System ● Pingala and the Binary System Ancient Indian techniques for predicting Rainfall ● Introduction ● Elements of the Indian Calendar ● Panchang and Nakshtras ● Non-bio indicators and Bio indicators **Self-Learning:** ● An Ecosystem for Sanskrit Language Processing ● Salient Features of the Indian Numeral System ● Salient Features of the Indian Numeral System ● Measurement for Time Distance and Weight ● Measurement of Time- An Illustration from Purana	**5**	**CO3**
IV	**Mathematics Foundations in Ancient India and Its Relevance to Information Technology**	4.1: Historical context and number ideas: Indian knowledge systems, the development of numeration, zero as placeholder and number, decimal place value. 4.2: Major contributors and procedural mathematics: Aryabhata, Brahmagupta, Bhaskara II, and examples of rule-based or sutra-like calculation methods in plain language. 4.3: Algorithmic thinking and computation: Historical mathematical procedures connect to algorithms, flowcharts, decomposition, and computational logic. 4.4: Modular arithmetic, binary-style ideas, and data representation: Clock arithmetic, remainders, cyclic reasoning, two-state representation, bits, bytes, and simple binary conversion.	**5**	**CO4**

		4.5: Relevance to IT and synthesis: Coding, checksums/parity, simple cryptography idea, data representation, and how ancient mathematical thinking supports modern IT problem-solving. **Self-Learning Topics and Activities**: ● Zero and place value in daily life ● Algorithmic thinking ● Modular arithmetic ● Binary representation .		
V	**Indian Logic and its applications in Information Technology**	5.1: Introduction to IKS and its relevance to contemporary technology. 5.2: Indian logic, language, and computation with simple examples. 5.3: Knowledge organization, data, and decision-making. 5.4: Ethics, society, sustainability, and responsible digital systems. 5.5 Mini activity where students design a small app, chatbot, database, or awareness tool using one IKS principle. **Self-Learning Topics and Activities**: •Basic ideas of Nyaya logic: observation, inference, debate, and evidence. •Introduction to Paninian grammar and why structured language matters in NLP. •Traditional knowledge classification and its connection with databases and knowledge graphs. •Technology for Indian languages, heritage preservation, digital museums, and public services. •Design thinking activity: identify a local problem and propose an IKS-inspired IT solution.	**5**	**CO5**
VI	**Sanskrit Grammar and Computational Modules**	6.1: Introduction to Indian Knowledge Systems, Sanskrit as structured language, and relevance to IT. 6.2: Basic Sanskrit grammar concepts: sounds, words, roots, suffixes, sandhi, and sentence structure. 6.3: Pāṇinian grammar as a rule-based system; relation with algorithms and formal grammar. 6.4: Computational modules: tokenization, morphology, parsing, language rules, and NLP examples. 6.5: Demonstration of Sanskrit digital tools, reflection activity, and discussion on interdisciplinary applications. **Self-Learning Topics and Activities**: ● Basic Sanskrit alphabet and sound classification. ● Simple sandhi examples and word joining rules. ● Introduction to Pāṇini and the Aṣṭādhyāyī. ● Dhātu, pratyaya, pada, and kāraka in simple terms. ● Basics of natural language processing: tokenization, morphology, and parsing. ● Exploration of Sanskrit digital dictionaries and Sanskrit reader tools. ● Comparison of grammar rules and computer program rules.	**5**	**CO6**

Text and Reference Books or Articles:

1. Acoustics of Chants, Conch-Shells, Bells & Gongs in Worship Spaces. Available: https://www.academia.edu/38208102/Acoustics_of_Chants_pdf?utm_source=chatgpt.com

2. V. R. S. Nalluri, V. J. K. S. Sonti, and G. Sundari, “Analysis of frequency dependent Vedic chanting and its influence on neural activity of humans,” International Journal of Reconfigurable and Embedded Systems, vol. 12, no. 2, pp. 230–239, Jul. 2023, doi: 10.11591/ijres.v12.i2.pp230-239.
3. The Garland Encyclopedia of World Music: South Asia. Available: https://api.pageplace.de/preview/DT0400.9781351544399_A30892028/preview-9781351544399_A30892028.pdf?utm_source=chatgpt.com
4. J. Pattanayak, Y. Belur, and J. Ilavarasu, “Acoustic and psychological effects of Omkar chanting in novice Indian adults: A quasi-experimental study,” Journal of Clinical and Diagnostic Research, 2025, doi: 10.7860/JCDR/2025/77365.21111. Available: https://www.jcdr.net/articles/PDF/21111/77365_CE%5BRa1%5D_F%28IS%29_QC%28PS_SS%29_PF1%28AB_IS%29_redo_PFA%28IS%29_PB%28AB_IS%29_PN%28IS%29.pdf?utm_source=chatgpt. com
5. “Indian Knowledge System: Integrating Heritage with Engineering” by Gagan Bansa, Deep Science Publishing; ISBN: 978-93-49307-77-3 E-ISBN: 978-93-49307-29-2;
6. The textbook of Indian Knowledge Systems, University of Mumbai, ISBN: 978-93-5915-553-1
7. Introduction to Indian Knowledge System: Concepts and Applications, Mahadevan, B., Bhat, Vinayak Rajat, Nagendra Pavana R.N. by PHI Learning Pvt Ltd. Delhi, ISBN: 978-93-91818-20-3
8. Echoes of the Monsoon: Ancient Indian techniques for predicting Rainfall, Janak Joshi, Kandarp
9. Parmar, Lakhan Jain, Sharmistha Bhowmik and Bindu Bhatt, DOI: https://www.doi.org/10.22271/27067483.2025.v7.i7a.386.
10. Kim Plofker, Mathematics in India: A clear historical overview of Indian mathematical developments, suitable for faculty use and advanced self-learners.
11. George Gheverghese Joseph, The Crest of the Peacock: An accessible introduction to non-European mathematical traditions, including India.
12. Charles Petzold, Code: The Hidden Language of Computer Hardware and Software: A beginner-friendly bridge from numbers and symbols to data representation and computing.
13. Simon Singh, The Code Book: Useful for simple explanations of ciphers, encryption, and the mathematics behind communication security.
14. T. A. Sarasvati Amma, Geometry in Ancient and Medieval India: A more specialized reference for deeper exploration of traditional mathematical methods.
15. Mahadevan, B., Bhat, V. R., and Nagendra Pavana R. N. Introduction to Indian Knowledge System: Concepts and Applications. PHI Learning.
16. Kapur, K., and Singh, A. K. (Eds.). Indian Knowledge Systems, Vol. 1. Indian Institute of Advanced Study, Shimla.
17. Bora, M. C. Bharatiya Knowledge Systems. Khanna Publishing House.
18. Rao, S. S. Engineering and Technology in Ancient India. Indian National Science Academy / related academic editions.
19. Selected open articles or chapters from IKS Division / AICTE / Ministry of Education resources may be used as supplementary reading.
20. S. D. Joshi and J. A. F. Roodbergen, The Aṣṭādhyāyī of Pāṇini.
21. V. N. Jha, The Philosophy of Sanskrit Grammar.
22. Peter M. Scharf, Sanskrit Syntax and Discourse Structures.
23. Daniel Jurafsky and James H. Martin, Speech and Language Processing — for introductory NLP concepts.
24. Rajeev Sangal, Akshar Bharati, and Vineet Chaitanya, Natural Language Processing: A Paninian Perspective.

Other Online References:

Sr. No.	Website Name
1.	NPTEL course page: For a structured academic overview of mathematics in India. https://nptel.ac.in/courses/111101080
2.	NPTEL YouTube playlist: For lecture-based learning that can support flipped or self-paced study. https://www.youtube.com/playlist?list=PLbMVogVj5nJThf31TNSQzuN7zqxe7HdRN
3.	IKS India: For broader Indian Knowledge Systems context and additional learning resources. https://www.nist.gov/cryptography
4.	Responsible AI and India-context resources: https://indiaai.gov.in/
5.	IKS Division, Ministry of Education / AICTE: https://iksindia.org/
6.	NPTEL: Introduction to Pāṇinian Grammar URL: https://onlinecourses.nptel.ac.in/noc24_hs44/preview
7.	Sanskrit Heritage Reader URL: https://sanskrit.inria.fr/

List of Experiments:

Sr No	List of Experiments	Hrs
01	Record a temple bell sound, conch sound, or classical “Sa” note and analyze it by plotting its time-domain waveform and frequency spectrum, identifying the fundamental frequency and harmonics, and comparing the observed harmonic structure with theoretical frequency ratios of Just Intonation and Equal Temperament.	04
02	Observe the movement of sunlight and shadow around a building, monument, or open courtyard at different times of the day, record the shadow directions and lengths, and analyze how solar position can guide orientation, time estimation, and passive architectural planning in the context of traditional Indian solar design.	04
03	Examine images, plans, or a simple model of the Konark Sun Temple or any sun-oriented structure, identify how geometry, orientation, and sunlight interaction are used in the design, and prepare a visual explanation of how ancient solar architecture supports modern sustainable building concepts.	04
04	Construct a small paper or cardboard windmill, test its rotation under natural wind or fan airflow at different speeds, and document how blade movement demonstrates the basic principle of converting wind energy into mechanical motion, linking it to traditional wind-powered systems and modern turbines.	04
05	Build a simple working model using cups, straws, thread, or cardboard to simulate water lifting or channel-based irrigation, observe how water can be moved with minimal energy input, and relate the model to ancient Indian irrigation systems and watermill-based practices.	04
06	Survey homes or nearby community spaces to identify traditional biomass fuels such as wood, agricultural residue, or cow-dung cakes, record their current use, benefits, and limitations, and compare the observed practices with modern ideas of bioenergy and sustainable fuel management..	04
07	Pronounce and classify selected Sanskrit sounds into vowels and consonants using a simple sound chart, group them according to articulation patterns, and analyze how this systematic sound organization supports pronunciation training, language learning, and speech-processing applications.	04
08	Create all possible combinations of short and long syllabic patterns of a fixed length, tabulate them systematically, and compare the resulting patterns with binary 0/1 combinations to understand how ancient combinatorial thinking connects with digital representation.	04
09	Solve real-life problems based on clocks, calendars, weekdays, or repeated events using modular arithmetic, record the remainder-based reasoning used in each case, and explain how cyclic patterns support time management, scheduling, and introductory cryptographic logic.	04

10	Convert selected decimal numbers into binary form, arrange them in bit patterns, append parity bits for even or odd parity, and interpret how binary representation and simple error-detection methods are used in digital systems for reliable data storage and transmission.	04
11	Use a basic Sanskrit dictionary, transliteration tool, or reader tool to input selected words, observe the output such as root, meaning, or grammatical analysis, and describe how such tools support language technology and digital preservation.	04
12	Select a real-life problem such as fake-news verification, waste segregation, or campus awareness, identify observation, evidence, inference, and conclusion using the basic framework of Indian logic, and represent the reasoning process as a decision tree, chatbot flow, or awareness tool concept.	04
13	Take a simple historical or rule-based calculation procedure associated with Indian mathematics, rewrite it as a sequence of input-process-output steps, and convert it into a flowchart to demonstrate the connection between procedural mathematics and modern algorithmic thinking.	04
14	Represent selected large and small numbers using a place-value table, deliberately insert or remove zero in different positions, and analyze how zero functions both as a number and as a placeholder, showing its importance in arithmetic, accounting, and digital data representation.	04
Sr No	**List of Assignments**	**Hrs**
01	Study the concepts of Nāda and Shruti, temple acoustics, and Vedic recitation methods, and briefly explain their relation to modern concepts of frequency, resonance, reverberation, and redundancy/error correction in digital communication systems. Also write a short note on how Indian Knowledge Systems demonstrate scientific understanding of sound and communication.	01
02	Study the ideas of solar architecture in traditional Indian structures such as the Konark Sun Temple or courtyard-based houses, and explain how sunlight, orientation, shading, and ventilation were used for thermal comfort and energy efficiency. Also write a short note on how these traditional design ideas are relevant to modern sustainable architecture and climate-responsive planning..	01
03	Study one traditional Indian renewable energy practice such as wind-powered water lifting, watermills, solar heating, or biomass use, and compare it with one modern renewable energy technology. Explain how traditional knowledge systems supported sustainable living and discuss how these ideas can inspire present-day energy conservation and green technology.	01
04	Study the basic ideas of Pāṇinian grammar as a structured rule-based system, and explain how concepts such as sounds, roots, suffixes, and word formation relate to modern ideas in natural language processing, machine translation, and language analysis. Also write a short note on why Sanskrit grammar is often considered relevant to computational linguistics.	01
05	Study the historical development of zero as both a number and a placeholder, and explain how the decimal place-value system simplified arithmetic and record keeping. Also write a short note on how zero and place value continue to be essential in digital computation, data storage, and computer-based numerical processing.	01
06	Study the basic ideas of Indian logic, especially observation, inference, debate, and evidence in the Nyāya tradition, and explain how these ideas support careful reasoning and decision-making. Also write a short note on how such logical methods can be applied in present-day contexts such as data analysis, fake-news verification, ethical decision-making, and responsible digital systems.	01
07	Identify one idea from Indian Knowledge Systems in the syllabus, such as renewable energy, Indian numeration, Indian logic, or Sanskrit grammar, and propose a simple real-life solution such as an app, chatbot, awareness campaign, digital archive, poster, database, or educational tool based on that idea. Explain the problem being addressed, the IKS principle used, and how the proposed solution can be useful in present-day society.	01
08	Study the basic Sanskrit grammar concepts of sounds, words, roots, suffixes, sandhi, and sentence structure, and explain how grammar rules work like structured instructions. Also compare grammar rules with computer program rules, and briefly discuss how tokenization, morphology, and parsing in computational systems resemble traditional rule-based language analysis.	01

Assessment:

Term Work:

Term work shall consist of at least 10-12 practical’s based on above list. Also Term work Journal must include at least 4 assignments

Term Work Marks:

50 Marks (Total marks) = 25 Marks (Experiment) + 20 Marks (Assignments) + 5 Marks (Attendance)

Vertical 6

<table>
<tr><th rowspan="2">Course Code</th><th rowspan="2">Course Name</th><th colspan="3">Teaching Scheme (Hrs/week)</th><th colspan="4">Credits Assigned</th></tr>
<tr><td>L</td><td>T</td><td>P</td><td>L</td><td>T</td><td>P</td><td>Total</td></tr>
<tr><td rowspan="4">2276611</td><td rowspan="4">Data Structures</td><td>--</td><td>--</td><td>2*+2</td><td>--</td><td>--</td><td>2</td><td>2</td></tr>
<tr><td colspan="7">Examination Scheme</td></tr>
<tr><td></td><td colspan="2">Term Work</td><td colspan="2">Practical Exam</td><td colspan="2">Total</td></tr>
<tr><td>Practical</td><td colspan="2">25</td><td colspan="2">-</td><td colspan="2">25</td></tr>
</table>

<table>
<tr><td>Pre-requisite Course Codes</td><td colspan="2"></td></tr>
<tr><td></td><td colspan="2"></td></tr>
<tr><td rowspan="6">Course Outcomes</td><td colspan="2">After the successful completion students should be able to</td></tr>
<tr><td>LO1</td><td>Implement various operations of linear data structures.</td></tr>
<tr><td>LO2</td><td>Implement various operations of non-linear data structures.</td></tr>
<tr><td>LO3</td><td>Develop solutions for real world problems by selecting appropriate data structure and algorithms.</td></tr>
<tr><td>LO4</td><td>Analyze electronic Circuits using MOSFET, its DC and AC analysis</td></tr>
<tr><td>LO5</td><td>Analyze the complexity of the given algorithms.</td></tr>
</table>

<table>
<tr><th>Module No.</th><th>Lab No.</th><th>Name of the experiment</th><th>Refe renc e</th><th>Hrs.</th></tr>
<tr><td rowspan="3">Module 1</td><td></td><td>Stack ADT: Operations on stack, Applications of stack</td><td rowspan="3">1,2</td><td rowspan="3">6Hrs</td></tr>
<tr><td>1</td><td>a. Implement Stack ADT using array
b. Analyze time complexities of all operations of stack.</td></tr>
<tr><td>2</td><td>a. Convert Infix to Postfix and evaluate the postfix using Stack ADT</td></tr>
<tr><td rowspan="3">Module 2</td><td></td><td>Queue ADT : Operations on queue, Types of queues: Linear, Circular, Priority, Applications of queue</td><td rowspan="3">1,2</td><td rowspan="3">6Hrs</td></tr>
<tr><td>3</td><td>a. Implement Linear Queue ADT using array. OR
b. Implement Circular Queue ADT using array</td></tr>
<tr><td>4</td><td>a. Implement Priority Queue ADT using array.
b. Analyze time complexities of all operations of queue</td></tr>
<tr><td rowspan="3">Module 3</td><td></td><td>Linked List ADT, Types of Linked List: Singly, Circular, Doubly LL, Applications of linked list</td><td rowspan="3">1,2</td><td rowspan="3">10 Hrs</td></tr>
<tr><td>5</td><td>a. Implement following operations on Singly Linked List ADT.
1. Insert (all three)
2. Delete (all three)
3. Split
4. Concatenate
5. Copy
6. Reverse
7. Traverse
b. Analyze time complexities of all operations of linked list</td></tr>
<tr><td>6</td><td>a. Implement following operations on Circular Linked List ADT
1. Insert (all three)</td></tr>
</table>

		2. Delete (all three) 3. Traverse b. Analyze time complexities of all operations of circular linked list.		
	7	a. Implement Doubly Linked List ADT. 1. Insert (all three) 2. Delete (all three) 3. Traverse Forward 4. Traverse Backward b. Analyze time complexities of all operations of doubly linked list.		
	8	a. Addition of two polynomials using linked list		
Module 4		**Tree terminologies, Binary Tree, Tree traversal techniques, BST ADT, Expression Tree. Applications of Binary Tree**	1,2	10 Hrs
	9	a. Implement following operations on Binary Search Tree ADT using Linked List. 1. Insert 2. Delete (All three cases) 3. Search 4. Display b. Analyze the time complexities of all the operations on BST		
	10	Construct an expression tree from given postfix form of expression.		
	11	Implement a program to represent infix, prefix and postfix form of arithmetic expressions using binary tree traversal techniques. The expression is represented as a binary tree, where each operator is a parent node, and its operands are the left and right children		
Module 5		**Graph terminologies, Graph representations, Graph traversal techniques.**	1,2	4 Hrs
	12	Implement a program to represent a graph using an adjacency list or adjacency matrix data structure. Then, perform non recursive breadth first search (BFS)		
	13	Implement non recursive depth-first search (DFS) traversal algorithms on Graph represented using Adjacency matrix.		
Module 6		**Sorting: Insertion sort, selection sort, quick sort, merge sort**	1,2	6Hrs
	14	a. Implement Modified Bubble Sort, Insertion sort and Selection sort. b. Analyze the time complexity of the algorithms.		
	15	a. Implement Quick sort. b. Analyze the time complexity of the algorithm.		
	16	a. Implement Merge sort b. Analyze the time complexity of the algorithm		
Module 7		**Searching and Hashing: Binary search, Hashing, Hash functions, Collision resolution techniques**	1,2	6Hrs
	17	a. Implement the binary search algorithm to search for a key element in a sorted array. b. Analyze the time complexity of the algorithm		
	18	a. Implement a hash table data structure using an array and handle collisions using chaining (linked lists) b. Analyze the time complexity of the algorithm.		
	19	a. Implement a hash table data structure using an array and handle collisions using linear probing b. Analyze the time complexity of the algorithm		

Mini Project: (Suggested list of Mini Project Topics)				
		a. Text Edition Application: Implement a text editor with an undo feature using a stack. Every time a change is made to the text, push the previous state onto the stack. When the user performs an undo operation, pop the last state from the stack and revert the text to that state **b.** Develop a print job scheduler using a queue. Users submit print jobs to the queue, and they are processed in the order they were received. Once a job is completed, it is dequeued from the queue. **c.** Implement a task management system using a singly linked list. Each task is represented as a node in the linked list, containing information such as task description, priority, and deadline. Users can add, edit, delete, and search for tasks within the linked list **d.** Create a browser history manager using a doubly linked list. Each webpage visited is represented as a node in the linked list, with pointers to the previous and next pages. Users can navigate forward and backward through their browsing history, add new pages, delete visited pages, and search for specific pages. **e.** Implement a dictionary using a binary search tree (BST). Each node in the tree represents a word and its associated meaning. The tree is organized such that for any node, all words to the left have lexicographically smaller values, and all words to the right have lexicographically greater values. **f.** Given a network of cities connected by roads with different weights representing distances, find the minimum spanning tree to connect all cities with minimum total distance.		
			Total	**39**

Assessment:

Term Work: Term Work shall consist of at least 10 to 12 practicals' based on the above list. Also, Term work Journal must include at least 2 assignments.

Term Work Marks: 25 Marks (Total marks) = 15 Marks (Experiment) + 5 Marks (Assignments) + 5 Marks (Attendance)

Practical/ Oral Exam: An Oral examination will be held based on the above syllabus.

Recommended Books:

1. "Data Structures using C and C++" Yedidyah Langsam, Moshe J. Augenstein, Aaron M. Tenenbaum
2. "Data Structures using C", Reema Thareja, Third Edition.
3. "Data Structures and Program Design in C++", Robert L. Kruse, Alexander J. Ryba, Prentice-Hall India.
4. "Data Structures and Algorithm in Java", Goodrich and Tamassia, John Wiley and Sons, Sixth Edition 2014.
5. "Data Structures and Pseudocode approach with C", 2nd Edition by Richard F. Gilberg & Behrouz A. Forouzan

Online References:

NPTEL courses on data structures

Term Work:

At least 10 experiments covering the entire syllabus of Data Structure should be set to have well predefined inference and conclusion. The experiments should be student centric and attempts should be made to make the experiments meaningful and interesting. Experiments must be graded from time to time. The grades should be converted into marks as per the Credit and Grading System manual and should be added and averaged. The grading and term work assessment should be done based on this scheme. The final certification and acceptance of term work ensure satisfactory performance of laboratory work and minimum passing marks in term work. Practical and Oral exam will be based on the entire syllabus.

Detail Syllabus
(Common Syllabus as per BoS Groups)

QUESTION PAPER PATTERN
(External and Internal)

Letter Grades and Grade Points:

Semester GPA/ Programme CGPA Semester/ Programme	% of Marks	Alpha-Sign/ Letter Grade Result	Grading Point
9.00 - 10.00	90.0 – 100	O (Outstanding)	10
8.00 - < 9.00	80.0 - < 90.0	A+ (Excellent)	9
7.00 - < 8.00	70.0 - < 80.0	A (Very Good)	8
6.00 - < 7.00	60.0 - < 70.0	B+ (Good)	7
5.50 - < 6.00	55.0 - < 60.0	B (Above Average)	6
5.00 - < 5.50	50.0 - < 55.0	C (Average)	5
4.00 - < 5.00	40.0 - < 50.0	P (Pass)	4
Below 4.00	Below 40.0	F (Fail)	0
Ab (Absent)	-	Ab (Absent)	0

Sd/-

Dr. R. N. Awale
BoS-Chairman-Electronics and Computer Engineering
Faculty of Technology

Sd/-

Dr. Deven Shah
Associate Dean
Faculty of Science & Technology

Sd/-

Prof. Shivram S. Garje
Dean
Faculty of Science & Technology