AC –
Item No. –

As Per NEP 2020

University of Mumbai

Syllabus for Major Vertical – 1, 4 & 5		
Name of the Programme – B.E. (Computer Engineering)		
Faulty of Engineering		
Board of Studies in Computer Engineering		
U.G. Third Year Programme	Exit Degree	B.Sc. Tech Engineering- 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. (Computer Engineering)**
2	**Exit Degree**	**B.Sc. Tech Engineering-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**

DRAFT

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

DRAFT

B.Sc. Tech Engineering- Computer Engineering .

Credit Structure (Sem. V & VI)

	R: _____C									
Level	**Semester**	**Major**		**Minor**	**OE**	**VSC, SEC (VSEC)**	**AEC, VEC, IKS**	**OJT, FP, CEP, CC,RP**	**Cum. Cr. / Sem.**	**Degree/ Cum. Cr.**
		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]

DRAFT

T.E.
Scheme

DRAFT

Program Structure for Third Year of 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
2115111	Software Engineering	3	--	--	3	--	--	3
2115112	Computer Network	3	–	--	3	--	--	3
2115113	Artificial Intelligence & Soft Computing	3	--	--	3	--	--	3
211511#	Program Electives -I	3	--	--	3	--	--	3
2115211	Multi-Disciplinary Minor	3	--	--	3	--	--	3
2115117	Software Engineering Lab	--	2	--	--	--	1	1
2115118	Computer Network Lab	--	2	--	--	--	1	1
2115119	Artificial Intelligence & Soft Computing Lab	--	2	--	--	--	1	1
211512#	Program Electives -I Lab	--	2	--	--	--	1	1
2115212	Multi-disciplinary Minor Lab	--	2@	--	--	--	1	1
2115311	To be taken from the bucket provided by the University from other Faculty	2#	--	---	2	--	--	2
2115511	Indian Knowledge System (Syllabus common as per Group)	--	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 -I	
Course Code	**Course Description**
2115114	Datawarehouse & Mining
2115115	Computer Graphics
2115116	Internet Programming
Lab Code	**Lab Description**
2115120	Datawarehouse & Mining Lab

DRAFT

2115121	Computer Graphics Lab
2115122	Internet Programming Lab

Program Structure for Third Year of Computer Engineering

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)					
2115111	Software Engineering	20	20	40	60	2	--	--	100
2115112	Computer Network	20	20	40	60	2	--	--	100
2115113	Artificial Intelligence & Soft Computing	20	20	40	60	2	--	--	100
211511#	Program Electives -I	20	20	40	60	2	--	--	100
2115211	Multi-disciplinary Minor	20	20	40	60	2	--	--	100
2115117	Software Engineering Lab	--	--	--	--	--	25	25	50
2115118	Computer Network Lab	--	--	--	--	--	25	25	50
2115119	Artificial Intelligence & Soft Computing Lab	--	--	--	--	--	25	25	50
211512#	Program Electives -I Lab	--	--	--	--	--	25	-	25
2115212	Multi-disciplinary Minor Lab	--	--	--	--	--	25	25	50
2115311	To be taken from the bucket provided by the University from other Faculty	--	--	--	--	--	50	--	50
2115511	Indian Knowledge System (Syllabus common as per Group)	--	--	--	--	--	50	--	50

DRAFT

Total	100	100	200	300	10	225	100	825

Program Electives -I	
Course Code	**Course Description**
2115114	Datawarehouse & Mining
2115115	Computer Graphics
2115116	Internet Programming
Lab Code	**Lab Description**
2115120	Datawarehouse & Mining Lab
2115121	Computer Graphics Lab
2115122	Internet Programming Lab

Program Structure for Third Year of 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
2116111	Machine Learning	3	--	--	3	--	--	3
2116112	System Programming & Compiler Construction	3	--	--	3	--	--	3
211611#	Program Electives -II	3	--	--	3	--	--	3
211611#	Program Electives -III	3	--	--	3	--	--	3
2116211	Multi-disciplinary Minor	3	--	--	3	--	--	3
2116411	Mini Project-II	--	4	--	--	--	2	2
2116511	Professional Skill	--	2*+2	--	--	--	2	2
2116119	Machine Learning Lab	--	2	--	--	--	1	1
2116120	System Programming & Compiler Construction Lab	--	2	--	--	--	1	1
211612#	Program Electives -II Lab	--	2	--	--	--	1	1
211612#	Program Electives -III Lab	--	2	--	--	--	1	1
2116212	Multi-disciplinary Minor Lab	--	2@	--	--	--	1	1

DRAFT

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	2116113	Internet of Things	2116121	Internet of Things Lab
	2116114	Digital Imaging Techniques and Analysis	2116122	Digital Imaging Techniques and Analysis Lab
	2116115	Natural Language Processing	2116123	Natural Language Processing Lab
Program Electives -III	2116116	Augmented and Virtual Reality	2116124	Augmented and Virtual Reality Lab
	2116117	Blockchain Technology	2116125	Blockchain Technology Lab
	2116118	User Experience Design	2116126	User Experience Design Lab

DRAFT

Program Structure for Third Year of Computer Engineering

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)					
2116111	Machine Learning	20	20	40	60	2	--	--	100
2116112	System Programming & Compiler Construction	20	20	40	60	2	--	--	100
211611#	Program Electives -II	20	20	40	60	2	--	--	100
211611#	Program Electives -III	20	20	40	60	2	--	--	100
2116211	Multi-disciplinary Minor	20	20	40	60	2	--	--	100
2116411	Mini Project-II	--	--	--	--	--	50	--	50
2116511	Professional Skill	--	--	--	--	--	25	--	25
2116119	Machine Learning Lab	--	--	--	--	--	25	25	50
2116120	System Programming & Compiler Construction Lab	--	--	--	--	--	25	25	50
211612#	Program Electives -II	--	--	--	--	--	25	25	50

	Lab								
211612#	Program Electives -III Lab	--	--	--	--	--	25	25	50
2116212	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	2116113	Internet of Things	2116121	Internet of Things Lab
	2116114	Digital Imaging Techniques and Analysis	2116122	Digital Imaging Techniques and Analysis Lab
	2116115	Natural Language Processing	2116123	Natural Language Processing Lab
Program Electives -III	2116116	Augmented and Virtual Reality	2116124	Augmented and Virtual Reality Lab
	2116117	Blockchain Technology	2116125	Blockchain Technology Lab
	2116118	User Experience Design	2116126	User Experience Design Lab

DRAFT

Vertical – 1

Major

DRAFT

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2115111	Software Engineering	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					
2115111	Software Engineering	20	20	40	60	2	--	--	100

Course Objectives:

1. To provide knowledge of the software engineering discipline.
2. To apply analysis, design and testing principles to software project development.
3. To demonstrate and evaluate real-time projects with respect to software engineering principles.
4. To develop project planning, scheduling, and tracking skills using standard software engineering tools and techniques.
5. To apply quality assurance, risk management, and maintenance strategies in software development.
6. To introduce the integration of emerging technologies such as AI and ML concepts within software engineering practices.

Course Outcomes:

1. Demonstrate basic knowledge in software Engineering.
2. Identify requirements, analyze and prepare models.
3. Plan, schedule and track the progress of the projects.
4. Design & develop the software projects.
5. Apply testing principles on software projects.
6. Identify risks, manage the change to assure quality and understand the maintenance and concepts of AI/ML concepts in software projects

Prerequisite: Object Oriented Programming with Java , Python Programming

DETAILED SYLLABUS:

Sr. No.	Module	Detailed Content	Hours	CO Mapping
0	**Prerequisites and Course outline**	Class ,Objects, Examples Characteristics of OOPS , Design Thinking		
I	**Introduction To Software Engineering and Process Models**	Nature of Software, Software Engineering, Software Process, Capability Maturity Model (CMM). Prescriptive Process Models: The Waterfall, Incremental Process Models, Evolutionary Process Models: RAD & Spiral.Agile process model: Extreme Programming (XP), Scrum, Kanban, Introduction to AI/ML concepts in software Engineering. Advanced trends in software engineering.	**08**	**CO1**

DRAFT

		Self-Learning Topics:-Applications of Agile approach(To do list for small projects),Case studies of on software process models.		
II	**Software Requirements Analysis and Modeling**	Requirement Engineering, Requirement Modeling, Scenario based model. (Usecase & Activity diagram) Software Requirement Specification document format(IEEE), Data flow diagram **Self-learning Topics:** More examples on Scenario based models and SRS preparation.	**05**	**CO2**
III	**Software Estimation Metrics**	Software Metrics, Software Project Estimation (LOC, FP, COCOMO), Project Scheduling using Gantt & PERT Chart **Self-learning Topics:** Project Management -Case studies/case studies of specific examples(Ex-Disaster Management)	**07**	**CO3**
IV	**Software Design**	Design Principles & Concepts, Effective Modular Design, Cohesion and Coupling, Architectural design, UML Diagrams **Self-learning Topics:** Case studies on UML diagrams on more real time applications	**07**	**CO4**
V	**Software Testing**	Unit testing, Integration testing, validation testing, System testing, Testing Techniques White-box testing: Basis path, Control structure testing Black-box testing: Graph based Static Testing, Equivalence, Boundary Value analysis **Self-learning Topics:** Preparing test cases for real time applications using V-model, usecase diagrams	**06**	**CO5**
VI	**Software Configuration Management, Quality Assurance and Maintenance**	Risk Analysis & Management: Risk Mitigation, Monitoring and Management Plan (RMMM). Quality Concepts and Software Quality assurance Metrics, Formal Technical Reviews, Software Reliability, The Software Configuration Management (SCM). Version Control and Change Control. Types of Software Maintenance, Re-Engineering, Reverse Engineering. **Self-learning Topics:** Difference of software Engineering using traditional and AI/ML approach.	**06**	**CO6**

DRAFT

Text Books:

1. Roger Pressman, “Software Engineering: A Practitioner‘s Approach”, 9th edition ,McGraw-Hill Publications, 2019
2. Ian Sommerville, “Software Engineering”, 9th edition, Pearson Education, 2011
3. Ali Behfrooz and Fredeick J. Hudson, "Software Engineering Fundamentals", Oxford University Press, 1997
4. Grady Booch, James Rambaugh, Ivar Jacobson, “The unified modeling language user guide”, 2nd edition, Pearson Education, 2005
5. Dr.Hitesh Mohapotra, Dr.Amiya Kumar Rath**, “Fundamentals of Software Engineering: Comprehensive insights into SDLC design quality and AI/ML in software” -, 2nd Edition, 2025.**

References:

1. Pankaj Jalote, "An integrated approach to Software Engineering", 3rd edition, Springer,2005
2. Rajib Mall, "Fundamentals of Software Engineering", 5th edition, Prentice Hall India, 2014
2. Jibitesh Mishra and Ashok Mohanty, “Software Engineering”, Pearson , 2011
3. Ugrasen Suman, “Software Engineering – Concepts and Practices”, Cengage Learning,2013
4. Waman S Jawadekar, “Software Engineering principles and practice”, McGraw Hill Education, 2004

Online References:

Sr. No.	Website Name
1.	https://onlinecourses.nptel.ac.in/noc25_cs108/preview
2.	https://onlinecourses.swayam2.ac.in/cec25_cs08/preview
3.	https://onlinecourses.nptel.ac.in/noc23_cs46/preview
4	https://www.scirp.org/journal/paperinformation?paperid=93452

Assessment:

Internal Assessment (IA) for 20 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

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.

DRAFT

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.

DRAFT

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2115112	Computer Network	3	-	-	3	-	-	3

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				
2115112	Computer Network	20	20	40	60	-	-	100

Course Objectives

1. To provide foundational knowledge of computer networks, network architectures, devices, and transmission techniques essential for understanding data communication systems.
2. To understand the principles, design issues, and protocols of the data link layer for reliable and efficient data communication.
3. To understand network layer functions, routing algorithms, and protocols that ensure efficient data delivery and congestion control in computer networks.
4. To understand the role and functionality of the transport layer and the design and performance of modern network architectures in computer networks.
5. To understand the concepts, paradigms, and protocols of the application layer in computer networks.
6. To familiarize students with the latest trends in computer networking technologies.

Course Outcomes

1. Explain the functions of network devices, topologies, data transmission methods and encoding schemes used in modern communication networks.
2. Apply framing techniques to demonstrate error and flow control mechanisms.
3. Design small network topologies using appropriate routing algorithms.
4. Understand and differentiate the role and protocols of the transport layer in providing reliable communication.
5. Explain standard client-server applications and functioning of protocols such as HTTP, FTP, DNS, Email, TELNET, and SSH.
6. Identify common network security threats, vulnerabilities, and corresponding mitigation techniques.

Prerequisite: Programming Fundamentals, Basics of Electronics & Communication.

DRAFT

DETAILED SYLLABUS

Sr. No.	Module	Detailed Content	Hours	CO Mapping
0	**Prerequisite**	1. Number systems and data representation (binary, hexadecimal, encoding) 2. Fundamentals of computer organization 3. Operating system basics (processes, threads, file I/O) 4. Data structures and algorithms (arrays, stacks, queues, graph concepts)		
I	**Fundamentals of Computer Networks**	1.1 LAN, MAN, WAN, Wireless networks, Networks Software, Protocol, 1.2 Design issues for the Network layers. Network Models 1.3 OSI Reference Model, TCP/IP Model, Network Topologies, Types of Transmission Medium. Network Architectures: Client-Server, Peer To Peer. 1.4 Network Devices: Bridge, Switch, Router, Gateway, Access Point **Self-learning Topics:** Encodings, Frequency Hopping (FHSS) and Direct Sequence Spread Spectrum (DSSS).	**6**	**CO1**
II	**Data Link Layer and MAC Sub-layer**	2.1 Introduction, functions. 2.2 Design Issues: Services to Network Layer, Framing. Error Detection and correction, Parity Bits, Checksum, Hamming Codes and CRC. 2.3 Flow Control Protocols: Unrestricted Simplex, Stop and Wait, Sliding Window Protocol. 2.4 PPP and HDLC. 2.5 Pure and Slotted ALOHA, CSMA, CSMA/CD, CSMA/CA, 2,6 Introduction to IEEE standards (IEEE 802.3, IEEE 802.11). **Self-learning Topics:** Demonstration of DLL protocols on Simulator	**8**	
III	**Network Layer and Routing Algorithms**	3.1 Introduction to Network Layer, Network Layer Protocols, Routing Concepts, IIPv4 and IPv6 , addressing Subnetting and Supernetting 3.2 Routing Algorithms: Static Routing Algorithms (Non-adaptive), Dynamic Routing Algorithms (Adaptive). 3.3 Internet Routing Protocols, Congestion Control and Quality of Service (QoS). **Self-learning Topics:** QoS mechanisms and	**6**	**CO3**

DRAFT

		parameters		
IV	**Transport Layer Protocols**	4.1 The Transport Service: Transport service primitives, Berkeley Sockets, TCP Characteristics , header structure, TCP Connection management (Handshake), TCP timers, UDP Characteristics, header structure , Applications 4.2 TCP Flow control (sliding Window), TCP Congestion Control: slow start, The additive-increase/multiplicative-decrease (AIMD), Congestion avoidance (DECBit , RED), SCTP –Quality of service **Self-learning Topics:** Transport service primitives and Sockets programming	**7**	**CO4**
V	**Application Layer and Network Services**	5.1 Application layer Paradigms, Client-Server Paradigm: Application Programming Interface 5.2 Standard Client Server applications: World Wide Web and HTTP, FTP, TELNET, Secure Shell (SSH), Domain Name System (DNS) **Self-learning Topics:** Additional Application layer protocols SFTP, and SCP, SMTP, POP3, and IMAP	**6**	**CO5**
VI	**Emerging Network Technologies and Security**	6.1 MPLS : introduction, header, working, advantages and disadvantages 6.2 RADIO SPECTRUM: 5G spectrum landscape and requirements, Spectrum access modes and sharing scenarios, 5G spectrum technologies. 5G architecture 6.3 Introduction and need to study cyber security, Cyber Crime: Cybercrime definition, Types of Cybercrime **Self-learning Topics:** Cyber-Attacks and Vulnerabilities and precautions.	**6**	**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. R. Howard, *Cyber Security Essentials*, 1st ed. Boca Raton, FL: CRC Press, 2010.

References:

1. Computer Networking: A Top-Down Approach (7th Edition) by James F. Kurose and Keith W. Ross
2. TCP/IP Illustrated, Volume 1: The Protocols by W. Richard Stevens
3. Data and Computer Communications by William Stallings
4. S. Ahmadi, *5G NR: Architecture, Technology, Implementation, and Operation of 3GPP New Radio Standards*, 1st ed. Burlington, MA, USA: Academic Press (Elsevier), 2019.

Online Resources:

Sr. No	Website Name

DRAFT

1	https://docs.oracle.com/cd/E18752_01/html/816-4554/ipov-6.html
2	https://onlinecourses.nptel.ac.in/noc22_cs19/preview
3	https://www.youtube.com/watch?v=CaukSKg_sl0

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.

DRAFT

University of Mumbai

Course Code: _____and Course Name: .

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

NB: (3) Question 1 is compulsory. (4) 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

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2115113	Artificial Intelligence & Soft Computing	3	-	-	3	-	-	3

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	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-I	IAT-II	IAT-I + IAT-II					
2115113	Artificial Intelligence & Soft Computing	20	20	40	60	2	--	--	100

Course Objectives: To conceptualize the basic ideas and techniques of AI and SC.

1. To understand fundamental concepts of artificial intelligence and soft computing
2. To formulate a problem statement and gain knowledge of search techniques.
3. To understand knowledge representation and planning.
4. To understand fundamentals of Neural Networks and architecture
5. To design and implement associative memory networks for solving practical problems.
6. To design and develop fuzzy controller systems for intelligent decision-making applications.

Course Outcomes: Students should be able to -

1. Identify the various characteristics of Artificial Intelligence and Soft Computing techniques.
2. Choose an appropriate problem solving method for an agent to find a sequence of actions to reach the goal state
3. Apply knowledge representation for reasoning

DRAFT

4. Construct supervised and unsupervised ANN for real world applications.
5. Design an associative memory network for real world problems
6. Design fuzzy controller system.

Prerequisite: AOA, Data structures
DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
0	Prerequisite	Mathematics, Analysis of Algorithms, Data Structures		
I	**Introduction to Artificial Intelligence and Soft Computing**	1.1 Definition of Artificial Intelligence, History of AI Intelligent Agents : Agents and Environments , Rationality, Nature of Environment, Structure of Agent, types of Agent, PEAS Properties 1.2 Soft Computing: Introduction of soft computing, Randomness, Vagueness, Approximation, Uncertainty, soft computing vs. hard computing, various types of soft computing techniques. Applications of AI, Applications of Soft Computing. **Self-learning Topics:** Case studies of intelligent agents used in real-world systems (chatbots, recommendation systems, autonomous systems).	**4**	**CO 1**
II	**Problem Solving and Search Techniques**	2.1 Problem Formulation 2.2 Uninformed Search Methods: Depth Limited Search, Depth First Iterative Deepening (DFID) Informed Search Method: Hill climbing Search, A* Search 2.3 Constraint Satisfaction Problem Solving, Water Jug Problem 2.4 Zero-Sum Game: Min-Max Algorithm, Alpha-Beta Pruning **Self-learning Topics:** Memory Bound Algorithm: SMA, SMA*	**8**	**CO 2**
III	**Knowledge Representation, Reasoning and Planning**	3.1 Knowledge based agents 3.2 First order logic: syntax and Semantic Knowledge Engineering in FOL Inference in FOL : Unification, Forward Chaining, Backward Chaining and Resolution 3.3 Partial Order Planning, Hierarchical Planning and Conditional Planning **Self-learning Topics:** Bayesian Belief Network, Types of Planning	**6**	**CO 3**
IV	**Artificial Neural Network**	4.1 Introduction – Biological Neural System, Elements of Artificial Neural Networks, McCulloch-Pitts Neuron, Activation Functions: Unipolar and Bipolar Functions 4.2 Learning Rules: Hebbian Learning, Perceptron, Delta learning rule, Competitive Learning 4.3 Neural Network Architecture: Single layer Feed	**10**	**CO 4**

DRAFT

		Forward ANN, Multilayer Feed Forward ANN, Back Propagation algorithm. 4.4 Un-Supervised Learning algorithm: Kohonen Self Organizing Maps, Learning VQ **Self-learning Topics:** Feedback Neural Network, Recurrent Neural Network, Radial Basis Network		
V	**Associative Memory Network**	5.1 Introduction to Associative Memory, 5.2 Auto associative and Hetero associative Memory Network, 5.3 Bidirectional Associative Memory (BAM), 5.4 Hopfield network. **Self-learning Topics:** Learning rules used in associative memory networks (Hebbian learning, correlation learning).	**4**	**CO 5**
VI	**Fuzzy System**	6.1 Introduction to Fuzzy Sets: Fuzzy set theory, Fuzzy set versus crisp set, fuzzy relations, membership functions: Triangular, Trapezoid, Sigmoid, Gaussian 6.2 Fuzzy Logic: Fuzzy Logic basics, Fuzzy Rules and Fuzzy Reasoning 6.3 Fuzzy inference systems: Fuzzification Methods, Defuzzification Methods: Centre of Gravity, Mean-of-Max(MoM), Smallest of Max, Largest of Max 6.4 Mamdani Model for fuzzy controllers **Self-learning Topics:** Takagi-Sugeno Model, ANFIS,CANFIS, Genetic algorithms	**7**	**CO 6**

Text Books:

1. Stuart J. Russell and Peter Norvig, "Artificial Intelligence: A Modern Approach —Second Edition" Pearson Education.
2. S.N.Sivanandam, S.N.Deepa "Principles of Soft Computing" Second Edition, Wiley Publication.
3. Zimmermann H.S "Fuzzy Set Theory and its Applications" Kluwer Academic Publishers.
4. S.Rajasekaran and G.A.Vijayalakshmi Pai "Neural Networks, Fuzzy Logic and Genetic Algorithms" PHI Learning.
5. N.P.Padhy, ―Artificial Intelligence and Intelligent Systems‖, Oxford University Press.
6. Samir Roy and Chakraborty, ―Introduction to soft computing , Pearson Education.

Reference Books:

1. Elaine Rich and Kevin Knight — Artificial Intelligence Third Edition, Tata McGraw-Hill Education Pvt. Ltd., 2008.
2. Satish Kumar "Neural Networks A Classroom Approach" Tata McGrawHill.
3. Hagan, Demuth, Beale,"Neural Network Design" CENGAGE Learning, India Edition.
4. J.-S.R.Jang "Neuro-Fuzzy and Soft Computing" PHI 2003.
5. JacekM.Zurada "Introduction to Artificial Neural Sytems" Jaico Publishing House.
6. Fuzzy Logic in Engineering Applications, Timothy

Online References:

DRAFT

Sr. No.	Website Name
1.	An Introduction to Artificial Intelligence - Course https://share.google/umcQLMykKscKcgb5k
2.	Artificial Intelligence: Concepts and Techniques - Course https://share.google/8JeKlR2eOnVXNdBVt
3.	Introduction To Soft Computing - Course https://share.google/Gyex84j4McZQQw3zI
4.	Introduction To Soft Computing – NPTEL+ https://share.google/FWefkZDqu5BIHkiCn

Assessment:

Internal Assessment (IA) for 20 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

➢ **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.

University of Mumbai

Course Code: _____and Course Name: .

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

NB:
(5) Question 1 is compulsory.
(6) 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

DRAFT

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2115114	Data Warehousing and Mining	3	-	-	3	-	-	3

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.

DRAFT

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					
2115114	Data Warehousing and Mining	20	20	40	60	2	--	--	100

Rationale:
The subject Data Warehousing and Mining enables students to understand how large volumes of data can be systematically stored, managed, and analyzed to extract useful insights. It focuses on data warehouse design, OLAP operations, and data mining techniques for classification, clustering, and association analysis. This knowledge builds a foundation for data-driven decision-making and advanced analytics in real-world applications.

Course Objectives: Six Course Objectives

1. To introduce the fundamental concepts of data warehousing and data mining.

2. To familiarize students with data warehouse architecture, modeling techniques, and OLAP operations.
3. To develop understanding of various data preprocessing and data exploration techniques for knowledge discovery.
4. To impart knowledge on classification, clustering, and association rule mining algorithms for data analysis.
5. To enable students to apply appropriate data mining techniques for solving real-world problems.
6. To provide exposure to advance areas such as web, text, and social media mining.

Course Outcomes: Six Course outcomes (Based on Blooms Taxonomy)

1. Understand data warehouse fundamentals and design data warehouse with dimensional modelling and apply OLAP operations.
2. Apply data preprocessing and exploration techniques to prepare raw data for mining.
3. To implement and evaluate a classification algorithm for real-world case study
4. To implement and interpret results clustering algorithm for real-world case study
5. To understand and apply concept of association rule mining and market basket analysis.
6. Apply basic concepts of text, graph, and social media mining to extract information.

Prerequisite: Databases, Programming (especially languages like SQL, Python, or R), Statistics

DETAILED SYLLABUS: total six module for each subject (13 Weeks)

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
0	Prerequisite	Databases, Programming, Statistics		
I	**Introduction Data Warehouse**	Introduction to Data Warehouse, Data warehouse architecture, Data warehouse versus Data Marts, E-R Modeling versus Dimensional Modeling, Data Warehouse Schemas; Star Schema, Snowflake Schema, Factless Fact Table, Fact Constellation Schema. Update to the dimension tables. Major steps in ETL process, OLTP versus OLAP, OLAP operations: Slice, Dice, Rollup, Drilldown and Pivot. **Self-learning Topics:** *Relating cloud computing with data warehousing, real-time ETL, Introduction to Big Data*	**8**	**CO1**
II	**Data Mining, Exploration & Preprocessing**	Data Mining Task Primitives, Architecture, KDD process, Issues in Data Mining. Data Exploration: Types of Attributes, Statistical Description of Data, Data Visualization, Data Preprocessing: Descriptive data summarization, Cleaning, Integration & transformation, Data reduction, Data Discretization and Concept hierarchy generation. Applications of Data Mining, **Self-learning Topics:** Data Cleaning and Data normalization methods	**8**	**CO2**

DRAFT

III	**Classification**	Basic Concepts, ID3 algorithm, Naïve Bayesian Classification, Performance evaluation of classification models: Confusion matrix, precision. recall, F1score, specificity, AUC-ROC curve. Evaluating the Accuracy of a Classifier: Holdout & Random Subsampling, Cross Validation, Bootstrap. **Self-learning Topics:** Ensemble Models	**6**	**CO3**
IV	**Clustering**	Types of data in Cluster analysis, Partitioning Methods (k-Means, k-Medoids), Hierarchical Methods (Agglomerative, Divisive), Density based clustering: DBSCAN **Self-learning Topics:** Advanced clustering algorithms BIRCH, CLIQUE	**6**	**CO4**
V	**Rule Mining**	Market Basket Analysis, Frequent Item sets, Association Rule, Apriori Algorithm, Improving the Efficiency of Apriori, FP-Tree, FP-Growth algorithm, Introduction to Mining Multilevel Association Rules and Mining Multidimensional Association Rules. **Self-learning Topics: C**onstraint-based mining	**6**	**CO5**
VI	**Types of Mining**	Descriptive and Predictive Mining, Web Mining: Web content, Web Structure, Web Usage, Introduction to Graph Mining, Introduction to Text Mining, Introduction to Social Media Mining, Mining Time Series Data **Self-learning Topics:** Page Rank, HITS algorithm	**5**	**CO6**

DRAFT

Text Books:

1. Paulraj Ponniah, "Data Warehousing: Fundamentals for IT Professionals", Wiley India.
2. Han, Kamber, "Data Mining Concepts and Techniques", Morgan Kaufmann 3rd edition.
3. M.H. Dunham, "Data Mining Introductory and Advanced Topics", Pearson Education.

References:

1. Reema Theraja, "Data warehousing", Oxford University Press 2009.
2. Pang-Ning Tan, Michael Steinbach and Vipin Kumar, "Introduction to Data Mining", Pearson Publisher 2nd edition.
3. Ian H. Witten, Eibe Frank and Mark A. Hall, "Data Mining", Morgan Kaufmann 3rd edition.

Online References:

Sr. No.	Website Name
1.	https://hanj.cs.illinois.edu/bk3/bk3_slidesindex.htm

2.	https://onlinecourses.swayam2.ac.in/imb25_mg200/preview
3.	https://onlinecourses.nptel.ac.in/noc21_cs06/preview

Assessment:

Internal Assessment (IA) for 20 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

➢ **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

DRAFT

University of Mumbai

Course Code: _____and Course Name: .

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

NB: (7) Question 1 is compulsory. (8) 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.

DRAFT

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2115115	**Computer Graphics**	**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					

2115115	**Computer Graphics**	**20**	**20**	**40**	**60**	**2**	**--**	**--**	**100**

Rationale: This course introduces the fundamental concepts and techniques of computer g raphics. It focuses on 2D and 3D transformations, viewing, clipping, and rendering methods, enabling students to design and implement interactive graphical applications.

Prerequisites: Fundamentals of Mathematics and Programming Logic and Language.

Course Objectives:

1. Introduce the basic components of computer graphics systems and OpenGL fundamentals.
2. Develop the ability to draw and fill basic 2D graphical shapes using standard algorithms.
3. Apply 2D transformations, viewing, and clipping techniques.
4. Build understanding of 3D transformations, projections, and curve generation methods.
5. Implement and assess visible surface detection techniques and animation principles for graphical scenes.
6. Introduce shading models and AR/VR technologies, hardware, and software environments.

Course Outcomes:

Upon completion of the course, the learners will be able to:

1. Describe the structure and components of a computer graphics system and the use of OpenGL for drawing primitives.
2. Apply line, circle, and polygon filling algorithms to generate and fill 2D graphical shapes
3. Apply 2D geometric transformations and utilize viewing and clipping techniques for effective graphical representation
4. Apply 3D transformations and projection techniques on graphical objects and construct Bezier curves for shape representation.
5. Demonstrate visible surface detection and animation techniques in Computer Graphics
6. Apply basic shading techniques and recognize AR/VR hardware and software components.

Detailed Contents:

Sr. No.	Name of Module	Detailed Content	Hrs	CO
0	Prerequisite	Knowledge of C Programming and Basic Mathematics.		
I	**Introduction and Overview of Graphics System**	1.1 Representative uses of computer graphics 1.2 Graphics Hardware and Software 1.3 Overview of coordinate representation, Rasterization 1.4 Raster scan displays, Random scan displays 1.5 Architecture of Graphics rendering pipeline 1.6 Introduction to OpenGL, Libraries 1.7 OpenGl Basic Graphics Primitives	**6**	**CO1**
		Self-learning Topics: Display devices like Plasma Display, 3D Display, LED Display.		

DRAFT

II	**Output Primitives**	2.1 Introduction to Scan Conversion 2.2 Line Drawing Algorithms: DDA and Bresenham (Mathematical derivation and numerical) 2.3 Midpoint Circle Drawing Algorithm (Mathematical derivation and numerical) 2.4 Filled Area Primitives: Boundary Fill and Flood Fill 2.5 Polygon Representation and Edge Tables 2.6 Scan-line Polygon Fill Algorithm	**7**	**CO2**
		Self-learning Topics: Midpoint Ellipse Algorithm, Aliasing, Antialiasing techniques		
III	**2D Geometry: Transformations and Clipping**	3.1 Basic transformations: Translation, Scaling, Rotation Matrix representation 3.2 Homogeneous Coordinates, Composite transformation, 3.3 Advanced transformations: Reflection and Shear 3.4 Window to Viewport coordinate transformation 3.5 Clipping operations: Point clipping, Cohen-Sutherland Line clipping algorithm 3.6 Polygon Clipping Algorithms: Sutherland- Hodgeman Algorithm	**7**	**CO3**
		Self-learning Topics: Liang Barsky Line clipping algorithm, Weiler-Atherton Polygon Clipping		
IV	**3D Geometry: Transformations and Modeling**	4.1 3D Object Transformation 4.2 3D Viewing pipeline 4.3 Projections: Parallel, Perspective 4.4 Curve Generation: Bezier Curve 4.5 Introduction to Solid Modeling 4.6 Different representations of Solid Modeling	**6**	**CO4**
		Self-learning Topics: B-Spline Curve, Fractals		
V	**Visible Surface Detection, Animation**	5.1 Visible Surface Detection: Classification of Visible Surface Detection algorithm 5.2 Back Surface detection method, 5.3 Depth Buffer method 5.4 Animation: Introduction to Animation, Traditional Animation Techniques, 5.5 Key framing 5.6 Motion capture	**6**	**CO5**
		Self-learning Topics: Advanced Animation Concepts and Popular tools		
VI	**Rendering and Immersive Graphics Technologies**	6.1 Introduction to Rendering 6.2 Illumination models: Diffuse and specular reflection 6.3 Determining Surface normal and Reflection vector 6.4 Shading Algorithms: Gourad shading, Phong shading 6.5 Technologies Enabling Immersive Graphics: Equipment and System Overview of ARVR systems 6.6 Physical elements 6.7 APIs and Hardware	**7**	**CO6**
		Self-learning Topics: Ideas of Advanced Rendering Techniques		

DRAFT

Text Books:	1. Hearn & Baker, "Computer Graphics C version", 2nd Edition, Pearson Publication 2. James D. Foley, Andries van Dam, Steven K Feiner, John F. Hughes, "Computer Graphics Principles and Practice in C", 2ndEdition, Pearson Publication 3. Samit Bhattacharya, "Computer Graphics", Oxford publication **4.** Akenine-Möller, T., Haines, E., Hoffman, N., Pesce, A., Iwanicki, M., & Hillaire, S. (2018). *Real-Time Rendering (*4th ed.). CRC Press.
References Books :	1. 1.D. Rogers, "Procedural Elements for Computer Graphics", Tata McGraw-Hill Publications. 2. Zhigang Xiang, Roy Plastock, "Computer Graphics", Schaum"s Outlines McGraw-Hill Education 3. Rajesh K. Maurya, "Computer Graphics", Wiley India Publication. 4. F. S. Hill, "Computer Graphics using OpenGL", Third edition, Pearson Publications.
Online References:	https://onlinecourses.nptel.ac.in/noc22_cs111/preview https://nptel.ac.in/courses/106102065 https://nptel.ac.in/courses/106106090
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 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.

DRAFT

University of Mumbai

Course Code: _____and Course Name: .

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

NB: (9) Question 1 is compulsory. (10) 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.

DRAFT

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2115116	Internet Programming	3	-	-	3	-	-	3

		Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-I	IAT-II	IAT-I + IAT-II					
2115116	Internet Programming	20	20	40	60	2	--	--	100

Course Objectives:

1. To understand the fundamentals of internet architecture, protocols and version control using Git and GitHub.
2. Design and implement client-side applications using TypeScript
3. To learn and demonstrate a complete system using key AngularJS & Angular concepts including data binding, controllers, filters, and directives.
4. Utilize MongoDB for data storage and backend integration.
5. To Demonstrate XML and web extension techniques for data exchange.
6. To build and deploy lightweight web applications using Flask.

Course Outcomes:

1. Understand the fundamentals of internet protocols, HTTP communication and utilize Git and Github for version control and collaboration.
2. Design and develop client-side applications using TypeScript.
3. Apply AngularJS and Angular features to develop dynamic web interfaces.
4. Integrate MongoDB with web applications for CRUD operations.
5. Implement XML-based data exchange between frontend and backend system.
6. Develop and deploy web applications using Flask framework.

Prerequisite: Computer network, Java Programming

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
0	Prerequisite	Basic knowledge of computers, networking, and programming logic		
I	**Introduction to Internet and web technology**	Internet fundamentals, web architecture and protocols. -HTTP request and response messages, web clients and web servers	**6**	**1**

DRAFT

		-HTML: Introduction, history, versions and basic elements -HTML5: headings, paragraphs, colors, fonts, links, lists, tables, images, forms -Difference between HTML and HTML 5 -Git &GitHub: Introduction, pushing source code, pull requests, collaboration. **Self-learning Topics:** Create a personal "About Me" webpage using HTML5		
II	**Type script**	Architecture, Data types, variables and Operators, Decision Making and Loops, Functions Object oriented Programming: Classes and objects, Constructors and access modifiers, Inheritance and method overriding, Interfaces and abstract classes Advanced Typescript Concepts: Generics in functions and classes, Union and intersection types, Type aliases and utility types, Type inference Type Script Modules DOM Manipulation and API Integration **Self-learning Topics:** Difference between Javascript and Typescript,	**6**	**2**
III	**Angular JS**	Front End Framework Angular JS-framework using Angular CLI and NPM, CDN Link in script tag, Architecture, Expressions, Creation of modules, Data Binding, Controllers, Scope, Services, AJAX-$http, Tables, Event, Form Creation and Validation, API, Directives, Filters, Animation, Applications, Features. Angular-Templates, Components, Data Binding, Directives, events, Conditional Rendering, List, Forms, Router, Services & Dependency Injection, Pipes, Lifecycle Hooks, Styling. Difference between Angular Js and Angular. Self-learning Topics: Installation of Angular JS using Angular CLI, Concepts of JavaScript, JSON and Bootstrap. **Self-learning Topics:** Hands-on creation of reusable Angular components	**7**	**3**
IV	**Backend: Mongo DB**	Introduction of MongoDB, Why NoSQL? Difference between NoSQL and MongoDB, Data Types, Administering User Accounts, Configuring Access Control Getting Started with MongoDB and Node.js -Adding the MongoDB Driver to Node.js, Connecting to MongoDB from Node.js, Accessing and Manipulating Databases, Manipulating MongoDB Documents from Node.js, Accessing MongoDB from Node.js. **Self-learning Topics:** Working on NoSQL databases.	**7**	**4**

DRAFT

V	**Web Extensions – XML Integration using AngularJS and MongoDB**	Introduction to XML and XML Schema, Linking XML with DTD and Validation of XML document using DTD. XML DOM Parser, Transformation of XML with HTML using XSLT, $http service methods –get () and post (), Understanding flow of XML data from AngularJS to MongoDB, XML in REST web services. **Self-learning Topics:** XML with SAX Parser and SOAP web Service	**7**	**5**
VI	**Python-based Web Framework: Flask**	Introduction to Flask & Environment Setup, Flask Application Structure & Routing, Flask HTTP Methods & Request Object, Templates and Jinja2, Cookies, Sessions, and Flash Messages, File Uploading in Flask **Self-learning Topics:** Flask Vs Django	**6**	**6**

Textbooks:

1.Paul Deitel, Harvey Deitel, Abbey Deitel, "Internet &World Wide How to Program", Pearson Education,5e

2.Miguel Grinberg, "Flask Web Development: Developing Web Applications with Python", O'Reilly

3. Boris Cherny, "Programming TypeScript- Making Your Javascript Application Scale", O'Reilly Media Inc

4.Simon Holmes Clive Harber, "Getting MEAN with Mongo, Express, Angular, and Node", Manning Publications.

Reference Books:

1. Learning Web Design: A Beginner's Guide to HTML,CSS,JavaScript and Web Graphics by Jennifer Robbins (O'Reillt Media,5e)

2.Yakov Fain and Anton Moiseev, "TypeScript Quickly", Manning Publications.

3. Steve Fenton, "Pro TypeScript: Application - Scale Javascript Development", Apress

4. Brad Dayley, Brendan Dayley, Caleb Dayley, "Node.js, MongoDB and Angular Web Development: The definitive guide to using the MEAN stack to build web applications", 2nd Edition, Addison Wesley Professional

5. Bruno Joseph D'mello Mithun Satheesh Jason Krol, "Web Development with MongoDB and Node" Third Edition, Packt Publishing

Online References:

Sr. No.	Website Name
1.	https://www.typescriptlang.org/docs/
2.	https://nptel.ac.in/courses/106/106/106106242/
3.	https://git-scm.com/docs

Assessment:

Internal Assessment (IA) for 20 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

- **Question paper format**

  - Question Paper will comprise of a total of **six questions each carrying 15 marksQ.1** will be **compulsory** and should **cover maximum contents of the syllabus**

DRAFT

- **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: (11) Question 1 is compulsory. (12) 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.

DRAFT

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2115117	Software Engineering Lab	-	2	-	-	1	-	1

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Practical & Oral	Total
		Internal assessment			End Sem. Exam			
		IAT-	IAT-II	IAT-I + IAT-II				
2115117	Software Engineering Lab	--	--	--	--	25	25	50

Lab Objectives:

1. To select case studies to solve real life problems by applying software engineering principles
2. To impart state-of-the-art knowledge on Software Engineering and UM.L
3. To create test cases for selected case study using white box testing.
4. To develop skills in software modeling, documentation, and project planning using CASE tools.
5. To apply software estimation, risk management, and configuration management techniques in project development.
6. To implement validation, verification, and user acceptance testing practices for software quality assurance.

Lab Outcomes:

1. Identify requirements and apply process model to selected case study.
2. Prepare SRS in IEEE format for selected case study.
3. Analyze and design models for the selected case study using UML modeling.

DRAFT

4. Create and analyze test cases for application using white box testing.
5. Prepare accurate documentation for experiments performed.
6. Apply ethical principles like timeliness and adhere to the rules of the laboratory.

Prerequisite: Object Oriented Programming with Java , Python Programming
Suggested List of Experiments.
Assign the case study/project, including a detailed explanation of the problem, to a group of two or three students. Laboratory work shall adhere to the course syllabus, with a minimum of 10 experiments. Open-source computer-aided software engineering (CASE) technologies has the capability to utilize for performing the experiment.

Sr No	List of Experiments	Hrs
01	To Study software process models & DSP for selected case study.	2
02	To Prepare SRS in IEEE format for any case study.	2
03	Use project management tool to schedule project plan for a case study.	2
04	To Study software estimation techniques and calculate the cost for case study.	2
05	Draw Class diagram and use case diagram using any opensource tool.	2
06	Draw DFD for Case study using any opensource Tool	2
07	Develop Activity / State Transition diagram for the project using any opensource tool.	2
08	Develop RMMM plan for the project	2
09	Change specifications and make different versions of case study using any SCM tool	2
10	Develop test cases for the project using white box testing.	2
11	To create Prototypes, User acceptance testing for selected case study	2
12	To Explore agile methodology tools(Jira & Trello)	2

DRAFT

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
2115118	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				
2115118	Computer Network Lab	--	--	--	--	25	25	50

Lab Objectives:
1. To learn various simulator to implement and analyze the network in it.
2. To learn various TCP/IP utility command.
3. To learn different network topologies using appropriate routing algorithms.
4. To learn different protocols in transport layer stack
5. To implement basic terminal security commands on GNS3 to secure network devices and analyze captured network packets
6. To implement basic network security configurations and analyze captured packets to identify potential threats.

Lab Outcomes:
1. Compare Various network simulator.
2. Analyze TCP and UDP data traffic for network performance evaluation.
3. Design small network topologies using appropriate routing algorithms.
4. Demonstration of transport layer protocols on Wireshark/NS2
5. Develop a program for Client-Server architecture using socket programming
6. Identify security threats by analyzing captured network packets.

Prerequisite: Programming Languages - Java/C/C++/Python
Suggested List of Experiments:

DRAFT

Sr No	List of Experiments	Hrs
01	Comparative Analysis of Network Simulators.	02
02	Implement and analyze TCP/IP utility commands.	02
03	TCP and UDP Data Traffic GenerationusingNS2.	02
04	Configure network devices, such as hubs and switches within a network topology using Packet Tracer software.	02
05	Design and implement Static routing using GNS3 software.	02
06	Design and implement Dynamic routing using GNS3 software.	02
07	Write a program using TCP socket for wired network for following (any one) a. Write Hello to Each other b. Create sample Chat system c. Perform Calculator or compute and send back the factorial of a given number	02
08	Write a program using UDP Sockets to enable file transfer between two machines.	02
09	Capture packets using Wireshark, write the exact packet capture filter expressions to accomplish the following and save the output in file: a) Capture all TCP traffic to/from Facebook/instagram/youtube, during the time when you log in to your account b) Write a DISPLAY filter expression to count all TCP packets (captured under item #1) that have the flags SYN, PSH, and RST set. Show the fraction of packets that had each flag set. c) Count how many TCP packets you received from/sent to Facebook/instagram/youtube and how many of each were also HTTP packets.	04
10	To implement and analyze DHCP server and client.	02
11	Implement Basic terminal security command on GNS.	02
12	Analyze captured packets to identify potential security issues such as unauthorized access, clear-text credentials, or malicious activity.	02

Suggested List of Assignments:

Sr No	List of Assignments / Tutorials	Hrs
	Assignment no 1	
01	Explain the functions of each layer of OSI along with the diagram.	1
02	Explain Ethernet devices along with their function's advantages and disadvantages.	1
03	Explain CSMA/CD and CSMA/CA along with flow diagram.	2
04	What is sliding window technique, Explain it with an example.	2
05	An ISP grants 200.192.30.0/24 block of IP addresses. It is required to have at least 6 subnets and 30 host addresses per subnet. Perform subnetting and determine : 1) subnet mask 2) slash notation 3) broadcast address in first subnet.	3
	Assignment no 2	
06	Compare Ipv4 and Ipv6. Explain Ipv4 packet header.	1
07	Explain TCP three way handshaking during connection establishment and connection release with neat diagrams.	4
08	Explain the purpose of encryption and authentication in SSH communication.	2
09	Define the main components of 5G network architecture and describe their roles.	1
10	Demonstrate how phishing attacks exploit human vulnerabilities to gain	3

DRAFT

	unauthorized access to sensitive information.	

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. CCNP 1 : Advanced Routing, 2nd Edition, Pearson Education.
4. R. Howard, *Cyber Security Essentials*, 1st ed. Boca Raton, FL: CRC Press, 2010.

References:

1. Computer Networking: A Top-Down Approach (7th Edition) by James F. Kurose and Keith W. Ross
2. TCP/IP Illustrated, Volume 1: The Protocols by W. Richard Stevens
3. Data and Computer Communications by William Stallings

Online Resources:

Sr. No.	Website Name
1.	https://www.geeksforgeeks.org/computer-networks/transport-layer-protocols/
2.	https://en.wikipedia.org/wiki/Transport_layer
3.	https://www.sysnettechsolutions.com/en/configure-cisco-router-gns3/

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.

DRAFT

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2115119	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-	IAT-II	IAT-I + IAT-II				
2115119	Artificial Intelligence & Soft Computing Lab	--	--	--	--	25	25	50

Lab Objectives:

1. Problem formulation for real world case study
2. To analyse and apply different search techniques for problem solving in AI
3. To design and implement various types of neural network architectures for learning and pattern recognition.
4. To design associative memory network for given application
5. To understand and apply fuzzy logic for developing a fuzzy controller.
6. To study and explore different AI tools

DRAFT

Lab Outcomes:

1. To articulate the design of AI agents in terms of PEAS properties
2. To apply informed and uninformed search techniques to solve real-life problem statements
3. To design and train feedforward neural network using different learning rules
4. To design a fuzzy controller for real word application
5. To differentiate and apply evolutionary algorithm for given application
6. To explore modern AI tools for development of AI agents

List of Experiments.

Sr No	List of Assignments / Tutorials (Any TWO)	Hrs
01	Prepare a case study on AI Agents including, type, environment and PEAS properties	2
02	Implement any one informed search techniques for problem statements such as N Queen, 15- Puzzle problem, Water Jug Problem, Travelling Salesman	2
03	Implement any one uninformed search techniques for problem statements such as N Queen, 15- Puzzle problem, Water Jug Problem, Travelling Salesman	2
04	To implement Alpha-Beta Pruning	2
05	Build a SLP- NN architecture using any one learning rule	2
06	To implement the Supervised Learning algorithm.[MLP]	2
07	To implement the Unsupervised Learning algorithm	2
08	To implement Associative Memory Network for a given problem statement	2
09	To implement fuzzy membership functions.	2
10	Write a program to simulate a fuzzy control system.	2
11	Implement Genetic algorithm/Simulated annealing	2
12	To create AI agent for automation using tools like n8n	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.

DRAFT

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2115120	Data Warehousing and Mining 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				
2115120	Data Warehousing and Mining Lab	--	--	--	--	25	--	25

Lab Objectives: Six Lab Objectives

1. To familiarize students with the design and implementation of data warehouses and data marts using dimensional modeling.
2. To enable students to perform ETL operations and OLAP queries for multidimensional data analysis.
3. To provide practical exposure to data preprocessing and exploration techniques for data quality improvement.
4. To implement and analyze classification, clustering, and association rule mining algorithms.
5. To introduce the use of data mining and visualization tools such as WEKA, R, Power BI, and Tableau.
6. To encourage students to apply data mining techniques on real-world datasets for knowledge discovery and decision-making.

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

1. Design and construct a data warehouse schema using dimensional modeling concepts.

DRAFT

2. Perform OLAP operations such as Slice, Dice, Roll-up, Drill-down, and Pivot using SQL queries.
3. Implement data preprocessing techniques for data cleaning, transformation, and visualization
4. Develop and evaluate classification, clustering, and association rule mining algorithms.
5. Use data mining and visualization tools (WEKA, R, Power BI, Tableau) to extract and interpret useful patterns from datasets.
6. Use web scraping tools extract structured data from web pages

Prerequisite: Databases, Programming (especially languages like SQL, Python, or R), Statistics

List of Experiments.

Sr No	List of Experiments	Hrs
01	Select a case study on building Data warehouse/Data Mart Write Detailed Problem statement and design dimensional modelling (creation of star and snowflake schema)	2
02	Implementation of all dimension table and fact table based on experiment 1 case study	2
03	Implementation of OLAP operations (in SQL) : Slice, Dice, Rollup, Drilldown and Pivot based on experiment 1 case study	2
04	Data Preprocessing & Exploration: Implementation of Descriptive analysis, Null Values handling, Data Normalization, Data Discretization	2
05	Data Preprocessing & Exploration : Data Transformation, Outlier analysis, Visualization	2
05	Implementation of Naïve Bayesian/ Classification Tree algorithm	2
06	Implementation of Clustering algorithm (K-means/K-medoids)	2
07	Implementation of any one Hierarchical Clustering method	2
08	Implementation of Association Rule Mining algorithm (Apriori)	2
09	Perform data Pre-processing task and demonstrate Classification, Clustering, Association algorithm on data sets using data mining tool (WEKA/R tool)	2
10	Perform Data exploration, Data Visualization using PowerBI/Tableau	2
11	Perform web scraping using tools like Scarpy, Beautiful Soup etc	2

Sr No	List of Assignments / Tutorials	Hrs
01	1. Describe KDD process in detail with a suitable diagram 2. Explain Outlier Analysis in detail	2
02	1. Explain types of web mining with suitable examples. 2. Explain web crawlers and web scraping in detail.	2

Text Books:

1. Paulraj Ponniah, “Data Warehousing: Fundamentals for IT Professionals”, Wiley India.
2. Han, Kamber, “Data Mining Concepts and Techniques”, Morgan Kaufmann 3rd edition.
3. M.H. Dunham, “Data Mining Introductory and Advanced Topics”, Pearson Education.

References:

DRAFT

1. Reema Theraja, "Data warehousing", Oxford University Press 2009.
2. Pang-Ning Tan, Michael Steinbach and Vipin Kumar, "Introduction to Data Mining", Pearson Publisher 2nd edition.
3. Ian H. Witten, Eibe Frank and Mark A. Hall, "Data Mining", Morgan Kaufmann 3rd edition.

Online Resources:

Sr. No.	Website Name
1.	https://hanj.cs.illinois.edu/bk3/bk3_slidesindex.htm
2.	https://onlinecourses.swayam2.ac.in/imb25_mg200/preview
3.	https://onlinecourses.nptel.ac.in/noc21_cs06/preview

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
2115121	Computer Graphics 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					
2115121	Computer Graphics Lab	--	--	--	--	--	25	--	25

Prerequisites:	C Programming Language.
Lab Objectives:	1. Understand the need of computer graphics. 2. Learn the algorithms to draw line, circle and other graphics primitives. 3. Learn 2D transformation, viewing, clipping techniques. 4. Learn 3D transformation, projection, curve generation methods. 5. Learn visible surface detection techniques. 6. Learn shading,animation techniques and install ARVR software.

DRAFT

Lab Outcomes:	Upon completion of the course, the learners will be able to: 1. Use basic graphics functions, modes, and standards. 2. Implement various primitive algorithms like line,circle,polygon,etc. 3. Apply transformation,projection,and clipping algorithms on 2D graphical objects 4. Apply transformation algorithms on 3D graphical objects and curve generation methods. 5. Perform visible surface detection techniques. 6. Apply shading algorithms and develop a graphical application/Animation based on learned concepts.

Declaration:
The Subject In-charges of all affiliated colleges are requested to conduct any seven (7) experiments from the **LO1 to LO5**, but ensure at least one experiment of each LO. Mini project is compulsory, ensuring that a minimum of ten (10) experiments except mini projects are completed as part of the curriculum.
The following experiments serve as samples to illustrate the application of concepts covered in each unit. Instructors are encouraged to modify and adapt these experiments to meet the specific needs of the course and the learning objectives. It is essential to ensure that the fundamental concepts and skills outlined in each unit are adequately covered, even with modifications.

List of Experiments.

Sr. No.	Experiment Details/Description	Hrs	LO
1	Write a C program to draw moving cycle using various graphics objects (Pixel, Circle, Line, Ellipse, Rectangle, Triangle, Polygon) using graphics functions	2	LO1
2	Implement DDA Line Drawing algorithm (dotted/dashed/thick) and Bradenham's Line algorithm(dotted/dashed/thick) analyze both the algorithm in terms of time, space and accuracy. **OR** Implement any line Drawing algorithm to draw shape like square, rectangle, house etc.	2	LO2
3	Implement midpoint Circle algorithm and Bradenham's circle algorithm analyze both the algorithm in terms of time, space and accuracy. **OR** Implement any circle Drawing algorithm to show slideshow of emojis like smiley,angry,sad etc.	2	LO2
4	Implement Area Filling Algorithm: Boundary Fill, Flood Fill analyze both the algorithm in terms of time, space and accuracy.	2	LO2
5	Implement Scan line Polygon Filling algorithm.	2	LO2
6	Implement 2D Transformations: Translation, Scaling, Rotation, Reflection, Shear for moving car	2	LO3

DRAFT

7	Implement Line Clipping Algorithm: Cohen Sutherland / Liang Barsky for Road boundary in viewport.	2	LO3
8	Implement polygon clipping algorithm (at least one).	2	LO3
9	Implement 3D Transformations: Translation, Scaling, Rotation, Reflection, Shear for 3D cube transformation/3D house transformation/3D car transformation/3D pyramid transformation.	2	LO4
10	Implement Curve: Bezier for n control points, B Spline (Uniform)(at least one).	2	LO4
11	Implement Fractal generation method (anyone) for Fractal Tree (animated growing tree)	2	LO4
12	Character Generation: Bit Map method and Stroke Method for text rendering machine	2	LO4
13	Perform projection of a 3D object on Projection Plane: Parallel and Perspective.	2	LO4
14	Implement Back-Face detection for simple 3D scenes OR Implement Depth (Z-) Buffer for simple 3D scenes. OR Implement Area-Subdivision methods for simple 3D scenes.	2	LO5
15	Implement any one shading algorithm **OR** Case study on how an object in AR/VR environments appear realistic through shading. For example**:** • Virtual objects respond to **light source changes** in real time. • Shadows and reflections are updated dynamically. **OR** Case study on how shading helps visualize **3D scans** like CT or MRI data. For example: • 3D rendering of organs using **volume rendering with shading**. • Surface shading highlights features like tumors or fractures.	2	LO6
16	Installation of ARVR system	2	LO6
17	Mini project such as Implement Snake Game 3D Solar System Simulation Interactive 3D Car Racing Track Animated Flag or Cloth Simulation	2	LO6

DRAFT

	3D Solar Panel / Wind Turbine Simulator Animated Fireworks Simulation. Life cycle of flower.		

Text Books:	1 Hearn & Baker, "Computer Graphics C version", 2nd Edition, Pearson Publication 2 James D. Foley, Andries van Dam, Steven K Feiner, John F. Hughes, "Computer Graphics Principles and Practice in C", 2 ndEdition, Pearson Publication 3 Samit Bhattacharya, "Computer Graphics", Oxford Publication 4 Edward Angel, Dave Shreiner, "Interactive Computer Graphics", 7th Edition ,Pearson Publication 5 Donald Hearn & M. Pauline Baker, *Computer Graphics with OpenGL* ,4th edition,Pearson publication
References Books :	1 D. Rogers, "Procedural Elements for Computer Graphics", Tata McGraw-Hill Publications. 2 Zhigang Xiang, Roy Plastock, "Computer Graphics", Schaum"s Outlines McGraw-Hill Education 3 Rajesh K. Maurya, "Computer Graphics", Wiley India Publication. 4 F. S. Hill, "Computer Graphics using OpenGL", Third edition, Pearson Publication
Online References:	1 https://www.classcentral.com/course/interactivegraphics-2067 2 https://swayam.gov.in/nd2_ntr20_ed15/preview 3 https://nptel.ac.in/courses/106/106/106106090/ 4 https://www.edx.org/course/computer-graphics-2 5 https://onlinecourses.nptel.ac.in/noc20_cs90/preview

Sr. No	List of Assignments / Tutorials
01	Based on the Modules-1, 2 and 3 of Computer Graphics.
02	Based on the Moudles-4, 5 and 6 of Computer Graphics.

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)

DRAFT

DRAFT

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2115122	Internet Programming 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				
2115122	Internet Programming Lab	--	--	--	--	25	--	25

Lab Objectives:

1. To understand the structure of an HTML5 document for webpage design.

2. To learn basic Git commands and how to collaborate using GitHub.
3. To develop web applications using TypeScript and modular programming concepts.
4. Apply MongoDB for data management and connectivity with front-end frameworks..
5. Implement data integration using XML/JSON between Angular and MongoDB
6. Design and deploy web applications using the Flask framework.

Lab Outcomes:

1. Design and develop responsive webpages using HTML5 and CSS.
2. To initialize and manage repositories using Git and GitHub for version control and colloboration
3. To design web applications using Typescript
4. To Build MongoDB GUI.
5. Implement XML/JSON-based data exchange between Angular and MongoDB.
6. Design and deploy Flask-based web applications with integrated frontend and backend.

Prerequisite: Internet fundamentals and web architecture protocols

List of Experiments.

Sr No	List of Experiments (At least 8 to 10)	Hrs
01	Design and create a "Personal Portfolio" Webpage using HTML5	02
02	Demonstrate Git & GitHub for Source Code Management	02
03	Designing a calculator using Typescript	02
04	Inheritance/Access Modifiers Example using Typescript	02
05	Design a To-Do List using Typescript	02
06	Connecting Node.js to MongoDB and Performing CRUD Operations	02
07	Creating and Accessing a Secured MongoDB Database with User Accounts	02
08	Design an XML Document and Validate Using DTD Schema	02
09	Design an XML Document and Show the Flow of Data Using $http GET and POST Methods (AngularJS)	02
10	Building REST Web Services with XML	02
11	Design a Student Registration Form using Flask.	02
12	Design a Weather App using Flask (static weather data or API integration).	02

Sr No	List of Assignments / Tutorials	Hrs
01	Assignment on HTML5 and Form design	
02	Assignment on node.js and MongoDB connectivity	
03	Assignment to develop web application using typescript	
04	Assignment to Design a RESTful Web Service with XML Response	

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)

DRAFT

DRAFT

IKS

DRAFT

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-	**10**	**CO2**

DRAFT

		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		
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 • Measurement for Time Distance and Weight • Measurement of Time- An Illustration from Purana	**10**	**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. 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**

DRAFT

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. 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.	**10**	**CO6**

DRAFT

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=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 Resources:

DRAFT

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-	04

DRAFT

	detection methods are used in digital systems for reliable data storage and transmission.	
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

DRAFT

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)

DRAFT

Sem VI

DRAFT

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2116111	Machine Learning	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					
2116111	Machine Learning	20	20	40	60	2	--	--	100

Course Objectives: Six Course Objectives

1. To introduce students to the basic concepts and techniques of Machine Learning
2. To gain in depth knowledge of Supervised Learning algorithms.
3. To adapt unsupervised learning algorithms for real world applications.
4. To gain understanding to build Machine Learning models using ensemble techniques.
5. To assess the relevance of probabilistic learning for real world applications.
6. To gain fundamental knowledge of Dimensionality reduction.

Course Outcomes: Six Course outcomes (Based on Blooms Taxonomy)

1. Explain the fundamental concepts and techniques of Machine Learning and differentiate between various learning paradigms.
2. Apply and evaluate supervised learning algorithms such as regression and classification models.
3. Apply unsupervised learning techniques to analyze and interpret real-world applications..
4. Design and build Machine Learning models using ensemble methods.
5. Analyze and evaluate probabilistic learning methods for prediction and clustering.
6. Apply dimensionality reduction techniques to various applications.

DETAILED SYLLABUS: total six module for each subject (13 Weeks)

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
0	**Prerequisite**	Engineering Mathematics, Probability and Statistics, Data Structures, Algorithms, DWM		
I	**Introduction to Machine Learning**	**Fundamentals:** Machine Learning, Types of Machine Learning, Issues in Machine Learning, Application of Machine Learning, Steps in developing a Machine Learning Application Training	**05**	

DRAFT

		Error, Generalization error, Overfitting, Underfitting, Bias-Variance trade-off **Self-learning Topics:** Preprocessing Techniques, Feature Engineering		
II	**Supervised Learning**	**Learning with Regression:** Linear Regression, Multivariate Linear Regression, Logistic Regression **Classification using Decision Tree**: Classification and Regression Trees using GINI Index **Support Vector Machine** Optimal decision boundary, Margins and support vectors, SVM as Constrained optimization problem, Quadratic Programming, SVM for linear and nonlinear classification, Kernel trick. **Performance Evaluation Measures:** Accuracy, Precision, Recall, F1-measure, ROC_AUC, specificity. **Self-learning Topics:** Classification using ID3	**10**	
III	**Unsupervised Learning**	**Clustering:** **Fundamentals**: Definition of clusters, motivation **Distance and similarity measures**: Manhattan, Euclidian, Cosine Similarity, Inter-cluster distance, Intra-cluster distance **Graph Based Clustering:** Minimum Spanning Tree **Centroid based Clustering:** K-Means **Density based Clustering:** DBSCAN, Advantages over K-Means **Performance Evaluation Measures:** Silhouette Coefficient, Davies Bouldin Index (DBI), Dunn Index (DI) **Self-learning Topics:** Hierarchical Clustering, Agglomerative Clustering	**06**	
IV	**Ensemble Learning**	**Ensembles:** Motivation for Ensembles, Ensemble learning and generalization, Base learners vs Strong Learners, Hard voting vs Soft voting, Weighted Averaging **Bootstrap aggregating (Bagging)**, Out of Bag Error Estimation, Random Forest **Stacking:** Meta Learning, Base level and Meta level models, Blending **Boosting:** Stumps, Adaboost Comparison of Bagging vs Boosting **Self-learning Topics:** Gradient boosting, XGBoost, Hyper-parameter Tuning	**06**	
V	**Probabilistic Learning**	**Fundamentals**: Probabilistic modeling, motivation, Deterministic vs Probabilistic modeling, Likelihood, Prior, Posterior, Maximum Likelihood Estimation (MLE) **Bayesian Belief Network (BBN):** Probabilistic Reasoning, Bayesian network, Directed Acyclic Graph (DAG) Structure of BBN, Inference using	**06**	

DRAFT

		BBN **Expectation Maximization (EM):** Motivation and Need, Definition of EM, EM and MLE, EM Algorithm **Self-learning Topics:** Bayesian Neural Network and its comparison with BBN, Applications: Probabilistic forecasting/ Clustering		
VI	**Dimensionality Reduction Techniques**	**Fundamentals**: Motivation, Curse of Dimensionality **Algorithms:** Principal Component Analysis(PCA), Singular Value Decomposition (SVD), Linear Discriminant Analysis (LDA) **Applications of PCA**: Image Compression/Image Classification **Applications of SVD**: Image Reconstruction **Applications of LDA**: Financial risk assessment/ Customer segmentation **Self-learning Topics:** Independent Component Analysis (ICA)	**06**	

Text Books:

1. Peter Harrington, "Machine Learning in Action", DreamTech Press
2. Ethem Alpaydın, "Introduction to Machine Learning", MIT Press
3. Tom M.Mitchell, "Machine Learning", McGraw Hill
4. Stephen Marshland, "Machine Learning: An Algorithmic Perspective", CRC Press.
5. Kevin P. Murphy, "Machine Learning — A Probabilistic Perspective"

References:

1. Pattern classification by Richard Duda
2. Han Kamber, ―Data Mining Concepts and Techniques‖, Morgan Kaufmann Publishers
3. Margaret.H.Dunham, ―Data Mining Introductory and Advanced Topics‖, Pearson Education

Online References:

Sr. No.	Website Name
1.	Machine Learning repository, https://archive.ics.uci.edu/ml/index.php
2.	Introduction to Machine Learning, IIT, Madras, https://nptel.ac.in/courses/106106139
3.	Introduction to Machine Learning, IIT, Khargpur, https://nptel.ac.in/courses/106105152

Assessment:

Internal Assessment (IA) for 20 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

➢ **Question paper format**

- Question Paper will comprise of a total of **six questions each carrying 15 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

DRAFT

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.

DRAFT

University of Mumbai

Course Code: _____and Course Name: .

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

NB: (13) Question 1 is compulsory. (14) 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.

****

DRAFT

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2116112	System Programming & Compiler Construction	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					
2116112	System Programming & Compiler Construction	20	20	40	60	2	--	--	100

Prerequisites:	Theoretical computer science, Operating system. Computer Organization and Architecture
Course Objectives:	1. To understand the design and functioning of system programs such as assemblers, loaders, linkers, and macro processors. 2. Explain the phases of compiler design including lexical, syntax, and semantic analysis. 3. Apply algorithms and techniques for symbol table management and intermediate code generation. 4. Analyze different code optimization techniques. 5. Demonstrate the process of code generation and error handling. 6. Integrate system programming concepts with compiler construction for software development.
Course Outcomes:	Upon completion of the course, the learners will be able to: 1. Describe the structure and functions of system programs such as assemblers, loaders, linkers, and macro processors. 2. Design and simulate assembler and macro processor. 3. Design concepts of linker and loader. 4. Construct various phases of a compiler and their interactions. 5. Apply lexical and syntax analysis techniques to source programs. 6. Implement intermediate code generation and optimization modules.

Detailed Contents:

Sr. No.	Name of Module	Detailed Content	Hr	CO
0	Pre-requisite	Basic Mathematical Fundamentals and knowledge of Probability, Basics of Theory of Computer Science – Grammar.		
I		**Software** Overview of System Programs – Assemblers, Loaders, Linkers	1	CO1

DRAFT

	Introduction to System Software	**Software** Overview of System Programs – Macro Processors, Compilers, and Operating Systems.	1	
		Machine structure, Language processing activities. Gap identified between programming language and system programming, System software vs application software	1	
		Self-learning Topics: Evolution of programming environments, Role of OS in program execution, Cross-compilers.		
II	Assembler and Macro Processor	Assembler: Elements of Assembly Language programming, Assembly scheme.	4	CO2
		Assembler Design: Two pass assembler Design, Data structed used.		
		Macro Processor – Macro definitions and calls, Macro expansion, Design of Two pass macro processor, data structures used.	4	
		Self-learning Topics: Nested macros, Conditional and parameterized macros, Assembler directives in x86 and ARM architectures.		
III	Loader and Linker	Loaders– Functions, Types (absolute, relocating, dynamic).	4	CO3
		Linking – Linking and relocation concepts, Static vs dynamic linking. Linker design – Direct Linking Loader, algorithms and data structures.	4	
		Self-learning Topics: Dynamic linking in modern operating systems, Shared libraries (.dll, .so). bootstrap loaders.		
IV	Introduction to Compiler & Lexical Analysis	Phases of compiler – structure and functions.	1	CO4
		Lexical analysis: Role of lexical analyzer, Lexical analysis: Regular expressions, finite automata, token generation	1	
		Parsing techniques – concepts of First and Follow. Recursive descent Parser	3	
		Parsing techniques – LL(1) parser/ Predictive parser,	2	
		Parsing techniques – LR(0)/ LALR(1).	2	
		Self-learning Topics: Use of Lex tool for token generation, Automata-based pattern matching.		
V	Syntax and Semantic Analysis	Syntax-directed translation, Construction of parse trees, Symbol table organization.	2	CO5
		Semantic analysis and type checking, Intermediate code generation. 3Address Code Format.	2	
		Self-learning Topics: Implementing LL and LR parsers using Yacc / Bison tools, Syntax-directed translation schemes.		
VI		Optimization: Constant folding, Copy propagation, Dead code elimination, Loop optimization, Peephole optimization.	3	CO6
		Code generation – Target code issues, register allocation, Instruction selection, Directed Acyclic Graph (DAG) representation.	4	

DRAFT

Code Optimization and Generation		**Self-learning Topics:** JIT (Just-In-Time) compilation **Code**		

Text Books:	1. Aho, A.V., Lam, M.S., Sethi, R., Ullman, J.D. — *Compilers: Principles, Techniques, and Tools* (2nd Ed.) 2. Donovan, J. — *Systems Programming*, McGraw Hill 3. Dhamdhere, D.M. — *Systems Programming and Operating Systems*, Tata McGraw Hill 4. Alfred V. Aho & Ullman, J.D. — *Principles of Compiler Design*, Addison Wesley 5. *Pal, Srimanta.* Systems Programming. *1st ed., Oxford University Press, 2011. ISBN 978-0198070887.*
References Books :	1. Fischer, C.N., LeBlanc, R.J. — *Crafting a Compiler* 2. Holub, A.I. — *Compiler Design in C* 3. Levine, J.R. — *Lex & Yacc*, O'Reilly 4. Appel, A.W. — *Modern Compiler Implementation in C*
Online References:	Compiler Design (IIT Kharagpur) (via NPTEL) Link: https://onlinecourses.nptel.ac.in/noc21_cs07/preview An Overview of a Compiler – Part 1 (IISc) Link: https://nptel.ac.in/courses/106108052 **MOOC listings: Coursera – Compilers Courses & Certificates** Link: Coursera – Compilers
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 15 marks Qu-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 Qu-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

DRAFT

University of Mumbai

Course Code: _____and Course Name: _

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

NB: (15) Question 1 is compulsory. (16) 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.

DRAFT

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2116113	Internet of Things (IoT)	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					
2116113	Internet of Things (IoT)	20	20	40	60	2	--	--	100

Course Objectives:

1. To understand the fundamental architecture and software-driven design of IoT systems.
2. To analyze IoT communication models and integration with cloud platforms.
3. To describe IoT network concepts and its applications.
4. To integrate IoT devices with cloud-based software platforms
5. To apply data acquisition and analytics in real-time IoT systems.
6. To incorporate ethical and secure development practices in IoT software.

Course Outcomes

1. Explain IoT architecture, components, and software layers.
2. Apply IoT communication protocols for device interconnectivity.
3. Understand IoT network concepts and various communication protocols.
4. Integrate IoT devices with cloud-based software platforms.
5. Analyze IoT data using software analytics frameworks and apply security principles and ethics in IOT systems.
6. Building IoT application prototypes using IoT components, open source tools and APIs.

Prerequisite: C Programming, Digital Logic and Computer Architecture, Microprocessor,

DRAFT

Computer Networks.

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
I	Introduction to IoT	**Introduction**: IoT definition, characteristics, and evolution; IoT ecosystem and software-oriented architecture; IoT vs. traditional embedded systems. **Sensors, Actuators, smart objects, and Interfacing**: Types of sensors, signal conditioning; DAC/ADC, interfacing protocols (I2C, SPI, UART); Actuators (motors, relays, etc.) **Enabling technologies**: Cloud, AI, Big Data, 5G, Edge Computing; IoT application domains and use cases. **Self-learning Topics:** IoT Data Management and Compute Stack	6	CO1
II	IoT Architecture and Software Stack	**IoT architecture models**: Three-layer, Five-layer, SOA-based, IoTWF, M2M; Device abstraction and virtualization; IoT software stack and middleware components. **Overview of development boards**: Raspberry Pi, ESP32, Arduino; Operating systems for IoT: FreeRTOS, Contiki, TinyOS. **Self-learning Topics**: Edge–Fog–Cloud architecture and interoperability	6	CO2
III	IoT Communication Protocols	**IoT communication models and network design**:- PHY / MAC layer: BLE, ZigBee, LoRa, NB-IoT, WiFi, 802.15.4; IPv6 for constrained, 6LoWPAN, 6Lo, 6TiSCH; Routing protocols: RPL - Mobility, fragmentation, header compression. **Transport and Application layer Protocols**: Publish/subscribe models, message brokers; MQTT, CoAP, HTTP/HTTPS, AMQP, WebSocket; QOS, reliability, congestion issues; Message queuing and data serialization (JSON, CBOR); Case Study: MQTT-based smart monitoring. **Self-learning Topics**: Genetic Based Protocols	9	CO3
IV	IoT Software Platforms and Cloud Integration	**IoT software**: IoT middleware, APIs, and SDKs. **Cloud integration**: AWS IoT Core, Azure IoT Hub, Google IoT Core, ThingSpeak. **Data pipelines**: Device-to-cloud / Cloud-to-device; RESTful APIs and Node-RED flow programming. **Self-learning Topics**: IoT visualization dashboards	6	CO4
V	IoT Data Analytics and Security	IoT data lifecycle and management. **Databases**: InfluxDB, MongoDB, Firebase. **Edge analytics**: Edge	6	CO5

DRAFT

		analytics using Python and TensorFlow Lite. **IoT security fundamentals**: Authentication, Encryption, Secure Communication. **Self-learning Topics**: Data privacy and ethical design		
VI	**IoT Applications and Case Studies**	**IoT Applications**: Smart homes, smart cities, healthcare, industrial IoT; Precision agriculture and energy management; IoT–AI integration for intelligent decision-making; Sustainable IoT and environmental monitoring **Self-learning Topics**: Emerging IoT research directions Digital twins, blockchain in IoT, Industrial IoT (IIoT), Industry 4.0 , 5G/6G synergies.	**6**	**CO6**

DRAFT

Text Books:

1. Arshdeep Bahga, Vijay Madisetti — *Internet of Things: A Hands-on Approach*, Universities Press.
2. David Hanes , Gonzalo Salgueiro , Patrick Grossetete , Robert Barton , Jerome Henry - IoT *Fundamentals: Networking Technologies,* Cisco Press, First Edition, 2017**.**
3. Sudip Misra, Anandarup Mukherjee, Arijit Roy, "Introduction to IoT", Cambridge University, 2021
4. Raj Kamal — *Internet of Things: Architecture and Design Principles*, McGraw Hill.
5. Pethuru Raj, Anupama C. Raman — *The Internet of Things: Enabling Technologies, Platforms, and Use Cases*, CRC Press.
6. Brian Russell, Drew Van Duren — *Practical Internet of Things Security*, Packt Publishing.

Reference Books:

1. Dieter Uckelmann et al. — *Architecting the Internet of Things*, Springer.
2. Adrian McEwen — *Designing Connected Products*, O'Reilly.
3. Morgan Kaufmann — *Building the Internet of Things*.

Online References:

Sr. No.	Website Name
1.	https://onlinecourses.nptel.ac.in/noc19_cs65/preview
2.	https://nptel.ac.in/courses/106105166
3.	https://www.netacad.com/courses/introduction-iot?courseLang=en-US

Assessment:
Internal Assessment (IA) for 20 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.

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.

DRAFT

University of Mumbai

Course Code: _____and Course Name: .

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

NB: (17) Question 1 is compulsory. (18) 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
2116114	Digital Imaging Techniques and Analysis	3	-	-	3	-	-	3

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				
2116114	Digital Imaging Techniques and Analysis	20	20	40	60	-	-	100

Course Objectives

1. Apply color models and transformations to digital images.
2. To enable students to understand the concepts of image enhancement and filtering in spatial and frequency domains and develop the ability to apply these techniques to improve digital image quality.
3. To teach students methods for image restoration and compression to improve and efficiently store digital images.
4. To enable students to apply image segmentation and morphological processing techniques for effective analysis, enhancement, and interpretation of digital images.

DRAFT

5. To introduce students to image feature extraction, texture analysis, template matching, and classification techniques, enabling them to analyze and interpret digital images effectively.
6. To introduce students to deep learning concepts and Convolutional Neural Networks (CNNs) for image analysis, and to demonstrate practical applications using transfer learning in various domains.

Course Outcomes

1. Apply color models and transformations to digital images.
2. Students will be able to understand and apply image enhancement and filtering techniques in spatial and frequency domains.
3. Students will be able to restore degraded images and apply image compression techniques while evaluating their performance.
4. Apply image segmentation and morphological processing techniques to extract information and features from digital images
5. Students will be able to extract and analyze image features, apply template matching, and implement classification techniques for object and pattern recognition.
6. Students will be able to apply deep learning techniques, including CNNs and transfer learning, for tasks such as image classification, object detection, image segmentation, and facial recognition.

DETAILED SYLLABUS

Sr. No.	Module	Detailed Content	Hours	CO Mapping
0	Prerequisite	Basic knowledge of linear algebra, probability & statistics, digital signal processing fundamentals, programming (C/Python), and image representation concepts.		
I	**Fundamentals of Digital Imaging**	Introduction to Digital Imaging, Image Formation and Representation, Image Sampling and Quantization, Color Models and Image Representation Image Histogram and Basic Intensity Transformations, Spatial and Frequency Domain Fundamentals. **Self-learning Topics:** Study of basic image acquisition process and imaging sensors.	**04**	**CO1**
II	**Image Enhancement Techniques**	Introduction to digital images and processing, Gray Level Transformations, Zero Memory Point Operations, negative, and contrast stretching. Dynamic range Compression, Gray Level Slicing. Bit Plane Slicing. Histogram Processing, Histogram equalization. Different types of noise (e.g., Gaussian, Salt & Pepper). Neighbourhood Processing, Spatial Filtering, Smoothing and Sharpening Filters, spatial filters, including smoothing filters (e.g., mean and median) and sharpening filters (e.g., Laplacian and unsharp masking). Image Subtraction and Image Averaging. Zooming by Interpolation and Replication Frequency domain techniques, Discrete Fourier Transform (DFT) and its use in image enhancement. **Self-learning Topics:** Introduction to advance	**8**	**CO2**

DRAFT

		Colour Image Enhancement and AI based tools for Image Processing.		
III	**Image Restoration and Compression Methods**	**I. Image Restoration (4 Hours)** • Image degradation and restoration concepts, causes of degradation • Noise models: Gaussian, Salt & Pepper, Uniform, Rayleigh, Exponential • Spatial filters: Mean (Arithmetic, Geometric, Harmonic), Median, Max, Min • Frequency filters: Inverse, Wiener, Adaptive **II. Image Compression (4 Hours)** • Fundamentals: Need, compression model, redundancies • **Lossless:** RLC, Huffman, Arithmetic, LZW • **Lossy:** Transform coding (DCT–JPEG), Predictive coding (DPCM) • Performance metrics: Compression ratio, Bit rate, MSE, PSNR **Self-learning Topics**Wavelet-based coding (JPEG 2000), Real-world case studies: Medical imaging, Satellite image compression, Military application-sonar and radar.	**08**	**CO3**
IV	**Image Segmentation and Morphological Processing**	**Image Segmentation:** Detection of Discontinuities, Edge Linking and Boundary Detection, Thresholding (Global, Adaptive, Otsu's Method), Region Oriented Segmentation (Region Growing, Splitting and Merging, Watershed Segmentation). **Morphological Image Processing:** Dilation and Erosion, Structuring Element and Its Decomposition, Opening and Closing, Morphological Gradient, Hit-or-Miss Transformation, Top-hat and Bottom-hat Transformations, Morphological Reconstruction. **Self-learning Topics:** Marker-controlled Watershed Segmentation, Morphological Skeletonization and Thinning, K-means and Graph-based Image Segmentation Techniques.	**6**	**CO4**

DRAFT

V	**Feature Extraction and Analytical Techniques**	• **Image Features:** Local & global features, feature descriptors, boundary & region descriptors, feature extraction • **Texture Analysis:** GLCM, Haralick features, Laws' measures, Local Binary Patterns (LBP) • **Principal Component Analysis (PCA):** Feature selection and dimensionality reduction • **Template Matching:** Feature matching (Euclidean, cosine), stereo matching, SIFT, SURF • **Image Analysis:** Classification (supervised/unsupervised), KNN, SVM, Decision Trees, object & pattern recognition **Self Learning Topic:** Histogram of Oriented Gradients (HOG) for object detection, Gabor filters and Wavelet-based texture analysis, Remote sensing image analysis using texture and spectral features	**8**	**CO5**
VI	**Deep Learning for Image Understanding and Applications**	Introduction to deep learning concepts for image analysis, Role of Convolutional Neural Networks (CNNs) in feature extraction and classification, Overview of transfer learning using pretrained models Applications in object detection, image segmentation, facial recognition, medical and industrial imaging **Self-learning Topics:** Generative Adversarial Networks (GANs) for image synthesis and enhancement, Vision Transformers (ViT) and hybrid CNN-Transformer architectures, Multimodal Learning — integrating text and image features (e.g., CLIP, BLIP)	**4**	**CO6**

DRAFT

Text Books:

1. Digital Image Processing, Rafael C. Gonzalez and Richard E. Woods, Pearson International Edition
2. Image Processing, Analysis, and Machine Vision, Milan Sonka, Vaclav Hlavac, Roger Boyle, Cengage India, 4th Edition.

Reference Books:

1. R.C. Gonzalez and R.E. Woods, Digital Image Processing, Pearson Education.
2. Anil K. Jain, Fundamentals of Digital Image Processing.
3.William K. Pratt, Digital Image Processing: PIKS Inside.
4. Ian Goodfellow, Yoshua Bengio, and Aaron Courville, Deep Learning, MIT Press, 2016.
5. Adrian Rosebrock, Deep Learning for Computer Vision with Python, PyImageSearch, 2019.
6. Richard Szeliski, Computer Vision: Algorithms and Applications, Springer, 2022 (2nd Edition).

Online Resources:

Sr. No.	Website Name
1.	https://nptel.ac.in/courses/117105135
2.	https://onlinecourses.swayam2.ac.in/nou25_cs22/preview

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.

DRAFT

University of Mumbai

Course Code: _____and Course Name: .

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

NB: (19) Question 1 is compulsory. (20) 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.

DRAFT

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		TH	Pract.	Tut.	Theory	Pract.	Tut.	Total
2116115	Natural Language Processing	3	-	-	3	-	-	3

ourse Code	Course Name	Theory			Term work	Pract / Oral	Total
		Internal Assessment	End	Exam			

		IAT-I	IAT-II	IAT-I + IAT-II	Sem Exam	Duration (in Hrs)			
2116115	**Natural Language Processing**	**20**	**20**	**40**	**60**	**2**	**--**	**--**	**100**

Rationale:
Most of the engineering branches are being off-spring of Computer Science, and Natural Language Processing (NLP) has emerged as one of its most dynamic and interdisciplinary domains. It enables machines to process, understand, and generate human language using computational and linguistic approaches. The course aims to develop students' analytical and problem-solving skills through exposure to core NLP techniques and applications such as chatbots, information retrieval, sentiment analysis, and machine translation, preparing them for advanced research and industry roles in Artificial Intelligence.

Prerequisites:	Fundamentals of Mathematics with basic knowledge of Probabilities, Theory of Computer Science and Python for programming assignment.
Course Objectives:	1. Introduce foundational NLP concepts, challenges, and basic text processing techniques. 2. Understanding of statistical models for text processing, tagging, and classification. 3. Explain syntactic structures and parsing methods for linguistic analysis. 4. Develop understanding of semantic representation, lexical resources, and word embeddings. 5. Familiarize students with dialogue systems, reference resolution, and discourse concepts. 6. Introduce transformer-based language models and methods for building modern NLP applications.
Course Outcomes:	Upon completion of the course, the learners will be able to: 1. Apply fundamental NLP concepts and tools to analyze and process basic text data. 2. Design and evaluate text processing and tagging systems using statistical NLP models. 3. Evaluate parsing techniques to determine effective methods for syntactic analysis 4. Design semantic analysis systems using lexical resources and embedding techniques. 5. Create simple dialogue systems and resolve references in text. 6. Design and evaluate advanced NLP systems using transformer-based LLMs.

Detailed Contents:

Sr. No.	Name of Module	Detailed Content	Hrs	CO
0	Prerequisite	Basic Mathematical Fundamentals and knowledge of Probability, Basics of Theory of Computer Science – Grammar.		
I	**Introduction to NLP and Text Processing**	What is Natural Language Processing (NLP)? History of NLP, Generic NLP System, Levels of NLP	**1**	**CO1**
		Knowledge in Language Processing, Ambiguity and Layers of NLP.	**1**	
		Challenges of NLP, Processing Indian Languages, Applications of NLP.	**1**	
		Basic Text Processing: Tokenization, Stopword Removal, Stemming and Lemmatization, Regular Expressions	**2**	
		Introduction to NLP Tools: NLTK, SpaCy	**1**	

DRAFT

		Self-learning Topics: Limitations of NLP, Study the simple pipeline architecture for a spoken dialogue system		
II	**Word Level Analysis**	Unigram, Bigram and N-gram language models and smoothing techniques (Laplace, Good-Turing) Evaluating language models (Perplexity, Cross-entropy)	**2**	**CO2**
		Finite State Automata, Morphological parsing with Finite State Transducer (FST); Porter Stemmer algorithm. Text Classification: Bag-of-Words, TF-IDF, Naive Bayes	**2**	
		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	**3**	
		Self-learning Topics: TF-IDF, Neural POS Tagging		
III	**Syntax Analysis**	Context-Free Grammars (CFG) and parse trees; Linguistics of Parsing	**1**	**CO3**
		Algorithmic of Parsing, Constituency Parsing: Rule Based (Top-Down, Bottom-Up, CYK)	**1**	
		Statistical Parsing, Dependency Parsing and Neural Parsing	**3**	
		Self-learning Topics: Evaluating parsers, Parsers based language modelling, Regional languages POS tree banks		
IV	**Semantic Analysis**	Introduction, Meaning Representation; Lexical Semantics; Study of Various language dictionaries like WorldNet, Babelnet; Lexical Relations	**2**	**CO4**
		Word Embeddings: Word2Vec, CBOW, Skip-gram, GloVe	**1**	
		Word Sense Disambiguation (WSD) methods: Supervised, Unsupervised Document representation: TF-IDF, Doc2Vec	**2**	
		Named Entity Recognition (NER)	**1**	
		Self-learning Topics: Dictionaries for regional languages, Distributional Semantics, Topic Models		
V	**Pragmatic & Discourse Processing**	Discourse Structure, Cohesion, Discourse Coherence & Structure	**1**	**CO5**
		Reference Resolution, Reference Phenomena, Anaphora Resolution using Hobbs Algorithm, Centering Algorithm, Mitkovs Pronoun Resolution Algorithm	**3**	
		Self-learning Topics: Multimodal Discourse (Text + Image, Text + Video), Politeness and Emotion Detection in Conversations		
VI	**NLP Applications and Models**	Information extraction, Sentiment analysis, Text summarization Question answering systems, Machine translation:	**4**	**CO6**
		Introduction to Large Language Models(LLM), Fundamental components of the Transformer model	**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)	**4**	
		Self-learning Topics: Ethics, fairness, and bias in NLP Models, Use of NLP in Recommender Systems, Federated Learning for NLP Applications		

DRAFT

Text Books:	1. T. Siddiqui and U. Tiwari, Natural Language Processing and Information Retrieval, Oxford University Press 2. Pushpak Bhattacharyya and Aditya Joshi, Natural Language Processing, Wiley India Pvt. Ltd., Edition: 2023 3. Daniel Jurafsky. James H. Martin, Speech and Language Processing, An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models, Third Edition (draft) Aug 2025
References Books :	1. Bird, S., Klein, E., & Loper, E. *Natural Language Processing with Python*. O'Reilly. 2. Palash Goyal, Sumit Pandey, Karan Jain. *Deep Learning for Natural Language Processing*-2018 3. Christopher D.Manning and HinrichSchutze, Foundations of Statistical Natural Language Processing, MIT Press, 1999.
Online References:	https://onlinecourses.nptel.ac.in/noc23_cs45/preview https://archive.nptel.ac.in/courses/106/106/106106211/ https://www.nltk.org/book/ https://web.stanford.edu/class/cs224n/
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 15 marks Qu-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 Qu-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

DRAFT

University of Mumbai

Course Code: _____and Course Name: .

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

NB: (21) Question 1 is compulsory. (22) 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.

****

DRAFT

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2116116	Augmented and Virtual Reality	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					
2116116	Augmented and Virtual Reality	20	20	40	60	2	--	--	100

Rationale:
Extended Reality (XR)—including AR, VR, and MR—is rapidly transforming digital interaction, making it an essential interdisciplinary area for modern engineering. This course builds a strong foundation from core concepts of human perception and tracking to hands-on development using industry-standard tools such as Unity, ARKit, and ARCore. Students will gain practical expertise in 3D modeling optimization, immersive interaction design, and deployment across XR devices. By learning to design and implement innovative spatial applications, graduates will be prepared to contribute to advanced sectors such as industrial training, complex visualization, digital twins, and the emerging metaverse.

Prerequisites:	Computer Graphics Fundamentals, Mathematics for XR, Mobile App Development Basics
Course Objectives:	1. To introduce the fundamental concepts of Augmented Reality (AR), Virtual Reality (VR), Mixed Reality (MR), and Extended Reality (XR). 2. To gain practical skills in designing and developing XR environments. 3. To apply 3D graphics and content creation techniques in immersive systems. 4. To design effective user interactions and enhance usability in XR experiences. 5. To explore advanced and emerging technologies driving innovation in XR. 6. To develop and demonstrate XR solutions for real-world applications.
Course Outcomes:	Upon completion of the course, the learners will be able to: 1. Describe the fundamental concepts, hardware, and applications of AR, VR, and XR technologies. 2. Analyze XR hardware and software platforms and configure effective development environments for immersive applications. 3. Design and develop basic AR/VR applications with spatial awareness and user interactivity using standard tools. 4. Create and optimize 3D assets and integrate them effectively into immersive XR applications. 5. Evaluate interaction design and user experience to ensure accessible and comfortable XR environments. 6. Apply advanced XR concepts to develop and present an innovative functional prototype.

Detailed Contents:

DRAFT

Sr. No.	Name of Module	Detailed Content	Hrs	CO
0	Prerequisite	Basic Mathematical Fundamentals and knowledge of Probability, Basics of Theory of Computer Science – Grammar.		
I	**Introduction to Augmented & Virtual Reality**	Introduction to AR, VR, MR, and XR – Definitions, components and evolution, Human perception and immersion (stereo vision, FOV, motion tracking); Type of immersion; AR systems classification – marker-based, markerless, and spatial; Basics of VR systems – head-mounted displays, projection systems; Hardware components: Display devices: LCD, OLED, Audio: Speakers, Earphones, Bone conduction, Touch: Haptic Device, GPU and CPU, Input devices like game controllers, data gloves, joysticks Tracking Hardware: Inertial Measurement Unit (IMU), Gyroscope, Accelerometer; AR/VR use cases in gaming, education, healthcare, and manufacturing; Current XR devices – Meta Quest, Apple Vision Pro, HoloLens; Career opportunities and future trends in immersive technology.	**6**	**CO1**
		Self-learning Topics: Basics of coordinate systems, Basics of 3D coordinate systems (X, Y, Z axes, transformations), Basics of computer graphics (rendering pipeline, polygons), Reading on input/output devices used in XR		
II	**Hardware, Tracking & System Design in XR**	XR Hardware Overview: Headsets (mobile, standalone, PC-tethered) Controllers, sensors, cameras, Device constraints: processing, battery, field-of-view Tracking Techniques: Inside-out vs outside-in tracking, Basics of SLAM (Simultaneous Localization and Mapping), Hand/eye tracking introduction XR System Architecture (Introductory): Visual, aural, and haptic display overview, Integration of tracking and display for basic XR applications, Latency, refresh rate, and frame drop basics XR Development Platforms: Unity, Unreal Engine, ARKit, ARCore Overview of XR SDKs, standards, and APIs: OpenXR, WebXR basics Usability & Human Factors (Introductory): Space, scale, and ergonomics in immersive environments Basic comfort evaluation: gaze direction, field-of-view considerations Motion, simulator, and cyber sickness: causes and simple mitigation	**6**	**CO2**
		Self-learning Topics: Basic Mixed Reality UI Concepts, Simple Multi-Modal Feedback, Introduction to User Comfort Considerations Mobile XR vs Desktop XR		
III	**Development of AR/VR Applications**	Unity Basics for XR: Interface, scene hierarchy, prefabs Understanding objects, lights, cameras, and basic scene setup AR Development: Marker-based AR demo with ARCore/ARKit, Markerless AR: plane detection, anchors, object placement VR Navigation Techniques: Teleportation Locomotion, (keyboard/mouse or controller-based), Steering basics Object Interaction in VR: Grab, rotate, scale objects, Simple physics-based interactions, Integration of Audio & Haptics: Adding sound effects and vibration/feedback in XR apps Debugging & Deployment: Test and deploy XR apps on mobile devices Basic performance observation (latency, frame drop)	**7**	**CO3**

DRAFT

		Self-learning Topics: ARCore & ARKit Documentation, Unity Shaders & Materials Basics, Package Managers & SDK Updates, Simple Interaction Observations, Basic Navigation Awareness		
IV	**3D Assets and Content Creation**	3D models & file formats: FBX, OBJ, glTF, Understanding model structure, vertices, meshes, and hierarchy Using Online 3D Asset Libraries: Sketchfab, Google Poly, Unity Asset Store, How to search, import, and use assets in XR projects Basics of 3D modelling (Blender overview concept level): Textures, materials, lighting & shadows in Unity/Unreal Optimization for XR: low-poly modelling, draw calls, texture compression, Basics of photogrammetry & LiDAR scanning for 3D asset creation, Importing & optimizing captured 3D models into Unity.	**7**	**CO4**
		Self-learning Topics: Blender tutorials (basic modeling, UV mapping), glTF standard documentation, Basics of lighting & rendering.		
V	**Interaction and User Experience Design**	Human computer interaction (HCI) basics for XR, Interaction techniques: gaze, gesture, voice, controllers Designing comfortable XR interfaces (ergonomics, usability) Accessibility in XR (for visually/physically impaired users) Ethics & privacy issues in XR (data tracking, manipulation risks) Case studies of good & bad XR UI designs, Usability evaluation methods for XR	**7**	**CO5**
		Self-learning Topics: Review XR design guidelines (Google/Apple Human Interface Guidelines for XR), Research on ethical issues in XR (privacy in AR glasses), Basics of UX evaluation methods		
VI	**Emerging Trends and Capstone Project**	AI integration in AR/VR (object recognition, generative content). Cloud XR and remote rendering concepts. Multi-user and collaborative XR environments. Industry applications: healthcare, education, manufacturing, and training. Business and innovation in AR/VR – startups and monetization. Future directions: spatial computing, digital twins, and metaverse. Capstone project planning and demonstration guidelines. **Self Learning:** Explore as Unity, ARCore, ARKit, and Oculus SDK Tools	**6**	**CO6**

Text Books:	1. Ralf Doerner, (2024). Virtual and Augmented Reality (VR/AR). Springer. 2. Simeone, A. (Ed.). (2024). Everyday Virtual and Augmented Reality. Springer. 3. Tacgin, Z. (2024). Virtual and Augmented Reality: An Educational Handbook. Cambridge Scholars Publishing. 4. Schmalstieg, Dieter, and Tobias Hollerer (2016) Augmented reality: principles and practice. Addison-Wesley Professional
References Books :	1. Chen, J. Y. C., & Fragomeni, G. (Eds.). (2024). Virtual, Augmented and Mixed Reality. Springer. 2. Kimberly Burk Cordova (2024). Augmented and Virtual Reality: Unlocking Business Potential for Entrepreneurs. 3. Chetankumar G Shetty, "Augmented Reality- Theory, Design and Development", Mc Graw Hill, 2020 Edition. 4. Virtual and Augmented Reality An Educational Handbook By Zeynep Tacgin

DRAFT

Online References:	https://nptel.ac.in/courses/106103842 https://nptel.ac.in/courses/106106138 https://nptel.ac.in/courses/106106699 Unity Learn: Create with VR
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 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.

University of Mumbai

Course Code: _____and Course Name: .

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

NB:
(23) Question 1 is compulsory.
(24) 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.

DRAFT

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
2116117	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					
2116117	Blockchain Technology	20	20	40	60	2	--	--	100

Rationale:
To enable learners to stay updated with evolving blockchain innovations and prepares them to address future technological and research challenges.

Course Objectives:
1. To understand the cryptographic and distributed system principles underlying blockchain technologies.
2. To explore various blockchain architectures and governance models.
3. To interpret different consensus mechanism in public and private blockchains
4. To design and develop smart contracts and decentralized applications (Dapps) using leading blockchain platforms (Ethereum, Hyperledger, Polygon).
5. To understand different types of crypto assets.
6. To analyse blockchain scalability, privacy, and interoperability challenges.

Course Outcomes: Six Course outcomes
(Based on Blooms Taxonomy)

1. Explain the fundamental principles, cryptography, and architecture of blockchain systems. (BL2)
2. Explain blockchain architecture, components, and types. (BL2)
3. Compare and contrast consensus protocols and assess their performance in different blockchain types. (BL3)
4. Design, code, and deploy smart contracts using Solidity and Web3 frameworks. (L3/L5).
5. Interpret different Crypto assets and Crypto currencies. (L3).

DRAFT

6. Analyze ethical, regulatory, and sustainability issues in blockchain adoption. (L4)

Pre-requisite:

Computer networks, Cryptography, and Programming concepts (Python, Java, or C++)

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
0	**Pre-requisite**	**Fundamentals of Computer Networks:** Understanding of distributed systems, peer-to-peer communication, and internet protocols. **Basics of Cryptography**: Concepts of hashing, symmetric and asymmetric encryption, and digital signatures. **Database and Data Structures Concepts:** Knowledge of data storage models, indexing, and linked data structures for efficient transaction management.	**2**	
I	**Cryptographic Foundations**	Public key cryptography (RSA, ECC), Hash functions (SHA-256, Keccak-256), digital signatures, Evolution of distributed ledger systems, centralized vs. decentralized systems, Types of blockchain: Public, Private, Consortium, Hybrid, CAP Theorem, DLT *vs.* Blockchain, Software stack of blockchain platforms **Self-learning Topics:** Quantum-Resistant Cryptography for Blockchain Security, Cross-Chain Communication Protocols	**5**	**CO1**
II	**Blockchain Fundamentals**	Block Structure and characteristics of blockchain, Bitcoin basics, Key Concepts: Rewards, Miners, Difficulty Levels, Forks: Hard and Soft Fork, Wallet and its types, Merkle trees, UTXO, Bitcoin addresses, Transactions in Blockchain, Double Spending. **Self-learning Topics:** Bitcoin history, Bitcoin Addresses, Blockchain data visualization.	**5**	**CO2**
III	**Blockchain Consensus**	**Permissionless Blockchain Algorithms:** Proof of Work (PoW), Proof of Stake (PoS), Delegated PoS, Proof of Authority, Proof of Elapsed Time, and Proof of Burn. **Permissioned Blockchain Algorithm:** Byzantine Fault Tolerance (BFT) and Practical BFT, Solo, Kafka, RAFT, PAXOS algorithm, Performance,	**6**	**CO3**

DRAFT

		scalability, and energy considerations, open-source consensus frameworks. **Self-learning Topics:** Hybrid and Novel Consensus Mechanisms (e.g. Proof of History, Proof of Space-Time, DAG-based Consensus), and AI-optimized consensus algorithms.		
IV		**Ethereum**: Introduction, Turing Completeness, Ethereum Architecture, EVM (Ethereum Virtual Machine), Transaction Life cycle, Wallets, Nonce, Gas mechanism and EOA. **Smart Contracts**: Design, compilation, and deployment. **Solidity Programming**: Data types, functions, modifiers, and events, Development tools: Remix IDE, Truffle, Ganache, MetaMask. **Self-learning Topics:** Layer 2 Solutions and Ethereum Scaling (Optimistic Rollups, zk-Rollups, and Sidechains), Ethereum 2.0 and the Transition to Proof of Stake (The Merge and Sharding)	**7**	**CO4**
V	**Decentralized Applications and**	Architecture of Hyperledger Fabric, NFT, ICO,STO and ERC-20 & ERC-721 standards. **DApp Architecture**: Frontend-backend integration, Web3.js, Ethers.js, and blockchain APIs, Off-chain data storage (IPFS, Filecoin). **Case Study:** / Supply chain using blockchain. (Can include 2 case studies; one on Ethereum and one on Hyperledger fabric) **Self-learning Topics:** Decentralized Identity (DID) and Verifiable Credentials in Dapps, Cross-Chain DApp Development and Interoperable Token Standards (ERC-1155, ERC-4626)	**7**	**CO5**
VI	**Advanced Topics and Ethical**	Blockchain interoperability (Polkadot, Cosmos), Blockchain in IoT, AI, and Edge environments, Layer 2 scaling solutions (Lightning Network, Polygon), Blockchain ethics: privacy, immutability, and sustainable mining, Security attacks on blockchain: Sybil, replay, 51% attack, Regulatory and governance aspects **Self-learning Topics:** Zero-Knowledge Proofs (ZKP) and Privacy-Preserving Blockchain Protocols, Decentralized Autonomous Organizations (DAOs) and Blockchain Governance Models	**6**	**CO6**

DRAFT

Text Books:

1. Andreas M. Antonopoulos — *Mastering Bitcoin*, O'Reilly Media.
2. Imran Bashir — *Mastering Blockchain*, Packt Publishing.
3. Joseph Bonneau et al. — *Bitcoin and Cryptocurrency Technologies*, Princeton University Press.
4. Narayan Prusty — *Building Blockchain Projects*, Packt Publishing.
5. Roger Wattenhofer — *The Science of the Blockchain*, Inverted Forest Publishing.

6. Daniel Drescher — Blockchain Basics: A Non-Technical Introduction in 25 Steps, Apress.

Reference Books:

1. Arshdeep Bahga and Vijay Madisetti — *Blockchain Applications: A Hands-on Approach*, Universities Press.
2. Narayan Prusty — *Blockchain for Enterprise Applications*, Packt Publishing.
3. Henning Diedrich — *Ethereum: Blockchains, Digital Assets, Smart Contracts, Decentralized Autonomous Organizations*, Wiley.
4. Ritesh Modi — *Solidity Programming Essentials*, Packt Publishing.
5. Chris Dannen — *Introducing Ethereum and Solidity: Foundations of Cryptocurrency and Blockchain Programming for Beginners*, Apress.
6. David Shrier and Alex Pentland — *Trust::Data – A New Framework for Identity and Data Sharing*, MIT Press.

Online References:

1. Blockchain Fundamentals: https://www.coursera.org/learn/blockchain-basics
2. Ethereum Official Documentation: https://ethereum.org/en/developers/docs/
3. Bitcoin Whitepaper by Satoshi Nakamoto: https://bitcoin.org/bitcoin.pdf
4. IBM Blockchain Documentation and Tutorials: https://www.ibm.com/docs/en/blockchain
5. Hyperledger Foundation (Linux Foundation Projects): https://www.hyperledger.org/
6. Solidity Official Documentation: https://docs.soliditylang.org/
7. NPTEL Course — Introduction to Blockchain Technology (IIT Kharagpur): https://nptel.ac.in/courses/106105184
8. Chainlink Documentation (Oracles and Smart Contract Integration): https://docs.chain.link/
9. Polygon (Layer 2 Scaling Solution) Developer Docs: https://docs.polygon.technology/
10. Consensys Academy — Blockchain Developer Resources: https://consensys.net/academy/

University of Mumbai

Course Code: _____and Course Name: .

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

NB:
(25) Question 1 is compulsory.
(26) 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**

DRAFT

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:

5. All COs should be mapped on the basis of the syllabus.
1. Module weightage as per contact hours mentioned in the syllabus
2. Question 1 is compulsory, and it should include sub-questions covering all modules, with each sub-question carrying five marks.
3. 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
2116118	User Experience Design	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					
2116118	User Experience Design	20	20	40	60	2	--	--	100

Course Objectives: This course introduces students to the creative and analytical side of designing user experiences. By the end of the course, learners will be able to:

1 Understand the core principles, processes, and mindset behind great user experiences.
2 Learn how to conduct meaningful user research and transform findings into design insights.
3 Build conceptual models, wireframes, and design structures for intuitive interfaces.
4 Create interactive prototypes and evaluate them through usability testing.
5 Understanding how AI can enhance and personalize the design process.
6 Understand the core principles, processes, and mindset behind great user experiences.

Course Outcomes:

1. Explain the fundamental concepts, lifecycle, and framework of user experience design.
2. Conduct user research and requirement analysis to identify user goals and personas.
3. Develop conceptual models, interaction designs, and information architectures.

DRAFT

4. Build prototypes using usability principles..
5. Evaluate the prototype.
6. Apply AI tools and principles to enhance UX design and future-ready interfaces.

Prerequisite: Design Techniques, Software Engineering, Design Thinking

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
0	**Prerequisite**	Design Thinking	2	-
I	**Foundations of User Experience Design**	What is user Experience, Similarities and difference between UX and UI, Principles of usability, usefulness, and desirability, A UX process lifecycle template **Self-learning Topics: Designing for a Quality User Experience in 3D Applications** **EMERGING DESIRE FOR USABILITY, Affordance**	5	**CO1**
II	**User Research and Requirement Analysis**	User Persona for UX Design, Ideation, Sketching, User Flow in UX Design In UX, what is the user flow?, What do I need to make a user experience flow?, User Research for UX Design, What is user experience research? Needs and requirements: first span of the bridge, Formal requirements extraction, Abridged methods for requirements extraction. **Self-learning Topics: UX Research with Generative AI**	8	**CO2**
III	**Conceptual Design, Information Architecture & Interaction Design**	Design-informing models: second span of the bridge, Some general "how to" suggestions, User models. Usage models, Work environment models Mental models, Conceptual design, Storyboards, Model consolidation, Design paradigms, Design thinking, Design perspectives. **Self-learning Topics: Design Systems and AI**	6	**CO3**
IV	**Prototyping**	UX Design Prototypes, Prototyping for Amazing Reasons, How do you create a user experience prototype?, Why is it important to test designs when they are still in the prototype stage?, Wireframes in UX Design, What are the benefits of wireframes?, When should you use wireframes?, What is the best way to make wireframes?, Depth and breadth of a prototype, Fidelity of prototypes, Interactivity of prototypes, Choosing the right breadth, depth, level of fidelity, and amount of interactivity. **Self-learning Topics: Prototyping with AI, Wireframe and Information Architecture with AI**	8	**CO4**
V	**UX Evaluation**	Formative vs. summative evaluation, Types of formative and informal summative evaluation	5	**CO5**

		methods, Types of evaluation data, Some data collection techniques, Variations in formative evaluation results **Self-learning Topics: Testing and Validation with AI**		
VI	**AI-Enhanced UX**	The Paradigm Shift in Digital Design, What Is Prompt Engineering?, The Designer as a Conductor, Traditional Design vs. AI-Driven Design, Types of AI Useful for UX, Advantages of Prompt-Powered Design, Risks and Challenges, Real-World Applications **Self-learning Topics:** Suggested Activity: Your First Intentional Prompt	5	**CO6**

Text Books:

1. Rex Hartson & Pardha S. Pyla (2019) – The UX Book: Process and Guidelines for Ensuring a Quality User Experience, 2nd Ed., Morgan Kaufmann.
2. Albert Chipman (2022) – UX/UI Design 2022: A Complete Beginner to Pro Step-by-Step Guide to UX/UI Design.
3. UX+AI DESIGN WITH PROMPTS How to Create Exceptional Experiences with Generative AI (Hegel Gonzalez)
4. UI Design:Key to captivate user understanding, 1st Edition, SybGen Publication, 2021(Nilakshi Jain, D.Kalbande)

References:

1. Joel Marsh (2016) – UX for Beginners: A Crash Course in 100 Short Lessons, O'Reilly Media.
2. Steven Miller (2018) – UX Design.
3. Lise Pilot (2024) – AI for Designers: Using Artificial Intelligence to Supercharge Your Workflow and Build AI-Powered Products.

Online References:

Sr. No.	Website Name
1.	NPTEL: User Interface Design

Assessment:

Internal Assessment (IA) for 20 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

➢ **Question paper format**

NB: (27) Question 1 is compulsory. (28) Solve any three questions out of five.		
Q.1	**Solve any three questions out of five: (05 marks each)**	**15**
a)		05
b)		05

DRAFT

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:

- All COs should be mapped on the basis of the syllabus.
- Module weightage as per contact hours mentioned in the syllabus
- Question 1 is compulsory, and it should include sub-questions covering all modules, with each sub-question carrying five marks.
- 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
2116411	**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				
2116411	**Mini 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.

DRAFT

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.

DRAFT

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

DRAFT

- 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

DRAFT

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2116119	Machine Learning 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				
2116119	Machine Learning Lab	--	--	--	--	25	25	50

Lab Objectives:

1. To develop practical skills in data preprocessing and feature engineering
2. To provide hands-on experience on the core concepts of machine learning techniques
3. To enable students to develop proficiency in using Python libraries for developing ML models
4. To learn to select and apply appropriate performance evaluation techniques.
5. To practice fine-tuning, cross-validation and ensemble techniques for model improvement.

6. To understand and apply ethical principles and transparency in ML experimentations/case studies.

Lab Outcomes:

1. Perform data preprocessing, cleaning, transformation, and feature engineering to prepare datasets.
2. Implement and train various machine learning models using appropriate programming languages and libraries.
3. Evaluate and compare machine learning models using suitable performance metrics and validation techniques.
4. Apply fine-tuning and ensemble techniques to improve the model's performance.
5. Demonstrate awareness of ethical considerations and transparency for the development of models/ case studies.
6. Design and justify appropriate machine learning solutions by integrating preprocessing, model selection, dimensionality reduction, and evaluation techniques for real-world datasets.

Prerequisite: Analysis of algorithms, Python Programming

DETAILED SYLLABUS:

Sr. No.	Module	Detailed Content	Hours	LO Mapping
0	Prerequisite	Engineering Mathematics, Probability and Statistics, Data Structures, Algorithms, DWM		
I	**Introduction to Machine Learning**	**Fundamentals:** Machine Learning, Types of Machine Learning, Issues in Machine Learning, Application of Machine Learning, Steps in developing a Machine Learning Application Training Error, Generalization error, Overfitting, Underfitting, Bias-Variance trade-off **Self-learning Topics:** Preprocessing Techniques, Feature Engineering	5	1
II	**Supervised Learning**	**Learning with Regression:** Linear Regression, Multivariate Linear Regression, Logistic Regression **Classification using Decision Tree**: Classification and Regression Trees using GINI Index **Support Vector Machine** Optimal decision boundary, Margins and support vectors, SVM as Constrained optimization problem, Quadratic Programming, SVM for linear and nonlinear classification, Kernel trick. **Performance Evaluation Measures:** Accuracy, Precision, Recall, F1-measure, ROC_AUC, specificity. **Self-learning Topics: Classification using ID3**	10	2,3,5
III	**Unsupervised Learning**	**Clustering:** **Fundamentals**: Definition of clusters, motivation **Distance and similarity measures**: Manhattan, Euclidian, Cosine	6	2,3,5

DRAFT

		Similarity, Inter-cluster distance, Intra-cluster distance **Graph Based Clustering:** Minimum Spanning Tree **Centroid based Clustering:** K-Means **Density based Clustering:** DBSCAN, Advantages over K-Means **Performance Evaluation Measures:** Silhouette Coefficient, Davies Bouldin Index (DBI), Dunn Index (DI) **Self-learning Topics: Hierarchical Clustering, Agglomerative Clustering**		
IV	**Ensemble Learning**	**Ensembles:** Motivation for Ensembles, Ensemble learning and generalization, Base learners vs Strong Learners, Hard voting vs Soft voting, Weighted Averaging **Bootstrap aggregating (Bagging)**, Out of Bag Error Estimation, Random Forest **Stacking:** Meta Learning, Base level and Meta level models, Blending **Boosting:** Stumps, Adaboost Comparison of Bagging vs Boosting **Self-learning Topics: Gradient boosting, XGBoost, Hyper-parameter Tuning**	6	2,3,4,5
V	**Probabilistic Learning**	**Fundamentals**: Probabilistic modeling, motivation, Deterministic vs Probabilistic modeling, Likelihood, Prior, Posterior, Maximum Likelihood Estimation (MLE) **Bayesian Belief Network (BBN):** Probabilistic Reasoning, Bayesian network, Directed Acyclic Graph (DAG) Structure of BBN, Inference using BBN **Expectation Maximization (EM):** Motivation and Need, Definition of EM, EM and MLE, EM Algorithm **Self-learning Topics: Bayesian Neural Network and its comparison with BBN,** **Applications: Probabilistic forecasting/ Clustering**	2	2,3,5
VI	**Dimensionality Reduction Techniques**	**Fundamentals**: Motivation, Curse of Dimensionality **Algorithms:** Principal Component Analysis(PCA), Singular Value Decomposition (SVD), Linear Discriminant Analysis (LDA) **Applications of PCA**: Image Compression/Image Classification	6	1,2,3,5

DRAFT

		Applications of SVD: Image Reconstruction **Applications of LDA**: Financial risk assessment/ Customer segmentation **Self-learning Topics: Independent Component Analysis (ICA)**		

Text Books:

1. Peter Harrington, "Machine Learning in Action", DreamTech Press
2. Ethem Alpaydın, "Introduction to Machine Learning", MIT Press
3. Tom M.Mitchell, "Machine Learning" McGraw Hill
4. Kevin P. Murphy , Machine Learning — A Probabilistic Perspective
5. Stephen Marshland, "Machine Learning: An Algorithmic Perspective", CRC Press.

References:

1. Pattern classification by Richard Duda
2. Han Kamber, ―Data Mining Concepts and Techniques‖, Morgan Kaufmann Publishers
3. Margaret.H.Dunham, ―Data Mining Introductory and Advanced Topics‖, Pearson Education

Online References:

Sr. No.	Website Name
1	Machine Learning repository, https://archive.ics.uci.edu/ml/index.php
2	Machine Learning using Python Programming https://onlinecourses.swayam2.ac.in/ini25_cs02/preview
3	Python for Data Science, IIT, Madras, https://nptel.ac.in/courses/106106211
4	The Joy of Computing using Python, NPTEL, https://nptel.ac.in/courses/106106211

List of Experiments:

Sr No	Suggested List of Experiments	Hrs
01	Experiment on data preprocessing/data visualization techniques such as handling missing values, standardization, normalization, encoding etc.	2
02	Build and evaluate linear regression model for prediction on real-world dataset.	2
03	Build and evaluate logistic regression model for classification on real-world dataset.	2
04	Build and evaluate Support Vector Machine (SVM) model for classification/Regression on real-world dataset.	2
05	Build and evaluate ML model for graph-based clustering (MST)	2
06	Build and evaluate ML model for density-based clustering (DBSCAN)	2
07	Implement dimensionality reduction techniques such as PCA/SVD/ LDA	2
08	Implement Bayesian Belief Network (BBN) /Expectation Maximization	4
09	Case Study/Mini Project (Students should demonstrate team-work, engineering, ethical principles etc.)	4
	List of Assignments	
1	Assignment 1 on the first three modules.	2
2	Assignment 2 on the remaining three modules.	2

Assessment:

DRAFT

Term Work: Term Work shall consist of at least 8 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.

DRAFT

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2116120	System Programming & Compiler Construction Lab	-	2	-	-	1	-	1

Course Code	Course Name	Examination Scheme			
		Theory Marks	Term	Practica	Total

		Internal assessment			End Sem. Exam	Work	l/ Oral	
		IAT-I	IAT-II	IAT-I + IAT-II				
2116120	**System Programming & Compiler Construction Lab**	**--**	**--**	**--**	**--**	**25**	**25**	**50**

Lab Objectives:

1. Apply the knowledge of one pass and two pass assemblers to generate machine code.
2. Designing macro processor requires analysis of macro instruction.
3. Analyzing syntax structure and semantic rules to build parse tree.
4. Simulate relocation and linking process by understand the system concept
5. Understand the concept of lexical analysis.
6. Investigating and Using Advanced compiler construction and tools to build or understand compiler components.

Lab Outcomes:

1. Generate machine code by implementing two pass assemblers and Design Macro Processor **L3**
2. Simulate loader and linker **L4**
3. Identify and validate token from given High level language **L4**
4. Construct Top down/Bottom-up pass tree from the given input string **L3**
5. Implementation of Synthesis Phase of compiler **L2**
6. Explore the tools like LEX, YACC, GCC, ANTLR, Python (PLY library), LLVM. **L6**

Prerequisite: Basic Knowledge of any programming language like C, C++, Python, JAVA, R etc.

DETAILED SYLLABUS:

Sr. No.	Module	Detailed Content	Hours	LO Mapping
0	Prerequisite	Basic Knowledge of any programming language like C, C++, Python, JAVA, R etc.	2	-
I	**Assembler and Macro Processor**	Assembler: Elements of Assembly Language programming, Assembly scheme. Assembler Design: Two pass assembler Design, Data structure used. **Self-learning Topics :** Design of Two pass macro processor, data structures used.	2	**LO1**
II	**Loader and Linker**	Loaders– Functions, Types (absolute, relocating, dynamic) **Self-learning Topics:** Dynamic linking in modern operating systems, Shared libraries (.dll, .so). Bootstrap loaders.	2	**LO2**
III	**Introduction to Compiler and Lexical Analysis**	Lexical analysis: Role of lexical analyzer, Lexical analysis: Regular expressions, finite automata, token generation. **Self-learning Topics:** Use of Lex tool for token generation, Automata-based pattern matching.	2	**LO3, LO6**
IV	**Syntax and Semantic**	Parsing techniques – Recursive descent, LL(1), LR(0), SLR(1), LALR(1).	2	**LO4, LO6**

DRAFT

	Analysis	**Self-learning Topics:** Implementing LL and LR parsers using Yacc / Bison tools, Syntax-directed .translation schemes.		
V	**Code Optimization and Code Generation**	.Code optimization and Code generation. **Self-learning Topics:** JIT (Just-In-Time) compilation	2	**LO5**

References:

1. Andrew W. Appel Princeton University. Jens Palsberg Modern Compiler. Implementation in Java, Second Edition. Purdue University. CAMBRIDGE University press, 2002.
2. Charles N. Fischer, Richard J. LeBlanc Crafting a compiler with C , Pearson Education 2007
3. John R. Levine, Tony Mason & Doug Brown, Lex & YACC, O 'Reilly publication, second Edition

Online Resources:

Sr. No.	Website Name
3.	https://www.geeksforgeeks.org/compiler-design/compiler-construction-tools/
4.	https://www.thefreecountry.com/programming/compilerconstruction.shtml
3.	https://github.com/Amey-Thakur/SYSTEM-PROGRAMMING-AND-COMPILER-CONSTRUCTION-AND-SYSTEM-SOFTWARE-LA
4	https://www.meegle.com/en_us/topics/compiler-design/compiler-construction-tools

Tools Recommended: LEX, YACC, GCC, ANTLR, Python (PLY library), LLVM

Suggested List of Experiments:

Sr No	List of Experiments	Hrs
1	Implementation of two-pass assembler for a simplified assembly language.	2
2	Design of a macro processor supporting nested macros	2
3	Simulation of a linking loader with relocation functionality	2
4	Lexical analyzer design using **LEX** or **Python regular expressions**.	2
5	Design and implement a program to Syntax analyzer (Top-down parser) construction using. (Any parser)	2
6	Design a program for Syntax analyzer (Bottom-up parser) construction using **YACC / PLY / ANTLR**. (Any parser)	2
7	Implementation of simple code optimization techniques	2
8	Intermediate code generation for arithmetic expressions.	2

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

Assessment:

Term Work: Term Work shall consist of at least 7 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.

DRAFT

DRAFT

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total

2116121	**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**				
2116121	**Internet of Things Lab**	**--**	**--**	**--**	**--**	**25**	**25**	**50**

Lab Objectives:

1. To familiarize students with the software tools, hardware platforms, and cloud services used in IoT applications.
2. To enable students to interface sensors, actuators, and communication modules with microcontrollers such as ESP32 and Raspberry Pi.
3. To develop programming skills for acquiring, processing, and transmitting the data to IoT cloud platforms like Firebase, ThingSpeak, and MongoDB.
4. To provide hands-on experience in implementing IoT communication protocols such as MQTT and HTTP for real-time data exchange.
5. To cultivate analytical skills for integrating IoT systems with databases and cloud platforms, supporting intelligent data-driven applications.
6. To create the database for machine learning applications

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

1. Identify various IoT hardware, software tools and cloud platforms required for IoT system development.
2. Demonstrate the interfacing of sensors and actuators with ESP32 and Raspberry Pi for real-time monitoring and control applications.
3. Develop programs for acquiring sensor data and uploading it to cloud databases.
4. Implement IoT communication protocols like MQTT for data publishing and subscription.
5. Integrate IoT devices with NoSQL databases and visualize the data for decision-making.
6. Design and validate a complete IoT-based system that includes sensing, communication, storage, and actuation components.

Suggested list of Experiments.

Sr No	Title of the Experiments	LO
01	Installation of software tools such as Arduino IDE, MongoDB NoSQL Databse, Node JS, libraries of various peripherals such as DHT11/DHT22 library, Max30100, I2C LCD, Goolge FireBase etc. for Arduino IDE	LO1
02	Interfacing switch buttons and LEDs with ESP32. Develop a program to control LED as actuators based on action given by the switch buttons.	LO2
03	Interface LEDs with ESP32 and control these LEDs via smartphone by configuring Bluetooth Low Energy (BLE).	LO2
04	Interface Ultrasonic sensor SR-04 with ESP32 and write a program to upload the data on ThinkSpeak. Visualize the data using a gauge. Trigger an event after crossing a threshold.	LO3

DRAFT

05	Interface DHT11/22 with ESP32 and write a program to upload temperature and Humidity data on Google FireBase database.	LO3
06	Interface Max30100 sensor with ESP32 and upload the data on Google FireBase. (Use the RTOS to enable uploading the data)	LO3
07	Implement MQTT protocol to publish and subscribe to topics with the help of ESP32,Node-RED and broker like Mosquitto.	LO4
08	Interface GSM module (SIM800L) with ESP32 and write a program to establish a call, disconnect a call and send the SMS.	LO6
09	Read the data from the sensor by using ESP32 and upload on MongoDB database. Export the data in CSV format.	LO5
10	Interfacing LED with Raspberry Pi and control the LED using MQTT Protocol.	LO4
11	Read the data from DHT11 sensor using Raspberry Pi and upload this data on Thingspeak cloud.	LO3
12	Capstone Project: Complete IoT system (e.g., Smart Parking / Environmental Monitor).	LO6

Online References:

1. https://nodered.org/docs/getting-started/
2. https://www.arduino.cc/en/software
3. https://www.raspberrypi.com/software/
4. https://www.mongodb.com/

Assessment:

Term Work: Term Work shall consist of at least 10 practicals based on the above list.

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.

DRAFT

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2116122	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				
2116122	Digital Imaging Techniques and Analysis Lab	--	--	--	--	25	25	50

Lab Objectives:

1. To convert images between different color models analyze image characteristics through spatial and frequency domain transformations
2. To provide hands-on experience in implementing and analyzing image enhancement and filtering techniques for improving image quality in both spatial and frequency domains.
3. To convert images between different color models and analyze image characteristics.
4. To implement various image segmentation and morphological processing techniques using suitable tools such as Python.
5. To implement image compression or restoration algorithm
6. To apply texture and keypoint feature extraction techniques and implement classification and matching algorithms for effective image analysis and object recognition.
7. To perform image classification using a pre-trained deep learning model.

Lab Outcomes:

1. Apply image intensity transformation techniques to convert a color image into its grayscale equivalent and analyze the pixel intensity distribution using histogram representation.
2. Implement and analyze image enhancement and filtering techniques to improve image quality in spatial and frequency domains.
3. Convert images between different color models and analyze image characteristics.
4. Students will be able to apply edge detection, thresholding, and morphological operations to segment, analyze, and extract objects from digital images.
5. Students will be able to extract texture and keypoint features, perform classification and feature matching, and evaluate their effectiveness in image analysis and object recognition tasks.
6. Students understand how transfer learning enables image classification without building a model from scratch.

DRAFT

Suggested List of Experiments.

Sr No	List of Experiments	Hrs
01	To convert a color image into grayscale and plot its histogram.	02
02	To represent color images using different color models like and analyze spatial and frequency domain transformations	02
03	Implement techniques using a coding environment (e.g., Python with OpenCV). Observe the effect of and use Image Histograms for analysis and manipulation (equalization and specification). Point Processing Techniques, Zooming.	02
04	Create and apply simple frequency-domain filters (low-pass and high-pass) and compare them with spatial filters.	04
05	Add different types of noise to a clean image and use filters to remove it.	02
06	Use of Enhancement technique for Steganography.	02
07	Implementation of mean and median filters for noise removal.	02
08	Image restoration using Wiener filter in MATLAB/Python.	02
09	Image compression using JPEG (DCT-based) and calculation of PSNR.	02
10	Edge detection using Sobel/Prewitt/Canny operators (one of them may be performed or all 3 can be performed and a comparative analysis could be made)	02
11	Segmentation using Global and Adaptive Thresholding methods (may be separated as 2 experiments)	02
12	Application of Morphological Operations (Erosion, Dilation, Opening, Closing) for Noise Removal and Object Extraction	02
13	Implement statistical texture analysis technique such as Gray-Level Co-occurrence Matrix (GLCM) to extract texture features from images.	02
14	Implement feature matching between images using Scale-Invariant Feature Transform (SIFT) descriptors.	02
15	Image Classification using a Pre-Trained CNN Model	02
16	Object Detection using a Pre-Trained Model	02

Suggested Assignments:

Two Assignments Which Covered all LO's
(Assignment no 1 (LO1, LO2, LO3) + Assignment no 2 (LO4, LO5, LO6)

Text Books:

1. Rafael C. Gonzalez and Richard E. Woods, Digital Image Processing, 4th Edition, Pearson Education, 2018
2. Anil K. Jain, Fundamentals of Digital Image Processing, Prentice Hall of India, 2010.
3. B. Chanda and D. Dutta Majumder, Digital Image Processing and Analysis, PHI Learning, 2011.

References:

1. Kenneth R. Castleman, Digital Image Processing, Pearson Education, 2015.
2. Sridhar, S., Digital Image Processing, Oxford University Press, 2011.
3. Adrian Rosebrock, Deep Learning for Computer Vision with Python, PyImageSearch, 2019.

DRAFT

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.

DRAFT

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		TH	Pract.	Tut.	Theory	Pract.	Tut.	Total
2116123	Natural Language Processing 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					
2116123	Natural Language Processing Lab	--	--	--	--	--	25	25	50

Prerequisites:	Python Programming Language.
Lab Objectives:	1. To understand and implement fundamental text preprocessing and tokenization. 2. To explore language modeling techniques and smoothing strategies. 3. To perform morphological analysis using stemming and lemmatization. 4. To understand syntactic analysis using tagging and parsing techniques. 5. To apply semantic analysis and document similarity techniques. 6. To develop end-to-end NLP applications using machine learning and deep learning.
Lab Outcomes:	Upon completion of the course, the learners will be able to: 1. Apply tokenization and regular expressions to preprocess and clean raw text. 2. Construct unigram and bigram models and evaluate with smoothing techniques. 3. Implement morphological transformations of input text. 4. Develop POS taggers, CFG parsers, and dependency parsers for sentence structure analysis. 5. Use WordNet, TF-IDF, and cosine similarity to extract semantics and document distances. 6. Build text classifiers and fine-tune BERT models for NLP tasks using HuggingFace.

Declaration:
The Subject In-charges of all affiliated colleges are requested to conduct any seven (7) experiments from the **LO1 to LO5**, but ensure at least one experiment of each LO. Additionally, teachers may

DRAFT

select any three (3) experiments from the last module, ensuring that a minimum of ten (10) experiments are completed as part of the curriculum.
The following experiments serve as samples to illustrate the application of concepts covered in each unit. Instructors are encouraged to modify and adapt these experiments to meet the specific needs of the course and the learning objectives. It is essential to ensure that the fundamental concepts and skills outlined in each unit are adequately covered, even with modifications.

List of Experiments.

Sr. No.	Experiment Details/Description	Hrs	LO
1	Apply Text Preprocessing Techniques: Tokenization, Stop word Removal, Stemming, and Lemmatization Using NLTK	2	LO1
2	Write a Python program that uses Regular Expressions to extract all valid email addresses and Indian mobile numbers from a given unstructured text document containing mixed information such as names, dates, and random numeric sequences. The program should correctly identify patterns that match: Email format: username@domain.extension and the required Mobile number format. The extracted results should be printed separately as two well-formatted lists.	2	LO1
3	Evaluate N-gram Models with Laplace and Good-Turing Smoothing to Optimize Text Prediction Accuracy. **OR** Develop an N-gram Based Predictive Text Model for Daily Messaging Applications Using Laplace Smoothing.	2	LO2
4	Compare Text Classification Performance of Naive Bayes using Bag-of-Words versus TF-IDF for Spam Detection. **OR** Develop a Naïve Bayes Spam Classifier Using TF-IDF with POS Tagging Assistance for Social Media or Email Messages	2	LO2
5	Investigate Morphological Parsing using the Porter Stemmer and FST to Improve Document Search Relevance	2	LO2
6	Marathi and Hindi are highly inflectional languages; one-word form represents diverse grammatical features (gender, tense, plurality), often confusing search engines and chatbots. Our goal is to address this functional failure caused by morphological complexity. Students will develop and test an FST-based morphological parser capable of reliably identifying stems and affixes, ensuring practical usability across smart digital applications.	2	LO2
7	Apply Hidden Markov Model and Viterbi Decoding to Resolve Part-of-Speech Ambiguity for Text Chunking.	2	LO2
8	Implement Context-Free Grammars and Generate Parse Trees for Programming Language Constructs Using NLTK	2	LO3
9	Implement (as well as compare) different rule-based parsing algorithms (Top-Down, Bottom-Up, and CYK) in order to assess their efficiency, applicability, and limitations in handling syntactic ambiguity.	2	LO3
10	Apply statistical and dependency parsing methods using NLP tools and evaluate their effectiveness in understanding syntactic relations among words.	2	LO3
11	Apply neural parsing techniques for extracting syntactic and dependency structures from software documentation and bug reports, enhancing intelligent text analysis in engineering applications.	2	LO3

DRAFT

12	Lexical Semantics and Meaning Representation using WordNet : To understand meaning representation and explore lexical relations such as synonymy, antonymy, hyponymy, and meronymy using WordNet for given sets of words and sentences.	2	LO4
13	Word Embedding Generation using Word2Vec and GloVe: To generate and visualize word embeddings using Word2Vec (CBOW and Skip-gram models) and GloVe, and compare their semantic similarity performance.	2	LO4
14	Word Sense Disambiguation (WSD) Techniques: To implement supervised and unsupervised WSD approaches for resolving word sense ambiguity in given text samples. **OR** Words used in daily news (e.g., *bank*, *charge*, *virus*) often carry multiple meanings depending on context. Incorrect interpretation leads to misinformation in automated systems such as news summarizers, search engines, and sentiment analyzers. The problem is to design and evaluate WSD models that correctly predict the intended sense of ambiguous words in real news datasets. Analyze and implement Word Sense Disambiguation (WSD) using both supervised and unsupervised methods, and evaluate their performance on real-world news articles.	2	LO4
15	**Text Document Representation and Entity Recognition:** To represent text documents using TF-IDF and Doc2Vec models and apply Named Entity Recognition (NER) techniques to identify key entities in the text. **OR** NER is the task of finding and classifying names in text into types such as location, person, and so on. This makes it similar to word sense disambiguation. While the goal of word sense disambiguation is to predict the exact meaning of a word in a given context (often based on a knowledge base of words and their known meanings or senses —one such example is the WordNet), NER aims to achieve a coarse sense disambiguation of sorts. In the case of word sense disambiguation (WSD), every word has its own set of possible senses. NER aims to identify entities and classify them as a set of predetermined categories. These categories may be place or person in a general domain corpus. So, the goal of NER would be to identify that 'Sachin Tendulkar' is a person and 'Maharashtra' is a place in the sentence 'Sachin Tendulkar was born in Maharashtra'. Design, implement, and evaluate an NER model to automatically extract and classify person and location entities from text. **OR** Smart cities receive large volumes of unstructured citizen feedback (e.g., social media complaints, e-governance portals). Extracting key entities like *road names, public facilities,* and *municipal authorities* is critical for automated issue routing and decision-making. The task is to create an NER system using contextual document representation that accurately detects domain-specific entities. Design, implement, and evaluate an NER model that automatically identifies location, organization, and person names from citizen complaint data to support smart city services.	2	LO4
16	**Implementation of Reference Resolution Techniques:** Perform reference resolution on sample text and identify noun phrase references using rule-based strategies.	**2**	LO5
17	**Pronoun Resolution using Hobbs Algorithm:** To implement Hobbs Algorithm for anaphora resolution and evaluate its accuracy in resolving pronouns within selected paragraphs or short documents.	**2**	LO5

DRAFT

18	**Machine Translation for Multilingual Text Understanding:** To implement and analyze machine translation techniques for converting text from one language to another while preserving meaning and fluency, highlighting the need for automated translation in multilingual environments.	**2**	**LO6**
19	Sentiment Analysis (SA) identifies whether a text expresses a positive, negative, or neutral opinion. In its Boolean form, it classifies statements as positive or negative, such as "We enjoyed the rollercoaster ride" and the Hindi sentence "Samose khaake bada mazaa aya," both positive. A three-class version adds a neutral category for factual text like "Nayi Dilli Bharat ki raajdhani hai." Negative opinions, for example, "Mujhe yaha ka khana achha nahi lagta" (I don't like the food here), are labeled as negative. **Sentiment Analysis for Polarity and Rating Prediction:** To develop and evaluate a model that automatically classifies text sentiment as positive, negative, neutral, or into fine-grained rating categories.	**2**	**LO6**
20	Question Answering (QA) systems rely heavily on an initial Document Retrieval stage to narrow down the vast "Source of Information" into a manageable set of "potentially relevant documents." The efficiency and accuracy of this stage directly impact the quality of the final answer. The challenge is to determine which common document retrieval strategy -a classical, interpretable method like TF-IDF, or a modern, contextualized embedding-based approach yields superior performance (measured by metrics like MRR or F1 score) within a constrained lab environment, particularly given a limited size corpus and a set of open-domain questions. Evaluate the Impact of Document Retrieval Strategy on Question Answering System Performance.	**2**	**LO6**
21	**Automatic Text Summarization Using Extractive and Abstractive Techniques:** Large volumes of digital text such as news, articles, and reports make manual summarization time-consuming, inconsistent, and prone to information loss. The challenge is to develop an automatic text summarization system that generates concise and meaningful summaries while preserving key information from the original document. The system should analyze and compare different summarization techniques, such as extractive and abstractive methods, to improve information retention, readability, and relevance in the generated summaries. So, implement and compare extractive and abstractive summarization methods for generating concise and meaningful summaries.	**2**	**LO6**
22	Implement the self-attention mechanism and its core components (queries, keys, values, multi-head attention) from scratch and demonstrate how they encode contextual relationships in token sequences.	**2**	**LO6**
23	Build an encoder–decoder Transformer prototype with positional encodings, train it on a small seq2seq task and assess the role of positional information in sequence modeling.	**2**	**LO6**
24	Fine-tune a pre-trained model (e.g., BERT/GPT/T5) on a domain-specific dataset, compare performance across models, and evaluate outputs using BLEU, ROUGE, and F1 metrics for selected tasks (summarization, classification, or generation).	**2**	**LO6**

DRAFT

Sr. No	List of Assignments / Tutorials
01	Based on the Modules all modules of the Natural Language Processing.
02	Based on all the Self-learning topics from the Natural Language Processing modules.

Text Books:	1. T. Siddiqui and U. Tiwari, Natural Language Processing and Information Retrieval, Oxford University Press. 2. Pushpak Bhattacharyya and Aditya Joshi, Natural Language Processing, Wiley India Pvt. Ltd., Edition: 2023 3. Daniel Jurafsky. James H. Martin, Speech and Language Processing, An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models, Third Edition (draft) Aug 2025
References Books :	1. Bird, S., Klein, E., & Loper, E. *Natural Language Processing with Python*. O'Reilly. 2. Palash Goyal, Sumit Pandey, Karan Jain. *Deep Learning for Natural Language Processing*-2018 3. Christopher D.Manning and HinrichSchutze, Foundations of Statistical Natural Language Processing, MIT Press, 1999.
Online References:	https://onlinecourses.nptel.ac.in/noc23_cs45/preview https://archive.nptel.ac.in/courses/106/106/106106211/ https://www.nltk.org/book/ https://web.stanford.edu/class/cs224n/

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.

DRAFT

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2116124	Augmented and Virtual Reality 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					
2116124	Augmented and Virtual Reality Lab	--	--	--	--	--	25	25	50

Prerequisites:	Python Programming Language.
Lab Objectives:	1. To understand and apply AR/VR/XR platforms and XR SDK configuration in Unity. 2. To implement AR tracking techniques for simple interactive scenes. 3. To design and develop basic AR/VR applications with interactive features. 4. To create and optimize 3D assets for immersive XR environments. 5. To design and evaluate XR interactions for usability and ergonomics. 6. To integrate multiple skills to build a functional XR prototype for real-world applications.

DRAFT

Lab Outcomes:	Upon completion of the course, the learners will be able to: 1. Apply knowledge of AR/VR/XR platforms by configuring Unity with relevant XR SDKs. 2. Configure XR plugins and apply AR tracking in simple scenes. 3. Design and develop simple AR/VR applications with basic interactions. 4. Create an optimized and immersive XR scene using custom 3D assets. 5. Design and evaluate XR interactions for usability and ergonomics. 6. Create a functional XR prototype addressing a real-world application.

Declaration:
Subject In-charges are requested to conduct any eight experiments from LO1 to LO5, ensuring at least one per LO, and choose two from the final module, completing a minimum of ten experiments. The listed experiments are samples; teachers may modify them as needed while ensuring core concepts and skills of each unit are covered.

List of Experiments

Sr. No.	Experiment Details/Description	Hrs	LO
1	Comparative study of AR vs VR apps (test existing apps).	**2**	**LO1**
2	Introduction to Unity Interface and Scene Setup: Exploring Unity layout, GameObjects, Components, and C# scripting for XR.	**2**	**LO1**
3	Configuring XR SDKs and Building Environment: Setup of ARCore/ARKit/OpenXR plugin for Unity and deployment on mobile devices. **OR** Storyboard of an AR/VR app idea **OR** Study different XR devices and write a report.	**2**	**LO1**
4	Marker-Based AR Application: Implementing simple AR using Vuforia/AR Foundation with image-based marker detection and object rendering. **OR** Setup Unity with OpenXR and run a sample VR scene.	**2**	**LO2**
5	Markerless AR (Plane Detection): Implementing AR object placement using plane detection and anchors in Unity with ARCore/ARKit. **OR** Build a marker-based AR app using Unity + ARCore/ARKit	**2**	**LO2**
6	Create a VR scene with teleportation & object interaction.	**2**	**LO2**
8	VR Scene Creation: Designing a virtual environment with 3D assets, camera setup, and basic teleportation navigation.	**2**	**LO2**
9	Import & optimize a 3D model in Unity	**2**	**LO3**
10	Photogrammetry capture with smartphone & integrate in Unity	**2**	**LO3**
11	Interaction in VR: Implementing object grabbing, scaling, and rotation using controller or mouse input.	**2**	**LO3**
12	Integrating Audio and Lighting Effects in XR Scene: Adding ambient sounds, directional lighting, and shadows to enhance immersion.	**2**	**LO4**
13	3D Model Import and Optimization: Importing and optimizing FBX/OBJ/glTF models from Sketchfab or Asset Store for performance in XR.	**2**	**LO4**
14	Design a gaze-based interaction prototype in Unity	**2**	**LO4**
15	Conduct usability testing of an XR app & document feedback.	**2**	**LO4**
16	Build a multi-user VR/AR demo using networking packages	**2**	**LO5**
17	Evaluating Usability and Comfort: Observing motion sickness, latency, and ergonomics through user testing in a sample XR application.	**2**	**LO5**
18	**Development of an Immersive XR-Based Virtual Showroom:** Traditional showrooms require space and resources, limiting accessibility and product	**2**	**LO6**

DRAFT

	exploration. Customers struggle to inspect items closely before purchasing. This project aims to develop an immersive XR-based virtual showroom that enables users to navigate, inspect, and interact with products in realistic 3D. Features like customization, scaling, and placement will improve engagement and provide a convenient, constraint-free shopping experience. To design and develop an immersive XR-based virtual showroom that enables users to interact with, explore, and customize products in a realistic 3D environment.		
19	**Development of an AR-Based Navigation System:** Traditional maps and GPS often struggle to provide intuitive, real-time guidance in complex indoor or outdoor environments, making navigation and route visualization difficult. This project aims to develop an AR-based navigation system that overlays directional cues, points of interest, and route information onto the real-world view, enhancing spatial awareness, simplifying wayfinding, and delivering an interactive, context-aware experience. To design and develop an AR navigation system providing real-time guidance and visual cues for improved wayfinding.	**2**	**LO6**
20	**Development of an XR-Based Educational Simulation System:** Traditional teaching methods often struggle to explain complex concepts in subjects like science, engineering, and medicine, limiting student engagement and understanding. Physical labs or field visits may be costly, unsafe, or inaccessible. This project aims to develop an XR-based educational simulation that provides immersive, interactive learning experiences, allowing students to explore, manipulate, and understand concepts in a safe, engaging, and realistic virtual environment. To design and develop an XR-based educational simulation that enhances learning by providing interactive, immersive, and safe experiences for complex concepts.	**2**	**LO6**

Text Books:	1. Ralf Doerner, (2024). Virtual and Augmented Reality (VR/AR). Springer. 2. Simeone, A. (Ed.). (2024). Everyday Virtual and Augmented Reality. Springer. 3. Tacgin, Z. (2024). Virtual and Augmented Reality: An Educational Handbook. Cambridge Scholars Publishing. 4. Schmalstieg, Dieter, and Tobias Hollerer (2016) Augmented reality: principles and practice. Addison-Wesley Professional
References Books :	1. Chen, J. Y. C., & Fragomeni, G. (Eds.). (2024). Virtual, Augmented and Mixed Reality. Springer. 2. Kimberly Burk Cordova (2024). Augmented and Virtual Reality: Unlocking Business Potential for Entrepreneurs. 3. Chetankumar G Shetty, "Augmented Reality- Theory, Design and Development", Mc Graw Hill, 2020 Edition. 4. Virtual and Augmented Reality An Educational Handbook By Zeynep Tacgin
Online References:	https://nptel.ac.in/courses/106103842 https://nptel.ac.in/courses/106106138 https://nptel.ac.in/courses/106106699 Unity Learn: Create with VR

Sr. No	List of Assignments / Tutorials
01	Based on the Modules all modules of the Augmented and Virtual Reality.
02	Based on all the Self-learning topics from the Augmented and Virtual Reality.

DRAFT

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.

DRAFT

rse Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2116125	Blockchain Technology Lab	-	2	-	-	1	-	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				
2116125	Blockchain Technology Lab	--	--	--	--	25	25	50

Lab Objectives:
1. To develop hands-on expertise in cryptographic primitives essential for blockchain security, including hashing, ECDSA signatures, and Merkle structures.
2. To enable learners to simulate blockchain components such as blocks, transactions, consensus, and proof-of-work mechanisms using Python/JavaScript.
3. To train students in designing, compiling, testing, and deploying smart contracts using Solidity, Hardhat, and Remix on local and public Ethereum test networks.

4. To provide practical skills in creating Web3-enabled decentralized applications (DApps) using modern libraries like Ethers.js, Web3.js, and MetaMask.
5. To introduce enterprise blockchain frameworks like Hyperledger Fabric and enable learners to set up networks and deploy chaincode.
6. To guide students in end-to-end development of a mini-project involving smart contracts, tokens/NFTs, wallet integration, frontend UI, and deployment on test networks.

Lab Outcomes:

1. Implement and evaluate cryptographic primitives like hashing, ECDSA, Merkle tree structures.
2. Develop smart contracts using Solidity with industry-standard tools (Hardhat/Truffle/Remix).
3. Deploy and test smart contracts on local blockchain environments and Ethereum testnets.
4. Design custom cryptocurrency tokens and NFTs and perform on-chain interactions via wallets.
5. Develop DApps integrating smart contracts with frontend frameworks (React/Next.js).
6. Build, test, and deploy a complete DApp or enterprise blockchain application using Ethereum or Hyperledger Fabric.

Prerequisite:

DETAILED SYLLABUS:

Sr. No.	Module	Detailed Content	Hours	LO Mapping
0	**Prerequisite**	Overview of Blockchain, Hashing, Digital Signatures - Programming Basics for Smart Contracts and DApps (Python/JS) - Self-Learning: Intro to Web3, EVM, MetaMask	2	
I	**Cryptography & Blockchain Foundations**	SHA-256, KECCAK-256, ECDSA key generation, signing and verification, Merkle Trees, Transaction Hashing, Simulation of Blocks and Chain Linking **Mini Project Activity:** Group formation **Self-Learning:** Consensus Algorithms, Block Structure	2	LO1
II	**Smart Contracts with Solidity**	Solidity Language Basics, Contract Structure, Data Types, Events, Modifiers, Writing First Smart Contract, Chaincode Overview (Go/Node) for Hyperledger. **Mini Project:** Title Submission **Self-Learning:** Error handling, security practices	4	LO2
III	**Contract Deployment & Testing**	Using Remix IDE, Hardhat/Truffle workflow, Ganache Local Blockchain, Ethereum Testnets (Sepolia/Holesky), Wallet Integration (MetaMask), Unit testing using Hardhat/Truffle **Self-Learning:** Gas fees, transaction lifecycle, Fabric setup	4	LO3
IV	**Tokens, NFTs & IPFS**	ERC-20 Token Development, ERC-721 / ERC-1155 NFT Standards, Minting,	4	LO4

DRAFT

		Transferring & Metadata, Storing Assets via IPFS/Pinata. **Self-Learning:** Tokenomics, NFT marketplaces		
V	**DApp Development & Web3 Integration**	DApp Architecture, Ethers.js, Web3.js integration, Connecting to Wallets & Networks, Reading/Writing Smart Contract Data, Frontend using React/Next.js. **Self-Learning:** Hyperledger Fabric Enterprise DApps	4	LO5
VI	**Mini-project on Design and Development of a DApps using Ethereum/Hyperledger Fabric**	Implementation of Mini Project 1. Design, configure and testing of mini project 2. Report submission as per guidelines	6	LO6

Text Books:

1. Ethereum Smart Contract Development, Mayukh Mukhopadhyay, Packt publication.
2. Solidity Programming Essentials: A Beginner's Guide to Build Smart Contracts for Ethereum and Blockchain, Ritesh Modi, Packt publication. University of Mumbai, B. E. (Information Technology), Rev 2016 279
3. Hands-on Smart Contract Development with Hyperledger Fabric V2, Matt Zand, Xun Wu and Mark Anthony Morris, O'Reilly.

References:

1. Mastering Blockchain, Imran Bashir, Packt Publishing
2. Introducing Ethereum and Solidity, Chris Dannen, APress.
3. Hands-on Blockchain with Hyperledger, Nitin Gaur, Packt Publishing.

Online Resources:

Sr. No.	Website Name
1.	https://trufflesuite.com/
2.	https://portfolio.metamask.io/
3.	https://remix.ethereum.org/
4.	https://www.hyperledger.org/use/fabric

List of Experiments.

Sr No	List of Experiments	Hrs
01	Implement Cryptographic Hash Functions.	02
02 (Any one Experiment)	Generate keypairs using ECDSA.	02
	Implement a mini wallet-like signing script.	
	Construct a Merkle tree from transactions.	
	Simulate a Blockchain in Python/JS, add proof of work or Nonce.	
03	Implement smart contracts on local Blockchain network using Truffle.	02
04	Implement smart contracts on local Ethereum test network using Ganache.	02

DRAFT

05	Create Smart Contract using Solidity and Remix IDE for Balance Transfer.	02
06	Perform Embedding of wallet by creating transactions using Smart Contract using Solidity and Remix IDE for Balance Transfer.	02
07	Implement an ERC-20 Token Contract, perform transaction and view in MetaMask.	02
08	Build a DApp Frontend Using Web3.js/Ethers.js	04
09	Store and Retrieve Files Using IPFS	02
10	Mini Project	06

Sr No	List of Assignments / Tutorials	Hrs
01	Comparative study of all hashing algorithms	
02	Write Solidity data types and special built in variables like msg, block, etc. and their properties	
03	List different ways to compile solidity file and generate ABI representations	
04	Explore Ethereum Explorer and find current block height, difficulty level and number of transaction in last but one block.	

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 (5Experiment+10 Mini Project) + 5 Marks (Assignments) + 5 Marks (Attendance)
Practical& Oral Exam: An Oral & Practical exam will be held based on the above syllabus.

DRAFT

Course Code	Course Name	Teaching Scheme (Contact Hours)			Credits Assigned			
		Theory	Pract.	Tut.	Theory	Pract.	Tut.	Total
2116126	User Experience Design 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				

2116126	**User Experience Design Lab**	--	--	--	--	**25**	**25**	**50**

Lab Objectives:

1. To apply the principles of user-centered design in creating wireframes and prototypes.
2. To develop interactive interfaces using Figma and other open-source UX design tools.
3. To integrate usability evaluation and feedback loops into the design process.
4. To explore AI-assisted UX workflows for generating and refining design ideas.
5. To design accessible and responsive user interfaces following usability and accessibility standards.
6. To collaborate effectively in teams for prototyping, testing, and delivering user interface solutions.

Lab Outcomes:

1. Apply the user-centered design process to develop low and high-fidelity wireframes.
2. Use open source tools to design, prototype, and test user interfaces effectively.
3. Evaluate usability and accessibility through heuristic analysis and user testing.
4. Integrate AI-powered tools to accelerate ideation, prototyping, and interaction design.
5. Improve user interfaces using usability, heuristic, rapid and emotional design evaluations
6. Demonstrate collaborative design practices and deliver functional prototypes for real-world applications.

Prerequisite: Design Thinking

List of Experiments:

Sr No	List of Experiments (At least 8 to 10)	Hrs
01	**Understanding UX Tools and Interface Setup**: Explore Figma interface, workspace setup, design grids, and basic tools.	2
02	**Creating Wireframes for a Simple App**: Develop low-fidelity wireframes for a mobile or web app following user-centered design steps.	2
03	**Building Layouts and UI Components**: Create reusable components, apply styles, and use auto-layout for responsiveness.	2
04	**Prototyping User Flows**: Create interactive prototypes and navigation links between screens; simulate user journeys.	2
05	**Heuristic Evaluation and Usability Testing**: Perform usability evaluation using Nielsen's heuristics and record observations.	2
06	**Accessibility Design Check**: Check color contrast, typography legibility, and navigation accessibility.	2
07	**Collaborative Design Using Figma Teams**: Work on shared projects and handoff assets to developers using Inspect panel.	2
08	**AI-Assisted Wireframe Generation**: Use AI tools like Uizard or Galileo AI to generate UI layouts from text prompts; refine in Figma.	2
09	**Prototype Testing and User Feedback Integration**: Conduct simple usability testing, record feedback, and iterate design.	2
10	**Mini Project : Mini Project will be based on contents from the syllabus. Design prototype will be using Open source UX tool and implementation by using Open Source tools.**	-

Note: Any open-source tool can be used to design the above experiments.

Assessment :

Term Work: Term Work shall consist of at least 6 to 8 practicals' based on the above list & Project. Also, Term work Journal must include at least 2 assignments.

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

DRAFT

+ 5 Marks (Attendance)

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

DRAFT

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

DRAFT