Shree Rahul Education Society's (Regd.)

SHREE L. R. TIWARI COLLEGE OF ENGINEERING

An Autonomous Institute Affiliated to University of Mumbai,

Approved by AICTE & DTE, Maharashtra State | NAAC Accredited, NBA Accredited Program | ISO 9001:2015 Certified |
DTE Code No: 3423 | Recognized under Section 2(f) of the UGC Act 1956 | Minority Status (Hindi Linguistic)

Shree Rahul Education Society's (Regd.)

Shree L.R. Tiwari College of Engineering

Shree L.R. Tiwari Educational Campus,

Mira Road–East, Thane-401107.

Maharashtra, India

www.slrtce.in

An Autonomous Institute Affiliated to the University of Mumbai

Department of Computer Engineering

Curriculum TY A.Y 2026-27

Third Year Syllabus

Revision: R-2025

PREAMBLE

"Think. Design. Innovate. – Learning that Leads the Future."

SLRTCE is proud to be among the youngest institutions to attain academic autonomy—an achievement that strengthens our commitment to becoming a world-class centre for socio- economic development and quality education. It is with great pride and purpose that we introduce the restructured curriculum under academic autonomy at SLRTCE. This initiative marks a significant step toward transforming engineering education by equipping our graduates with strong technical foundations, research acumen, interdisciplinary perspective, and an innovation-driven mindset essential for success in today's evolving professional landscape.

1. **Curriculum Design:** Theme-Based, Experiential, and Outcome-Oriented The autonomous curriculum adopts a bottom-up, learner-centric approach aligned with all 12 Program Outcomes. Each semester is driven by a central societal or technological theme, with problem statements rooted in local needs, socio-economic priorities, and the UN Sustainable Development Goals (UNSDGs). Students engage with these challenges through individual projects in the IDEA Lab, integrating social science methodologies with engineering solutions to foster innovation, contextual learning, and socially responsible problem-solving.

2**. Purpose of Autonomy:** Empowering Innovation and Real-World Learning Our pursuit of autonomy is rooted in bridging the gap between academic learning and industry relevance. By integrating undergraduate research through the IDEA Lab—our dedicated innovation and problem-solving hub—students engage with real-world challenges, apply interdisciplinary knowledge, and develop socially impactful solutions. This approach nurtures engineers who are not only technically sound but also responsible and creative thinkers.

3. **Alignment with National Education Policy-2020:** Aligned with NEP 2020, the curriculum emphasizes multidisciplinary, hands-on learning. Through theme-based projects, electives, value-added courses, and flexible assessments, SLRTCE ensures holistic student development driven by real-world application and academic rigor. SLRTCE, as a part of Rahul Education, benefits from being in a network of HEIs. All Higher Education Institutions under Rahul Education can collectively contribute to curriculum development, promoting multidisciplinary studies.

4. **Innovative Pedagogy Initiative by IQAC As part of its transformation strategy**, IQAC has introduced an innovative pedagogy model centred on interdisciplinary research and contextual problem-solving. The IDEA Lab enables students to apply subject knowledge through socially relevant projects, supported by design thinking and faculty mentorship. A two-tier assessment framework—covering both project execution and subject-wise application—ensures academic depth, continuous mentoring, and quality assurance.

5. **Empowering Faculty as Mentors and** Innovators: SLRTCE believes that the success of academic autonomy lies in the hands of its faculty. By empowering educators to think beyond conventional boundaries, we enable them to instil Knowledge, Skills, and Attitudes (KSA) in students while enriching their own professional growth. Faculty serve as research mentors and project guides, supported through structured training in innovation pedagogy and assessment**.**

Sincerely,

Principal
SLRTCE

COO
Rahul Education

Preface (BOS)

Dear Stakeholder,

It is an honour to present the revised Bachelor of Technology (Computer Engineering) curriculum our first under SLRTCE's recently granted Autonomous status. The programme aligns with the National Education Policy 2020, which mandates a four-year multidisciplinary engineering framework, multiple entry-exit options and a semester credit band for all autonomous engineering colleges.

This curriculum is directed through a coherent academic pathway that includes the **Master of Technology in Computer Engineering** and **B.Voc programmes Cyber Security & Digital Forensics** and **Big Data & Cloud Computing** ensuring learners can progress from foundational study to specialised expertise in high-growth domains.

Each semester of the B.Tech. programme is built around real-world problem statements mapped to the United Nations Sustainable Development Goals. Students tackle these challenges through our IDEA Lab pipeline, moving from design thinking to prototyping and product development. This approach balances rigorous theory with hands-on practice while nurturing ethical leadership, social responsibility and entrepreneurial spirit.

Laboratory session, open electives, value-added courses, audit courses, and internships cultivate broad competence, while independent-study components reinforce lifelong learning. Field projects and community-engagement activities now embedded as credit-bearing requirements strengthen our industry-academic linkages and societal impact, exactly as the state directives envisage.

We extend sincere thanks to Respected Mr Utsav Tiwari, Chief Operating Officer of Rahul Education, and to our Respected Principal, Prof. Dr. Umesh S. Bhadade, for their strategic guidance. I am equally grateful to our faculty colleagues, industry experts, alumni and student representatives whose insights shaped this curriculum.

Your constructive feedback will enable us to uphold SLRTCE's vision of producing technically skilled, ethically grounded and socially conscious professionals equipped for the challenges of the twenty-first century.

Warm regards,

Dr Bhushankumar P. Nemade
Chairperson, Board of Studies, Computer Engineering SLRTCE

A. Abbreviations

AEC	Ability Enhancement Course
AECL	Ability Enhancement Course Laboratory
AU	Audit Course
BSC	Basic Science Course
BSCL	Basic Science Course Laboratory
CC	Co-Curricular Courses
CEP	Community Engagement Project
ELC	Experiential Learning Course
ESC	Engineering Science Course
ESCL	Engineering Science Course Laboratory
EMC	Entrepreneurship/economics/management courses
HSSM	Humanities Social Sciences and Management
IKS	Indian Knowledge System
INTR	Internship
L	Lecture
LC	Laboratory Course
LLC	Liberal Learning Course
MDM	Multidisciplinary Minor
MDML	Multidisciplinary Minor Laboratory
OE	Open Elective
OJT	On Job Training
P	Practical
PROJ	Project
PCC	Program Core Course
PCCL	Program Core Course Lab
PEC	Program Elective Course
RM	Research Methodology
SL	Self-Learning
T	Tutorial
VEC	Value Education Course
VSEC	Vocational and Skill Enhancement Course

Curriculum Structure and Examination Scheme for B.Tech. in Computer Engineering
(FY with Effect from AY 2026-2027) Curriculum Structure – FY Semester-I Group A

Course Code	Course Type	Course Name	Teaching Scheme (Contact Hours Per Week)			Teaching Scheme (Contact Hours Per Semester)					Total Credits (C) (Notional Learning Hour/30)	Examination Scheme				Exam Duration for Theory (in Hrs.)	Term Work	Practical & Oral	Total
												In-Semester Assessment			End Semester Exam (ESE)				
			L	T	P	L	T	P	SL	Notional Learning		IAT-I	IAT-II	Total					
12111101	BSC	Engineering Mathematics-I	3	1	--	45	15	--	60	120	4	20	20	40	60	2.5	25	–	125
12111102	BSC	Engineering Physics	2	--	--	30	--	--	30	60	2	20	20	40	60	2.5	–	–	100
12121103	ESC	Engineering Mechanics	3	--	--	45	--	--	45	90	3	20	20	40	60	2.5	–	–	100
12211104	PCC	Structured Programming Approach	2	--	--	30	--	--	30	60	2	20	20	40	60	2.5	–	–	100
12421105	AEC	Professional Communication & Ethics	2	--	--	30	--	--	30	60	2	20	20	40	60	2.5	–	–	100
12112106	BSC Lab	Engineering Physics Lab	--	--	1	--	--	15	--	15	0.5	–	–	–	–	–	25	–	25
12122107	ESC Lab	Engineering Mechanics Lab	--	--	2	--	--	30	--	30	1	–	–	–	–	–	25	–	25
12212108	PCCL	Internet technology Lab	--	--	2	--	--	30	--	30	1	–	–	–	–	–	25	–	25
12422109	AECL	Professional Communication & Ethics Lab	--	--	2	--	--	30	--	30	1	–	–	–	–	–	–	25	25
12122110	ESC Lab	IDEA Lab	--	--	2	--	--	30	--	30	1	–	–	–	–	–	25	50	75
12412111	VSEC	C Programming Lab	--	--	2	--	--	30	--	30	1	–	–	–	–	–	–	25	25
98462112	CC	Universal Human Values	1	--	1	15	--	15	15	45	1.5	–	–	–	–	–	25	–	25
Total			**13**	**1**	**12**	**195**	**15**	**180**	**210**	**600**	**20**	**100**	**100**	**200**	**300**	**12.5**	**150**	**100**	**750**

Curriculum Structure and Examination Scheme for B.Tech. in Computer Engineering (FY with Effect from AY 2026-2027) Curriculum Structure – FY Semester-II Group A

\Course Code	Course Type	Course Name	Teaching Scheme (Contact Hours Per Week)			Teaching Scheme (Contact Hours Per Semester)					Total Credits (C) (Notional Learning Hr/30)	Examination Scheme				Exam Duration for Theory (in Hrs.)	Term Work	Practical & Oral	Total
												In-Semester Assessment			End Semester Exam (ESE)				
			L	T	P	L	T	P	SL	Notional Learning		IAT-I	IAT-II	Total					
12111201	BSC	Engineering Mathematics-II	3	1	--	45	15	--	60	120	4	20	20	40	60	2.5	25	–	125
12111202	BSC	Engineering Chemistry	2	--	--	30	--	--	30	60	2	20	20	40	60	2.5	–	–	100
12121203	ESC	Basic Electrical & Electronics Engineering	3	--	--	45	--	--	45	90	3	20	20	40	60	2.5	–	–	100
12121204	ESC	Engineering Graphics	2	--	--	45	--	--	45	90	3	20	20	40	60	2.5	–	–	100
12211205	PCC	Digital Logic & Computer Organization (DLCO)	3	--	--	30	--	--	30	60	2	20	20	40	60	2.5	–	–	100
12112206	BSC Lab	Engineering Chemistry Lab	--	--	1	--	--	15	--	15	0.5	–	–	–	–	–	25	–	25
12122207	ESC Lab	Basic Electrical & Electronics Engineering Lab	--	--	2	--	--	30	--	30	1	–	–	–	–	–	25	–	25
12122208	ESC Lab	Engineering Graphics Lab	--	--	2	--	--	30	--	30	1	–	–	–	–	–	25	--	25
12212209	PCC LAB	Digital Logic & Computer Organization Lab (DLCO)	--	--	2	--	--	30	--	30	1	–	–	–	–	–	--	25	25
12122210	ESC Lab	IDEA Lab	--	--	2	--	--	30	--	30	1	–	–	–	–	–	25	50	75
12412211	VSEC	Python Programming	--	--	2	--	--	30	--	30	1	–	–	–	–	–	–	25	25
98442212	IKS	Indian Knowledge System	1	--	1	15	--	15	15	45	1.5	–	–	–	–	–	25	–	25
Total			**13**	**1**	**12**	**210**	**15**	**180**	**225**	**630**	**21**	**100**	**100**	**200**	**300**	**12.5**	**150**	**100**	**750**

Curriculum Structure and Examination Scheme for B.Tech. in Computer Engineering
(FY with Effect from AY 2026-2027) Curriculum Structure – TY Semester-V

Course Code	Course Type	Course Name	Teaching Scheme (Contact Hours Per Week)			Teaching Scheme (Contact Hours Per Semester)					Total Credits (C) (Notional Learning Hour/30)	Examination Scheme				Exam Duration for Theory (in Hrs.)	Term Work	Practical / Oral	Total
												In-Semester Assessment			End Semester Exam (ESE)				
			L	T	P	L	T	P	SL	Notional Learning		IAT-I	IAT-II	Total					
12211501	PCC	Theory of Computation	3	--	--	45	--	--	45	90	3	20	20	40	60	2.5	--	--	100
12211502	PCC	Computer Networks	3	--	--	45	--	--	45	90	3	20	20	40	60	2.5	--	--	100
12211503	PCC	Data ware Housing and Mining.	3	--	--	45	--	--	45	90	3	20	20	40	60	2.5	--	--	100
12212504	PCC LAB	Computer Networks lab	--	--	2	--	--	30	--	30	1	--	--	--	--	--	25	25	50
12212505	PCC LAB	Data ware Housing and Mining Lab	--	--	2	--	--	30	--	30	1	--	--	--	--	--	25	25	50
12212506	PCC LAB	IDEA LAB -5 (Innovation Design Engineering and Apply	--	--	2	--	--	30	--	30	1	--	--	--	--	--	25	25	50
1222157W	PEC-I	**Elective***	3	--	--	45	--	--	45	90	3	20	20	40	60	2.5	--	--	100
9931158WW	MDM-II	Multidisciplinary Minor	3	--	--	45	--	--	45	90	3	20	20	40	60	2.5	--	--	100
12412509	VSEC	**Open Source Technology**	--	--	2	--	--	30	--	30	1	--	--	--	--	--	25	--	25
99511510WW	OE-III*	Open Elective	2	--	--	30	--	--	30	60	2	--	--	--	--	--	25	--	25
Total			**17**	**–**	**8**	**285**	**–**	**90**	**285**	**630**	**21**	**100**	**100**	**200**	**300**	**12.5**	**125**	**2.5**	**700**

* Recommended that OE to be offered in the form of0020MOOCs (NPTEL/Swayam)

Program Elective Courses-I Semester V		
Sr. No.	**Course code**	**Name of course**
1	**12215701**	Internet of Things
2	**12215702**	Digital Communication
3	**12215703**	Artificial Intelligence
4	**12215704**	Object Oriented Modeling & Design
5	**12215705**	Digital Signal Processing

Open Elective Courses-III Semester V		
Sr. No.	**Course code**	**Name of course**
1	**9951151001**	International Financial Management
2	**9951151002**	Marketing Management - I
3	**9951151003**	HR Analytics
4	**9951151004**	Operations and Supply Chain Management
5	**9951151005**	Quantitative Investment Management
6	**9951151006**	Economics of Banking and Finance Markets
7	**9951151007**	Corporate Finance
8	**9951151008**	Principles of Behavioral Economics

Students who achieve a CGPI of 7.5 or higher by the end of the semester VI are eligible to pursue an additional 18 credits from the semesters V to VIII to qualify for a B. Tech degree with Honors.

Curriculum Structure and Examination Scheme for B.Tech. in Computer Engineering (TY with Effect from AY 2026-2027) Curriculum Structure – TY Semester-VI

Course Code	Course Type	Course Name	Teaching Scheme (Contact Hours Per Week)			Teaching Scheme (Contact Hours Per Semester)					Total Credits (C) (Notional Learning Hour/30)	Examination Scheme				Exam Duration for Theory (in Hrs.)	Term Work	Practical & Oral	Total
												In-Semester Assessment			End Semester Exam (ESE)				
			L	T	P	L	T	P	SL	Notional Learning		IAT-I	IAT-II	Total					
12211601	PCC	Compiler Construction	3	--	--	45	--	--	45	90	3	20	20	40	60	2.5	–	–	100
12211602	PCC	Cyber Security and Digital Forensics.	3	--	--	45	--	--	45	90	3	20	20	40	60	2.5	–	–	100
12211603	PCC	Machine Learning	3	--	--	45	--	--	45	90	3	20	20	40	60	2.5	–	–	100
12212604	PCC LAB	Cyber Security Lab	--	--	2	--	--	30	--	30	1	--	--	--	--	--	25	25	50
12212605	PCC LAB	Machine Learning Lab	--	--	2	--	--	30	--	30	1	--	--	--	--	--	25	25	50
12812606	Dissertation-I	Dissertation	--	--	2	--	--	30	--	30	1	--	--	--	--	--	50	50	100
1222167W	PEC-II	**Elective***	3	--	--	45	--	--	45	90	3	20	20	40	60	2.5	–	–	100
9931168WW	MDM-III	**Multidisciplinary Minor**	3	--	--	45	--	--	45	90	3	20	20	40	60	2.5	–	–	100
12451612	EMC/ VEC*	Professional Communication & Ethics	2	--	--	30	--	--	30	60	2	–	–	–	–	–	25	–	25
99511610WW	OE-IV*	**Open Elective-IV**	2	--	--	30	--	--	30	60	2	–	–	–	–	–	25	--	25
Total			**19**	**--**	**6**	**285**	**--**	**90**	**285**	**660**	**22**	**100**	**100**	**200**	**300**	**12.5**	**150**	**100**	**750**

* Recommended that OE, IKS, VSEC, CC to be offered in the form of MOOCs (NPTEL/Swayam)

Program Elective Courses- II Semester VI		
Sr. No.	**Course code**	**Name of course**
1	**12216701**	POSIX Programming
2	**12216702**	Advance Computer Network
3	**12216703**	Natural Language Processing
4	**12216704**	Parallel Algorithms
5	**12216705**	Image Processing

Open Elective Courses-IV Semester VI		
Sr. No.	**Course code**	**Name of course**
1	**9951161001**	Moral Thinking: Introduction to Values and Ethics,
2	**9951161002**	Ethics in Engineering Practice,
3	**9951161003**	Introduction to Psychology,
4	**9951161004**	Direct foreign language courses (like German, French),
5	**9951161005**	Financial Institutions and Markets,
6	**9951161006**	Investment Management
7	**9951161007**	Introduction to Sociology
8	**9951161008**	Cultural Studies Language for Competitive Exams

Note:

1. Questions based on self-learning must not be asked in internal assessment and End semester Exam

2. IDEA Lab: The IDEA Lab fosters individual interdisciplinary projects through theme-based, subject-integrated, experiential learning, assessed via standardized rubrics for innovation, application, and documentation.

3. Notional Learning Hours: In higher education, Notional Learning Hours (NLH) indicates the estimated time that a typical learner, at a specific academic level, is expected to invest in order to achieve the set learning outcomes of a course or module. This time includes a range of structured and independent learning activities, each contributing to the development of knowledge and competencies within a module

4. Self-Learning (SL): Self-learning is integrated into both theory and practical subjects, particularly where practical sessions have only one contact hour per week. It emphasizes student-driven efforts in preparing for assignments, practical, and examinations. Additionally, self-learning includes topics that go beyond the prescribed syllabus to align with current industry trends and requirements. Assessment of self-learning will follow a formative approach. For theory subjects that include term work, 30% of the assignment questions and student presentations/discussions should be based on self-learning topics. In subjects without term work, the corresponding laboratory sessions will incorporate self-learning, carrying a weightage of 30%. The subject with no term work and practical will have no assessment on self-learning topics. This may involve laboratory work using the latest tools or preparing technical write-ups/research papers on advanced topics.

SEMESTER- V

Semester V

Intelligent, scalable and inclusive platforms for Digital Empowerment and Sustainable Futures

(Harnessing Intelligent, Ubiquitous, and Sustainable Technologies to Build an Inclusive Digital Ecosystem for Mira Bhayandar and Beyond)
This theme blends intelligence (CN,BI, Web), and sustainability (Green IT) to focus on urban challenges (traffic, waste, e-health, cyber safety, energy use). It ensures that technology growth = sustainable + ethical + inclusive growth.
UN SDGs Alignment

- SDG 3: Good Health & Well-Being
- SDG 7: Affordable & Clean Energy
- SDG 9: Industry, Innovation & Infrastructure
- SDG 11: Sustainable Cities & Communities
- SDG 12: Responsible Consumption & Production
- SDG 13: Climate Action
- SDG 16: Peace, Justice & Strong Institutions

Theme : Urban System

1. **To study challenges in urban systems and demonstrate a very basic system for improved service efficiency.**
2. **To observe urban resource usage and demonstrate a very basic system for better coordination.**
3. **To study urban service practices and demonstrate a very basic system supporting sustainability.**
4. **To observe operational patterns in urban environments and demonstrate a very basic system for improvement.**
5. **To study urban infrastructure needs and demonstrate a very basic system for better management.**
6. **To study mobility-related challenges in urban environments and explore approaches that improve movement and accessibility.**
7. **To investigate the interaction between urban services and citizens and identify opportunities for improved service delivery.**
8. **To observe public service usage and demonstrate a very basic system for efficient operation.**
9. **To study urban system behavior and demonstrate a very basic system supporting organized functioning.**
10. **To observe service-related challenges in urban areas and demonstrate a very basic system for enhancement.**

Course Code	Course Name	Teaching Scheme (Contact Hours Per Week			Teaching Scheme (Contact Hours Per Semester)					Total Credits (C) (Notional Learning Hour/30
		L	T	P	L	T	P	SL	Notional Learning Hour	
12211501	Theory of Computation	3	-	-	45	-	-	45	90	3

Course Code	Course Name	Examination Scheme					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-I	IAT-II	Total					
12211501	Theory of Computation	20	20	40	60	2.5	--	--	100

Rationale :

The course **Theoretical Computer Science** provides fundamental knowledge of formal languages, automata, and computational models. It helps students understand how machines recognize languages and process information using models such as finite automata, pushdown automata, and Turing machines. The course develops analytical skills for designing automata and solving language recognition problems. It also introduces the limits of computation through concepts like decidability and undecidability. This foundation supports advanced areas such as compiler design, artificial intelligence, and algorithm analysis.

Course Objectives:

Describe the basic components and classifications of formal grammars and languages.

1. Explain the theoretical concepts behind deterministic and non-deterministic finite automata.
2. Design finite automata for specific language recognition problems.
3. Construct pushdown automata for context-free languages.
4. Compare different types of Turing machines and explain their real-world applications.
5. Interpret and evaluate the concept of undecidability in computational problems.

Course Outcomes:

1. Define and describe fundamental concepts of formal grammars, languages, and automata.
2. Explain the working principles of deterministic and non-deterministic finite automata and pushdown automata.
3. Construct finite automata and pushdown automata for given language specifications.
4. Differentiate between regular, context-free, and recursively enumerable languages using automata theory.
5. Evaluate the computational power and limitations of Turing machines in solving problems.
6. Design Turing machines and formal models to solve real-world computational problems, and justify their correctness.

Prerequisite: Discrete Structures

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
I	**Basic Concepts and Finite Automata**	Importance of TCS, Alphabets, Strings, Languages, Closure properties, Finite Automata (FA) and Finite State machine (FSM). Deterministic Finite Automata (DFA) and Nondeterministic Finite Automata (NFA): Definitions, transition diagrams and Language recognizers, Equivalence between NFA with and without ε- transitions, NFA to DFA Conversion, Minimization of DFA, FSM with output: Moore and Mealy machines, Applications and limitations of FA.	08	CO1

		Self-learning Topics:. Applications of Finite Automata in real-world systems (pattern matching, lexical analysis, control systems). Representation of languages using set notation and operations on languages (union, concatenation, Kleene star). Extended transition function (δ*) and string processing in DFA. Limitations of Finite Automata and motivation for more powerful models. Simulation and visualization of automata using tools like JFLAP.	08	
II	**Regular Expressions and Languages**	Regular Expression (RE), Equivalence of RE and FA, Arden 's Theorem, RE Applications Regular Language (RL), Closure properties of RLs, Decision properties of RLs, Pumping lemma for RLs.	07	CO2
		Self-learning Topics: Applications of Regular Expressions. Conversion of Regular Expression to Finite Automata. Conversion of Finite Automata to Regular Expression using Arden's Theorem. Applications of Regular Languages in computing. Verification of Closure Properties of Regular Languages. Decision Properties of Regular Languages. Pumping Lemma examples for proving non-regular languages. Practice of RE and FA using JFLAP	07	
III	**Grammars & Context Free Grammars (CFG)**	Grammars and Chomsky hierarchy: Regular Grammar (RG), Equivalence of Left and Right linear grammar, Equivalence of RG and FA. Definition, Sentential forms, Leftmost and Rightmost derivations, Parse tree, Ambiguity, Simplification and Applications, Normal Forms: Chomsky Normal Forms (CNF) and Greibach Normal Forms (GNF), Context Free language (CFL) - Pumping lemma, Closure properties.	08	CO3
		Self-learning Topics: Applications of Grammars and Chomsky Hierarchy in language classification. Understanding the Pumping Lemma for Context-Free Languages with examples. Conversion between Regular Grammar and Finite Automata.	08	

IV	**Pushdown Automata (PDA)**	Definition, Language of PDA, PDA as generator, decider and acceptor of CFG, Deterministic PDA, Non-Deterministic PDA, Application of PDA.	07	CO4
		Self-learning Topics: Real-world applications of Pushdown Automata (PDA) in parsing and language processing.. Simulation and practice of PDA design using JFLAP.	07	
V	**Turing Machine (TM)**	Definition, Design of TM as generator, decider and acceptor, Variants of TM: Multitrack, Multitape, Universal TM, Applications, Power and Limitations of TMs.	08	CO5
		Self-learning Topics: Applications of Turing Machines in modern computing theory**.** Discussion on computational power and limitations of Turing Machines**.** Simulation and visualization of Turing Machines using JFLAP.	08	
VI	**Undecidability**	Decidability and Undecidability, Recursive and Recursively Enumerable Languages, Halting Problem, Rice's Theorem, Post Correspondence Problem.	07	CO6
		Self-learning Topics: Difference between recursive languages and recursively enumerable languages**.** Overview and interpretation of Rice's Theorem with practical examples. Understanding the Post Correspondence Problem (PCP) and its role in proving undecidability.	07	

Text Books:

1. John E. Hopcroft, Rajeev Motwani, Jeffery D. Ullman, "**Introduction to Automata Theory, Languages and Computation"**, 3rd Edition, Pearson Education, 2008.

2. Michael Sipser, **"Theory of Computation"**, 3rd Edition, Cengage learning. 2013**.**

3. Vivek Kulkarni, **"Theory of Computation"**, Illustrated Edition, Oxford University Press, (12 April 2013) India.

References:

1. J. C. Martin, **"Introduction to Languages and the Theory of Computation"**, 4th Edition, Tata McGraw Hill Publication, 2013.

2. Kavi Mahesh, **"Theory of Computation: A Problem-Solving Approach"**, Kindle Edition, Wiley-India, 2011.

Online References:

1. https://nptel.ac.in/courses/106/104/106104028/
2. https://nptel.ac.in/courses/106/104/106104148/
3. www.jflap.org

Course Code	Course Name	Teaching Scheme (Contact Hours Per Week			Teaching Scheme (Contact Hours Per Semester)					Total Credits (C) (Notional Learning Hour/30
		L	T	P	L	T	P	SL	Notional Learning Hour	
12211502	Computer Networks	3	--	--	45	--	--	45	90	3

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration(in Hrs)			
		IAT-I	IAT-II	Total					
12211502	Computer Networks	20	20	40	60	2.5	--	--	100

Rationale :

This course equips students with foundational and advanced knowledge of computer networking principles, covering switching, routing, transport services, and application protocols. It strategically addresses the growing demand for secure, scalable, and efficient communication systems in modern industries. The curriculum aligns with NEP 2020 by fostering holistic, multidisciplinary education that integrates theory with hands-on practice. Skill-based learning is emphasized through protocol simulations, subnetting exercises, and case studies on real-world technologies like DNS, HTTP, and Ethernet. Experiential learning is achieved via labs and mini-projects that mirror industry scenarios. Importantly, the course contributes to UN Sustainable Development Goals such as Industry, Innovation, and Infrastructure (SDG 9), Sustainable Cities and Communities (SDG 11), and Quality Education (SDG 4) by preparing students to design resilient, inclusive, and future-ready networks. Guided by Bloom's Taxonomy, the pedagogy ensures cognitive development from basic understanding to application and evaluation. Graduates

will emerge as competent professionals capable of managing and innovating in the global digital economy.

Course Objectives:

1. To introduce the basic concepts and fundamental principles of data communication and computer networks.
2. To provide a comprehensive understanding of the structure and functions of the OSI reference model and its individual layers.
3. To examine the design challenges and operational aspects of the TCP/IP protocol suite.
4. To study and analyze different routing algorithms along with their comparative advantages and limitations.
5. To understand the functioning and significance of various transport layer protocols in ensuring reliable communication.
6. To explore application layer protocols and their role in enabling end-user network services.

Course Outcomes:

1. Demonstrate the fundamental concepts of data communication at the physical layer and compare the ISO-OSI model with the TCP/IP model.
2. Identify and explain key design issues and protocols at the data link layer.
3. Apply IP addressing techniques to design efficient network architectures using subnetting and supernetting.
4. Analyze transport layer functionalities, protocols, and evaluate various congestion control algorithms.
5. Explore and illustrate the role of common application layer protocols and services in real-world communication.
6. Integrate the concepts across all network layers to design and troubleshoot end-to-end network communication solutions.

Prerequisite: Linux Operating System.

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
I	**Introduction to computer networks**	Data Communication and Networking: A Communication Model, Data Communication, Evolution of Network, Network Components, Network Topology and Transmission Media, Protocol and Standards, Network Model (OSI, TCP/IP)	7	CO1
		Self-Learning Topics- Development of early computer networks (ARPANET), Types of networks	7	
II	**Circuit and Packet Switching**	Circuit Switching – Packet Switching – Digital and Analog Signals - Periodic Analog Signals - Transmission Impairments - Digital data transmission techniques - Analog data transmission techniques - Multiplexing and Spread Spectrum. -Introduction to Communication: Electromagnetic Spectrum. Guided Transmission Media: Twisted pair, Coaxial, Fiber optics.	8	CO2
		Self-learning Topics: Wireless Communication Technologies	8	
III	**Data Link Layer**	Error Detection and Correction - Parity - LRC - CRC - Hamming Code - Flow Control and Error Control - Stop and wait - ARQ - Sliding window - HDLC - Multiple Access Protocols - CSMA - CSMA/CD and CSMA/CA - IEEE 802.3 Ethernet.	8	CO3
		Self-learning Topics: Forward Error Correction (Reed–Solomon, Convolutional Codes)	8	

IV	**Network Layer**	Packet Switching and Datagram approach - IP Addressing methods (Ipv4 & Ipv6)) - Subnetting - Routing - Distance Vector Routing - RIP - Link State Routing - OSPF - BGP - Multicast Routing - MOSPF - DVMRP - Broadcast Routing. Congestion control algorithms: Open loop congestion control, Closed loop congestion control, QoS parameters, Token & Leaky bucket algorithms.	8	CO4
		Self-learning Topics: Software-Defined Networking (SDN) and Network Virtualization, Case study on the design of an enterprise network for a small company.	8	
V	**Transport Layer**	Transport Services - UDP - TCP - Congestion Control Effects of Congestion and Traffic Management – Congestion Avoidance Mechanisms- Queuing mechanism -Quality of Services (QOS) – QOS Parameters.	7	CO5
		Self-learning Topics: Real-Time Transport Protocols (RTP, RTCP) and Streaming Applications	7	
VI	**Application Layer**	Domain Name Space (DNS) - Electronic Mail - WWW - Cryptography Techniques. Case Studies – FTP, HTTP, SMTP, SNMP.	7	CO6
		Self-learning Topics: Web Security Mechanisms (SSL/TLS, HTTPS Security Model)	7	

Text Books:

1. Andrew S. Tanenbaum, David J. Wetherall, “Computer Networks”, Fifth Edition, Prentice Hall, 2011.
2. James F. Kurose, Keith W. Ross, “Computer Networking: A Top-Down Approach”, Eighth Edition, Pearson, 2020.
3. Behrouz A. Forouzan, “Data Communications and Networking”, Fifth Edition, McGraw-Hill Education, 2012.

References:

1. William Stallings, “Data and Computer Communications”, Pearson Education, Eleventh Edition, 2021.
2. Douglas E. Comer, “Internetworking with TCP/IP, Vol 1: Principles, Protocols, and Architecture”, Pearson, Sixth Edition, 2013

Online References:

Sr	Module	Free Online Courses
1	Module I	https://online.stanford.edu/courses/cs144-introduction-computer-networking?utm_source=copilot.com https://nptel.ac.in/courses/106105082
2	Module II	
3	Module III	
4	Module IV	https://www.coursera.org/learn/ip-addressing-routing-network-management
5	Module V	https://nptel.ac.in/courses/106105082 coursera.org/learn/ip-addressing-routing-network-management
6	Module VI	https://www.coursera.org/learn/ip-addressing-routing-network-management

Course Code	Course Name	Teaching Scheme (Contact Hours Per Week			Teaching Scheme (Contact Hours Per Semester)					Total Credits (C) (Notional Learning Hour/30
		L	T	P	L	T	P	SL	Notional Learning Hour	
12211503	Data Warehousing and Mining	3	-	-	45	-	-	45	90	3

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-I	IAT-II	Total					
12211503	Data Warehousing and Mining	20	20	40	60	2.5	--	--	100

Rationale:

The rapid growth of data across industries has transformed traditional decision-making processes into data-driven ecosystems. Conventional Data Warehousing and Data Mining approaches are no longer sufficient to handle large-scale, real-time, and heterogeneous data sources. Therefore, this course integrates foundational concepts of Data Warehousing and classical Data Mining techniques with modern advancements in Big Data Analytics, Machine Learning, and Cloud Data Warehousing.

Course Objectives:

1. To identify the significance and applications of Data Warehousing and Data Mining in various domains.
2. To understand the architecture, components, and operations involved in building a data warehouse.
3. To analyze large datasets and apply appropriate models and algorithms for mining useful patterns.
4. To explore different types of data mining techniques such as classification, clustering, and association rule mining.
5. To study the concepts and techniques involved in Big Data Analytics and AI-driven Data Mining.
6. To develop interest and understanding in current trends, tools, and research areas in data mining.

Course Outcomes:

1. Understand the fundamentals of data warehousing and apply dimensional modeling techniques along with OLAP operations.
2. Explain data mining principles and perform data preprocessing and visualization.
3. Identify and apply suitable classification techniques for real-world problems.
4. Apply clustering techniques for data analysis.
5. Analyze and generate association rules from transactional datasets.
6. Design and implement scalable, AI-enabled data mining solutions using Big Data frameworks and cloud technologies while ensuring ethical practices

Prerequisite: Database Management Systems, Basic Statistics and Probability, Programming Fundamentals.

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
I	**Data Warehousing Fundamentals**	Introduction to Data Warehouse, Data warehouse architecture, Data warehouse versus Data Marts, E-R Modeling versus Dimensional Modeling, Information Package Diagram, Data Warehouse Schemas; Star Schema, Snowflake Schema, Fact less 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.	07	CO1
		Self-learning Topics:. Slowly Changing Dimensions (SCD), Data Lake vs Data Warehouse, Real-world warehouse case studies.	07	
II	**Introduction to Data Mining, Data Exploration and Data Pre-processing**	Data Mining Task Primitives, Architecture, KDD process, Issues in Data Mining, Applications of 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.	07	CO2

		Self-learning Topics:. Feature Engineering, Handling Imbalanced Data, Advanced Data Visualization Tools.	07	
III	**Classification**	Basic Concepts, Decision Tree Induction, Naïve Bayesian Classification, Accuracy and Error measures, Evaluating the Accuracy of a Classifier: Holdout & Random Subsampling, Cross Validation, Bootstrap.	08	CO3
		Self-learning Topics: Random Forest, Support Vector Machines, Model Deployment Basics	08	
IV	**Clustering**	Types of data in Cluster analysis, Partitioning Methods (k-Means, k- Medoids), Hierarchical Methods (Agglomerative, Divisive).	08	CO4
		Self-learning Topics: DBSCAN, OPTICS, Cluster Validity Indices	08	
V	**Mining frequent patterns and associations**	Market Basket Analysis, Frequent Item sets, Closed Item sets, and Association Rule, Frequent Pattern Mining, Apriori Algorithm, Association Rule Generation, Improving the Efficiency of Apriori, Mining Frequent Itemsets without candidate 6 generation, Introduction to Mining Multilevel Association Rules and Mining Multidimensional Association Rules.	08	CO5
		Self-learning Topics: FP-Growth Algorithm, Sequential Pattern Mining, Lift & Conviction Metrics.	08	
VI	**Big Data Analytics & AI-Driven Data Mining**	Big Data Characteristics (5Vs), Hadoop Ecosystem (HDFS, MapReduce), Apache Spark, Real-Time & Streaming Analytics, Machine Learning Integration, Cloud Data Warehousing, Ethical and Responsible AI practices.	07	CO6
		Self-learning Topics: Deep Learning Basics, MLOps Concepts, Data Governance & GDPR	07	

Text Books:

1. Paulraj Ponniah, " Data Warehousing: Fundamentals for IT Professionals", Wiley India.
2. Han, Kamber, "Data Mining Concepts and Techniques", Morgan Kaufmann 2nd 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	Free Online Courses
1	https://onlinecourses.nptel.ac.in/noc21_cs06/preview
2	http://digimat.in/nptel/courses/video/106105174/L21.html
3	https://onlinecourses.nptel.ac.in/noc26_cs14/preview

Course Code	Course Name	Teaching Scheme (Contact Hours Per Week			Teaching Scheme (Contact Hours Per Semester)					Total Credits (C) (Notional Learning Hour/30
		L	T	P	L	T	P	SL	Notional Learning Hour	
12212504	Computer Networks lab	--	--	2	--	--	30	--	30	1

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Practical/ Oral	Total
		Internal assessment			End Sem. Exam			
		IAT-I	IAT-II	Total				
12212504	Computer Networks Lab	--	--	--	--	25	25	50

Lab Objectives:

1. To practically explore OSI layers and understand the usage of simulation tools.
2. To analyze, specify, and design the topological and routing strategies for an IP-based networking infrastructure.
3. To identify the various issues of a packet, transfer from source to destination, and how they are resolved by the various existing protocols.
4. To evaluate the performance of different routing algorithms and their impact on network efficiency.
5. To understand and implement various LAN, WAN, and wireless network configurations using simulation tools.
6. To examine protocol functionalities at different layers and assess their role in end-to-end communication.

Lab Outcomes:

1. Remember networking commands and Linux-based setup procedures for configuring a network environment.
2. Understand the functionalities of different networking tools and simulators such as NS2 and Wireshark.
3. Apply core programming APIs to implement basic networking concepts and socket programming.
4. Analyze the behavior of networking protocols through simulation and packet inspection.
5. Evaluate the performance of various routing and congestion control algorithms using simulators.
6. Create and configure a complete simulated network scenario to demonstrate communication protocols and services.

List of Experiments.

Sr No	List of Experiments	Hrs
01	Study of RJ45 and CAT6 Cabling and connection using crimping tool.	2
02	Use basic networking commands in Linux (ping, tracert, nslookup, netstat, ARP, RARP, ip, ifconfig, dig, route)	2
03	Build a simple network topology and configure it for static routing protocol using packet tracer. Setup a network and configure IP addressing, subnetting, masking.	2
04	Perform network discovery using discovery tools (eg. Nmap, mrtg)	2
05	Use Wire shark to understand the operation of TCP/IP layers: ● Ethernet Layer: Frame header, Frame size etc. ● Data Link Layer: MAC address, ARP (IP and MAC address binding) ● Network Layer: IP Packet (header, fragmentation), ICMP (Query and Echo) ● Transport Layer: TCP Ports, TCP handshake segments etc. ● Application Layer: DHCP, FTP, HTTP header formats	2
06	Use simulator (Eg. NS2) to understand the functioning of ALOHA, CSMA/CD.	2
07	Study and Installation of Network Simulator (NS3)	2

08	a. Set up multiple IP addresses on a single LAN. b. Using nestat and route commands of Linux, do the following: ● View current routing table ● Add and delete routes ● Change default gateway c. Perform packet filtering by enabling IP forwarding using IPtables in Linux.	2
09	Design VPN and Configure RIP/OSPF using Packet tracer.	2
10	Socket programming using TCP or UDP	2
11	Perform File Transfer and Access using FTP	2
12	Perform Remote login using Telnet server	2
13	Implement DHCP Server Configuration – Configure a Dynamic Host Configuration Protocol (DHCP) server to automatically assign IP addresses to clients and test dynamic allocation.	2
14	Simulate Congestion Control in TCP – Use a simulator (e.g., NS2/NS3) to study TCP congestion control mechanisms such as slow start, congestion avoidance, and fast retransmit.	2
15	Configure Firewall Rules and IDS – Set up basic firewall rules using Linux iptables and demonstrate intrusion detection with Snort or similar IDS tools.	2

Sr No	List of Assignments / Tutorials	Hrs.
01	Assignment covers the topics from first three units (Introduction, Circuit and Packet Switching, Data Link Layers.) limited to three Questions	2
02	Assignment covers the topics from Last three units (Network Layer, Transport Layer and Application Layer of Computer Networks) limited to three Questions	2

Text Books:

1. Andrew S. Tanenbaum & David J. Wetherall – Computer Networks, 5th Edition, Pearson (2011)
2. Behrouz A. Forouzan – Data Communications and Networking, 5th Edition, McGraw Hill (2012)

3. William Stallings – Data and Computer Communications, 10th Edition, Pearson (2013)
4. Douglas E. Comer – Internetworking with TCP/IP, Vol. 1, 6th Edition, Pearson (2014)
5. Larry L. Peterson & Bruce S. Davie – Computer Networks: A Systems Approach, 5th Edition, Morgan Kaufmann (2011)
6. James F. Kurose & Keith W. Ross – Computer Networking: A Top Down Approach, 7th Edition, Pearson (2017)

References:

1. W. Richard Stevens – UNIX Network Programming, Vol. 1, 3rd Edition, Pearson (2004)
2. W. Richard Stevens – TCP/IP Illustrated, Vol. 1, 2nd Edition, Addison-Wesley (2011)
3. Evi Nemeth et al. – UNIX System Administration Handbook, 4th Edition, Pearson (2010)
4. Chris Sanders – Practical Packet Analysis: Using Wireshark to Solve Real-World Network Problems, 3rd Edition, No Starch Press (2017)
5. NS2/NS3 Documentation & Tutorials – Latest online releases (community maintained)

Online Resources:

Sr. No.	Website Name
1.	https://www.netacad.com/courses/packet-tracer/introduction-packet-tracer 2
2.	https://www.coursera.org/projects/data-forwarding-computer-networks 3
3	https://www.edx.org/course/ilabx-the-internet-masterclass

Assessment:

Term Work: Term Work shall consist of all practicals based on the above list. Also, Term work Journal must include at least 2 assignments per subject.

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

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

Course Code	Course Name	Teaching Scheme (Contact Hours Per Week			Teaching Scheme (Contact Hours Per Semester)					Total Credits (C) Notional Learning Hour/30
		L	T	P	L	T	P	SL	Notional Learning Hour	
12212505	Data Warehousing & Mining Lab	--	--	2	--	--	30	--	30	1

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Practical/ Oral	Total
		Internal assessment			End Sem. Exam			
		IAT-I	IAT-II	Total				
12212505	Data Warehousing & Mining Lab	--	--	--	--	25	25	50

Lab Objectives:

1. To understand the concepts of data warehousing, dimensional modeling, and data marts for decision support systems.
2. To design and implement data warehouse schemas (Star and Snowflake) using relational databases.
3. To perform OLAP operations such as roll-up, drill-down, slice, dice, and pivot on multidimensional datasets.
4. To apply data preprocessing and exploratory data analysis techniques for preparing datasets for data mining.
5. To implement data mining algorithms such as classification, clustering, and association rule mining.
6. To analyze real-world datasets using data mining tools and visualization techniques for extracting useful knowledge.

Lab Outcomes:

1. Design dimensional models and data warehouse schemas for analytical applications.
2. Implement fact and dimension tables and perform OLAP operations on multidimensional data.
3. Apply data preprocessing and data exploration techniques to improve data quality
4. Develop classification models to predict outcomes using appropriate algorithms.
5. Apply clustering and association rule mining techniques to discover hidden patterns in datasets.
6. Analyze and visualize data using modern data mining tools to support data-driven decision making.

List of Experiments.

Sr No	List of Experiments	Hrs
01	Industry Case Study – Data Warehouse Design	2
02	Implementation of all dimension table and fact table based on experiment 1 case study Data Warehouse Implementation.	2
03	Implementation of OLAP operations: Slice, Dice, Rollup, Drilldown and Pivot based on experiment 1 case study	2
04	Implementation of Data Preprocessing Pipeline	2
05	Implementation of Data Visualization and Data Exploration Techniques	2
06	Implementation of Classification Techniques for Prediction	2
07	Implementation of Customer Segmentation using Clustering	2
08	Implementation of Hierarchical Clustering for Pattern Analysis	4
09	Implementation of Market Basket Analysis using Association Rules	4
10	Implementation of Web Mining and PageRank Algorithm	4
11	Implementation of End-to-End Mini Industry Project using Data Mining Techniques	4

Sr No	List of Assignments / Tutorials	Hrs.
01	Assignment covers the topics from first three units (Data Warehousing Fundamentals, Introduction to Data Mining, Data Exploration and Data Pre-processing, Classification) limited to three Questions	2
02	Assignment covers the topics from Last three units (Clustering, Mining frequent patterns and associations, Big Data Analytics & AI-Driven Data Mining) limited to three Questions	2

Text Books:

1. Paulraj Ponniah, " Data Warehousing: Fundamentals for IT Professionals", Wiley India..
2. Han, Kamber, "Data Mining Concepts and Techniques", Morgan Kaufmann 2nd 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 Resources:

Sr. No.	Website Name
1.	https://onlinecourses.nptel.ac.in/noc20_cs12/preview
2.	https://www.coursera.org/specializations/data-mining

Assessment:

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

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

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

Course Code	Course Name	Teaching Scheme (Contact Hours Per Week			Teaching Scheme (Contact Hours Per Semester)					Total Credits (C) (Notional Learning Hour/30)
		L	T	P	L	T	P	SL	Notional Learning Hour	
12212506	IDEA LAB 5 - (Innovation Design Engineering and Apply)	--	--	2	--	--	30	--	30	1

Course Code	Course Name	Examination Scheme					Term work	Pract / Oral	Total
		Internal Assessment			End Semester Exam	Exam Duration (in Hrs)			
		IAT-I	IAT-II	Total					
12212506	IDEA LAB – 5 (Innovation Design Engineering and Apply)	--	--	--	--	--	25	25	50

Rationale:

Aligned with the National Education Policy (NEP) 2020, the institution emphasizes experiential, interdisciplinary, and project-based learning through the IDEA Lab—a central hub for hands-on innovation.

To strengthen the undergraduate research ecosystem, the institution has adopted a theme-based academic model aligned with UN SGD. Each semester features six curated problem statements based on local need and aligned with core subjects in the same semester, enabling students to apply classroom knowledge to real-world challenges. Every student selects one problem and develops an individual, subject-integrated solution—enhancing both academic understanding and research skills.

The IDEA Lab supports this initiative with facilities for design thinking, prototyping, and product development. Students maintain a project logbook throughout the semester to track their progress and reflections.

To ensure academic accountability, a structured assessment framework is implemented.

1. Project assessment is carried out using standardized IDEA Lab rubrics with multiple evaluation parameters.
2. One of the key parameters is **Engineering Subject-Based Assessment**, which focuses on the effective application of same-semester subject knowledge in project work, ensuring integration of theoretical learning with practical implementation.

Lab Objectives:

1) To promote experiential and project-based learning that bridges theoretical knowledge with real-world problem-solving.
2) To encourage interdisciplinary integration by enabling students to apply concepts from multiple subjects within a single cohesive project.
3) To develop innovation and design thinking skills through hands-on activities and iterative solution development.
4) To foster critical thinking and creativity by engaging students in open-ended problems with multiple solution pathways.
5) To enhance communication, collaboration, and documentation skills essential for professional engineering practice.
6) To build an entrepreneurial and research mindset by guiding students to develop scalable, socially-relevant, and technically viable prototype

Lab Outcomes: Student will be able to

1) Recall and articulate key concepts from core and allied subjects relevant to the assigned project.
2) Explain the interdisciplinary nature of the problem and the role of each subject in addressing it.
3) Apply appropriate tools, techniques, and theoretical knowledge to develop project components.
4) Analyze problem constraints and user requirements to structure a feasible and efficient solution.
5) Evaluate multiple design options and justify the chosen solution based on technical and practical considerations.
6) Create a functional prototype or solution that demonstrates innovation, utility, and integration of interdisciplinary knowledge.

1) Guidelines for IDEA Project

a) Project Guidelines (Project Execution in IDEA Lab)

- Each student works on an individual interdisciplinary project aligned with the semester theme.
- Faculty in-charges mentor both the academic and technical aspects, and track weekly progress.
- Project assessment will be rubric-based, ensuring depth, innovation, documentation, and ownership.
- Students shall convert the best solution into working model using various components of their domain areas and demonstrate.

b) Guidelines for same semester Subject Concepts Applied within the Project

- Students must ensure that their IDEA Lab project meaningfully incorporates concepts from all core subjects of the current semester and maintain a project logbook as evidence of subject-wise application.
- Faculty in-charge must guide students to apply concepts from subjects studied in the same semester to their IDEA Lab projects.
- The term work evaluation will partially reflect the effective application of subject-specific knowledge.
- Faculty should regularly review the integration of academic content within the project work.

2) Implementation Strategy – IDEA Lab Learning Model

- A theme-based learning structure shall be implemented where each semester is anchored to a broad contextual domain such as Healthcare, Smart Cities, Energy, Agriculture, or Governance, providing the overall learning direction.
- From the identified theme, a system-level, open-ended problem statement shall be defined, serving as a solution-neutral framework that enables interdisciplinary thinking and innovation.
- The learning sequence shall follow a structured flow: Theme → Problem Statement → Same Semester Subject Application → Design–Implementation Experience, ensuring progressive transformation from context to solution.
- Themes shall provide the context (defining the domain of work), while problem statements shall provide the purpose, enabling students to identify meaningful real-world challenges.
- Subjects shall be treated as engineering tools, where each subject contributes specific knowledge components to solve different aspects of the problem.
- Students shall be encouraged to take ownership of their learning and projects, fostering curiosity, exploration, social relevance, and a mindset of creative risk-taking.
- Structured brainstorming and ideation sessions shall be conducted at the beginning of

the semester to guide students in problem identification and development of solution pathways through collaborative and interdisciplinary thinking.

- Guest lectures and expert talks shall be organized by inviting industry experts, researchers, and innovators aligned with the semester theme and subject areas to provide exposure to real-world challenges, trends, and opportunities.
- A mechanism shall be implemented to ensure uniqueness and originality of projects by systematically reviewing ideas to avoid duplication and encouraging novel, technology-driven solutions.
- Platforms shall be created for collaborative learning and discussion, enabling students to present ideas, receive peer and mentor feedback, and refine solutions through iterative engagement.
- Content beyond syllabus for as subject shall be integrated with IDEA Lab themes and problem statements to ensure deeper interdisciplinary application and enhanced conceptual understanding.
- Preference shall be given to ensuring same-semester academic alignment, where subject faculty contribute to effective contextual integration of project-based learning.
- Continuous monitoring and documentation shall be ensured through logbooks, milestone tracking, and periodic progress reviews to maintain academic rigor and outcome alignment.
- Coordination with subject faculty shall be ensured for effective mapping of theoretical concepts into project execution and for evidence-based term work assessment.
- The IDEA Lab shall function as an institutional mechanism for implementing the CDIO (Conceive–Design–Implement–Operate) framework, promoting experiential and outcome-based learning through structured mini-projects.

Implementation Notes:

- Guide will assess subject's contribution using specific evidence such as:
  - Logbooks
  - Subject-specific outputs (e.g., simulations, designs)
  - Paper publications or review presentations

3) Guidelines for Assessment

Assessment Month	**Marks**
Month 1 (Formative 1)	25 marks
Month 2 (Formative 2)	25 marks
Month 3 (Formative 3)	25 marks

Total Assessment Structure

Component	**Marks**	**Assessed By**
Term work – Project Execution	25 Marks	Project Guide – Average of all innovation
Viva Voce (Final Evaluation)	25 Marks	External Examiner

Criteria	Excellent (10)	Good (8)	Satisfactory (6)	Needs Improvement (4)
1. Innovation	Highly original and disruptive idea solving a real-world or futuristic problem	Novel idea with a creative approach to a known problem	Moderately innovative; adapts existing solutions	Common or repetitive idea with minimal novelty
2. Design	User-centric, systematic design with professional-level tools, prototyping, or simulations	Logical structure; clearly presented using standard tools	Functional design but limited user perspective or refinement	Poor or confusing design with lack of user consideration
3. Engineering	Applies concepts from all same-semester subjects with clear integration and interdisciplinary depth	Applies knowledge from 4 or more subjects with relevance	Applies 2–3 subjects with moderate clarity	Applies 1 subject with weak or shallow connection
4. Implementation	Complete, fully functional prototype with rigorous testing and real-world validation	Working model with minor errors; partially tested	Prototype is incomplete; basic testing or debug attempt	Poorly implemented; fails to function properly
5. Application	Strong real-world relevance with alignment to SDGs/National Missions; impact clearly identified	Clear practical use-case with user/market relevance	Some practical value but limited clarity on impact	Vague or irrelevant application; weak user connect

Course Code	Course Name (PEC-III)	Teaching Scheme (Contact Hours Per Week			Teaching Scheme (Contact Hours Per Semester)					Total Credits (C) (Notional Learning Hour/30
		L	T	P	L	T	P	SL	Notional Learning Hour	
12215701	Internet of Things	3	-	-	45	-	-	45	90	3

Course Code	Course Name (PEC-III)	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration(in Hrs)			
		IAT-I	IAT-II	Total					
12215701	Internet of Things	20	20	40	60	2.5	--	--	100

Rationale:

The Internet of Things (IoT) has emerged as a transformative technology that connects physical devices, sensors, and systems to the internet, enabling intelligent data collection, communication, and automation. IoT plays a significant role in various domains such as smart cities, healthcare, agriculture, transportation, industrial automation, and environmental monitoring.

This course is designed to provide students with a comprehensive understanding of IoT architecture, communication protocols, sensor technologies, and cloud integration. It also focuses on the design and development of IoT-based applications using embedded platforms and modern networking technologies. The course further introduces students to data management, cloud computing, and security aspects of IoT systems, enabling them to address real-world challenges and develop secure and scalable IoT solutions. Through theoretical concepts and practical exposure, students will gain the necessary skills to analyze, design, and implement IoT-based systems for real-world applications.

Course Objectives:

1) To impart knowledge of IoT infrastructure, including sensor technologies, embedded platforms, and networking technologies used in Internet of Things systems.
2) To understand different IoT architectures and design models, including layered architecture and IoT reference models.
3) To explore communication protocols used in IoT, such as MQTT, CoAP, IPv6, ZigBee, Bluetooth, and other networking technologies.
4) To analyze real-world problems and map them into IoT-based solutions using appropriate sensors, devices, and cloud platforms.
5) To design and develop IoT applications using microcontrollers, sensor interfacing, and cloud-based services.
6) To understand practical and societal applications of IoT, including smart cities, healthcare, agriculture, and industrial automation.

Course Outcomes:

By the end of the course, students will be able to:

1) Explain the fundamental concepts, architecture, standards, and key components of the Internet of Things.
2) Analyze IoT communication technologies and protocol stack including IPv6, 6LoWPAN, MQTT, CoAP, ZigBee, and Bluetooth.
3) Design and develop IoT-based applications using sensors, microcontrollers (Arduino/Raspberry Pi), and cloud platforms.
4) Apply data management, cloud computing, and analytics techniques for processing IoT data.
5) Identify IoT security challenges, vulnerabilities, and attacks, and apply appropriate security mechanisms to protect IoT systems.
6) Design IoT solutions for real-world applications such as smart cities, healthcare, agriculture, and industrial automation.

Prerequisite: Computer Networks and Wireless Networks, Microcontroller and Microprocessor, Mobile Computing

DETAILED SYLLABUS:

Sr. No.	Module	Detailed Content	Hours	CO Mapping
I	**IoT Fundamentals**	Introduction to Internet of Things, Concepts and Characteristics of IoT, Challenges and Issues in IoT, Convergence of IT and OT, Industry 4.0, IoT Functional Blocks, IoT Architectural Principles, Things-oriented vision. IoT Applications and real-world scenarios such as Smart Cities, Smart Healthcare and Smart Homes.	7	CO1
		Self-learning Topics: Emerging IoT platforms, Smart city IoT deployments in India, Industrial IoT (IIoT) standards.	7	
II	**Communication Technologies used in IoT**	RFID, NFC, ZigBee, BLE. Wireless Sensor Networks and Sensor Technologies. Data Link Layer protocols: Wi-Fi (IEEE 802.11), Bluetooth (IEEE 802.15.1), LRWPAN (IEEE 802.15.4), and 6LoWPAN communication standards.	8	CO2
		Self-learning Topics: Comparison of IoT wireless communication technologies, LPWAN technologies (LoRaWAN, NB-IoT)	8	
III	**IoT Architecture and Protocols**	IoT Reference Architecture and IoT World Forum (IoTWF) Architecture. IoT Functional Stack and Data Management Stack. Internet Layer protocols: IPv4 and IPv6 addressing, BLE over IPv6, ZigBee over IPv6. Transport protocols: TCP and UDP for IoT. SCADA, Application Layer protocols: MQTT, CoAP, AMQP, WebSocket.	8	CO3
		Self-learning Topics: Implementation of MQTT broker, Comparison of REST and CoAP protocols	8	
IV	**IoT Hardware and Development Platforms**	Working principles of sensors and actuators. Hardware platforms: Arduino, Raspberry Pi and other embedded devices. Programming IoT devices using Python. Sensor interfacing and data acquisition. Communication between IoT devices and mobile devices using Bluetooth, WiFi and USB.	7	CO4

		Self-learning Topics: Development of simple sensor-based IoT applications, Mobile app integration with IoT devices	7	
V	**Cloud Computing and Data Analytics in IoT**	Basics of cloud infrastructure: cloud storage and computational infrastructure. Integration of IoT and Cloud. Data transformation, aggregation and visualization in IoT stack. Distributed frameworks such as Hadoop ecosystem. IoT data analytics and visualization techniques. Edge computing and Fog computing.	7	CO5
		Self-learning Topics: IoT cloud platforms (AWS IoT, Azure IoT Hub), Edge computing frameworks for IoT	7	
VI	**IoT Applications**	Home Automation – Smart Lighting, Smart Appliances, Intrusion Detection, Smoke/Gas Detectors Cities – Smart Parking, Smart Lighting, Smart Roads, Structural Health Monitoring Energy – Smart Grids, Renewable Energy Systems, Prognostics Retail – Inventory Management, Smart Payments, Smart Vending Machines Agriculture – Smart Irrigation, Green House Control Industry – Machine Diagnostics & Prognosis, Indoor Air Quality Monitoring Health & Lifestyle – Health & Fitness Monitoring, Wearable Electronics Case Studies: Smart City initiatives in India Industrial automation and predictive maintenance Healthcare applications, such as remote patient monitoring	8	CO6
		Self-learning Topics: Surveillance Environment – Weather Monitoring, Air Pollution Monitoring, Noise Pollution Monitoring, Forest Fire Detection, River Floods Detection Logistics – Route Generation & Scheduling, Fleet Tracking, Shipment Monitoring	8	

Text Books:

1) David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Robert Barton, Jerome Henry, *IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things*, Cisco Press, 1st Edition, 2017.
2) Alessandro Bassi, Martin Bauer, Martin Fiedler, Thorsten Kramp, Rob van Kranenburg, Sebastian Lange, Stefan Meissner, *Enabling Things to Talk: Designing IoT Solutions with the IoT Architectural Reference Model*, Springer-Verlag Berlin, 2016.
3) Arshdeep Bahga and Vijay Madisetti, *Internet of Things: A Hands-on Approach*, Universities Press, 2015.

References:

1) Olivier Hersent, David Boswarthick, Omar Elloumi, *The Internet of Things: Key Applications and Protocols*, Wiley, 2012.
2) Rajkumar Buyya and Amir Vahid Dastjerdi, *Internet of Things: Principles and Paradigms*, Elsevier, 2016.
3) Raj Kamal, *Internet of Things: Architecture and Design Principles*, McGraw Hill Education, 2017.
4) Kai Hwang and Min Chen, *Big Data Analytics for Cloud, IoT and Cognitive Computing*, Wiley, 2018.

Online References:

Sr. No.	Website Name
1.	https://nptel.ac.in/courses/106/105/106105166/
2.	https://nptel.ac.in/courses/108/108/108108098/
3.	https://nptel.ac.in/courses/106/105/106105195/
4.	https://www.coursera.org/specializations/IoT

Course Code	Course Name PEC-III	Teaching Scheme (Contact Hours Per Week			Teaching Scheme (Contact Hours Per Semester)					Total Credits (C) (Notional Learning Hour/30
		L	T	P	L	T	P	SL	Notional Learning Hour	
12215702	Digital Communication	3	-	-	45	-	-	45	90	3

Course Code	Course Name PEC-III	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration(in Hrs)			
		IAT-I	IAT-II	Total					
12215702	Digital Communication	20	20	40	60	2.5	--	--	100

Rationale :

The Digital Communication syllabus is designed to build a strong foundation in information theory, coding, and modulation techniques essential for modern communication systems. It introduces students to source coding, error detection and correction methods, and baseband transmission principles, ensuring they can analyze and design reliable systems. By covering optimum detection and digital modulation schemes, learners gain the ability to evaluate performance under noise and bandwidth constraints. The inclusion of practical coding techniques and spectral analysis bridges theory with real-world applications. Overall, the syllabus equips students with analytical and problem-solving skills required for advanced studies and industry practices in communication engineering.

Course Objectives:

1. To introduce the fundamental concepts of information theory, source coding, and channel capacity in digital communication systems.
2. To develop understanding of error control mechanisms, including ARQ, FEC, and error detection techniques.
3. To explain the principles and implementation of error correction codes such as linear block codes, cyclic codes, and convolutional codes.
4. To provide knowledge of baseband transmission systems, line coding techniques, and the effects of channel impairments such as ISI and ICI.
5. To understand optimum detection techniques for baseband signals using matched filters and receiver structures.
6. To analyze digital modulation schemes in terms of generation, detection, bandwidth efficiency, and error performance.

Course Outcomes:

1. **Explain** the concepts of information theory, including entropy, information rate, and channel capacity.
2. **Apply** error detection and control techniques such as CRC, ARQ, and FEC for reliable data communication.
3. **Analyze** error correction codes, including linear block, cyclic, and convolutional codes, for error detection and correction capability.
4. **Evaluate** baseband transmission systems and compare various line coding techniques based on performance parameters.
5. **Interpret** optimum detection methods such as matched filter, correlator, and integrate-and-dump receiver.
6. **Analyze** digital modulation schemes using signal space representation and compare their performance in terms of bandwidth and error probability.

Prerequisite:

1. Engineering Mathematics-IV
2. Signals and Systems
3. Principles of Communication Engineering

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
I	Information Theory and Source Codes	Block diagram of digital communication system, Information content of a source symbol, Source entropy, Average information rate, AWGN channel, and Shannon-Hartley channel capacity theorem. Introduction of source code, Huffman code, Shannon-Fano code.	7	CO1
		Self-learning Topics: 1. **Digital Communication Block Diagram:** Source, encoder, channel, decoder, and destination. 2. **Information Content:** Measured in bits; source entropy shows average uncertainty. 3. **Information Rate:** Describes how fast information is produced by the source. 4. **AWGN Channel**: Standard model for noisy environments in communication systems. 5. **Shannon–Hartley Theorem**: Defines maximum channel capacity under bandwidth and signal-to-noise constraints.	7	
II	Error Control System and Error Detection Codes	Introduction of error control system, Automatic Retransmission Query (ARQ) system, Types of ARQ systems and comparison, Forward error correction (FEC) system. Comparison between FEC and ARQ. Error detection codes: Vertical Redundancy Check (VRC) code, Longitudinal Redundancy Check (VRC) code, Cyclic Redundancy Check (CRC) code and Checksum code.	7	CO2

		Self-learning Topics: 1. **Error Occurrence**: How transmission errors happen during data communication. 2. **Error Control Systems**: Techniques that improve reliability of communication. 3. **ARQ Methods**: Stop-and-Wait, Go-Back-N, and Selective Repeat for error detection and correction. 4. **Forward Error Correction (FEC)**: Methods to correct errors without retransmission. 5. **Practical Exercises**: Calculating parity bits (VRC, LRC), CRC generation/checking, identifying errors, and comparing ARQ vs FEC efficiency.	7	
III	Error Correction Codes	Linear block code: Code generation, calculation of minimum Hamming distance, error detection capability, error correction capability, implementation of encoder, error detection, syndrome table, error correction and implementation of decoder. Cyclic code: Code generation, calculation of minimum Hamming distance, error detection capability, error correction capability, implementation of encoder, error detection, syndrome table, error correction and implementation of decoder. Convolutional code: Generation, path responses, encoder, state transition table, state diagram, tree diagram, trellis diagram, decoding using Viterbi's algorithm.	8	CO3

		Self-learning Topics: 1. **Error Control Codes:** Fundamentals of linear block codes, cyclic codes, and convolutional codes for error detection/correction. 2. **Key Concepts**: Code generation, minimum Hamming distance, error detection/correction capability, encoder/decoder design, syndrome tables. 3. **Decoding Methods:** Understanding Andrew Viterbi's algorithm for convolutional codes. 4. **Practical Exercises:** Generating code words, calculating Hamming distance, constructing syndrome tables, and implementing encoders/decoders. 5. **Visualization**: Drawing state diagrams, tree diagrams, and trellis diagrams to decode convolutional codes.	8	
IV	Baseband Transmission	Block diagram of baseband transmitter-receiver system, Line codes (RZ and NRZ UniPolar formats, RZ and NRZ Polar formats, NRZ Bipolar format (AMI format), NRZ Manchester format, and Quaternary Polar format). Comparison of line codes with respect to bandwidth, power requirement, synchronization capability, DC level, polarity inversion error and complexity. Power spectral density and spectrum of NRZ Unipolar and Polar formats. Inter Symbol Interference (ISI), Inter Channel Interference (ICI). Nyquist criterion for distortionless baseband binary transmission, Nyquist bandwidth and practical bandwidth.	8	CO4
		Self-learning Topics: 1. **Baseband System:** Working of transmitter–receiver block diagram. 2. **Line Coding Techniques:** RZ, NRZ (Unipolar/Polar), AMI, Manchester, and Quaternary Polar formats. 3. **Comparison Factors**: Bandwidth, power requirement, synchronization, DC component,	8	

		polarity inversion error, and complexity. 4. **Practical Exercises**: Drawing block diagrams, sketching waveforms, comparing characteristics, analyzing PSD of NRZ formats. 5. **ISI/ICI & Nyquist**: Understanding Inter Symbol/Channel Interference, Nyquist criterion, Nyquist bandwidth, and distortionless transmission.		
V	Optimum Detection of Baseband Signal	Matched filter, Output SNR, Transfer function, Impulse response and Error probability. Integrate and dump receiver, Correlator receiver.	7	CO5
		Self-learning Topics: 1. **Matched Filter Concepts:** Understanding transfer function, impulse response, output SNR, and error probability. 2. **Role of Matched Filters:** How they maximize signal detection in noisy environments and improve receiver performance. **3. Practical Exercises:** Deriving transfer functions/impulse responses, calculating SNR and error probability. 4. **Receiver Comparison:** Studying matched filter, integrate-and-dump receiver, and correlator receiver.	7	
VI	Digital Modulations	Generation, Detection, Error probability (using signal space representation and Euclidean distance), Bandwidth (using PSD and spectrum except for MSK) and applications of the following modulations: Binary ASK, Binary PSK, Quadrature PSK, Off-Set QPSK, M-ary PSK, Binary FSK, M-ary FSK, 16-ary QASK and MSK.	8	CO6
		Self-learning Topics: 1. **Digital Modulation Techniques**: Binary ASK, Binary PSK, QPSK, Offset QPSK, M-ary PSK, Binary/M-ary FSK, 16-ary QASK, and MSK. 2. **Key Concepts**: Generation, detection, bandwidth requirements, and real-world applications of these modulation schemes.	8	

		3. **Signal Space Representation**: Understanding Euclidean distance and its impact on error probability and performance. 4. **Practical Exercises**: Drawing waveforms, constellation diagrams, block diagrams, calculating bandwidth/error probability, and comparing efficiency. 5. **Application Focus**: Identifying suitable communication scenarios for each modulation method.		

Text Books:

1.H. Taub, D. Schlling, and G. Saha-Principles of Communication Systems, Tata Mc- Graw Hill, New Delhi, Third Edition, 2012.

2. Lathi B P, and Ding Z-Modern Digital and Analog Communication Systems, Oxford University Press, Fourth Edition, 2017.

3. Haykin Simon-Digital Communications, John Wiley and Sons, New Delhi, Fourth Edition, 2014.

4. John G. Proakis-Digital Communications, McGraw-Hill, Fourth Edition

References:

1. Sklar B, and Ray P. K.-Digital Communication: Fundamentals and applications, Pearson,Dorling Kindersley (India), Delhi, Second Edition, 2009.

2. T L Singal-Analog and Digital Communication, Tata Mc-Graw Hill, New Delhi, First Edition,2012.

3. P Ramakrishna Rao-Digital Communication, Tata Mc-Graw Hill, New Delhi, First Edition,2011.

4. K. Sam Shanmugam-Digital and analog communication Systems, John Wiley and sons.

5. Upamanyu Madhow- Fundamentals of Digital Communication- Cambridge University Press

6. W.C. Huffman, Vera Pless- Fundamentals of Error Correcting Codes, Cambridge University Press

7. Graham Wade-Coding Techniques, Palgrave, New York

Online References:

Sr	**Module**	**Free Online Courses**
1	Module I	https://nptel.ac.in/courses/108/101/108101113/ https://nptel.ac.in/courses/108/102/108102096/ https://nptel.ac.in/courses/108/102/108102120/
2	Module II	
3	Module III	
4	Module IV	https://nptel.ac.in/courses/108/102/108102096/ https://nptel.ac.in/courses/108/102/108102120
5	Module V	https://nptel.ac.in/courses/108/102/108102096/ https://nptel.ac.in/courses/108/102/108102120
6	Module VI	https://nptel.ac.in/courses/108/102/108102120

Course Code	Course Name PEC-III	Teaching Scheme (Contact Hours Per Week			Teaching Scheme (Contact Hours Per Semester)					
		L	T	P	L	T	P	SL	Notional Learning Hour	Total Credits (C) (Notional Learning Hour/30
12215703	Artificial Intelligence	3	--	--	45	--	--	45	90	3

Course Code	Course Name PEC-III	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration(in Hrs)			
		IAT-I	IAT-II	Total					
12215703	Artificial Intelligence	20	20	40	60	2.5	--	--	100

Rationale :

Artificial Intelligence is widely used in industries such as healthcare, banking, retail, and automation. This course introduces students to core AI concepts along with modern industry trends like Machine Learning, Natural Language Processing, and Generative AI. It helps students understand how intelligent systems are designed and applied in real-world scenarios.

Course Objectives:

1. To explain basic concepts and techniques of Artificial Intelligence.
2. To apply problem-solving methods using search algorithms.
3. To describe knowledge representation and reasoning techniques.
4. To apply machine learning concepts and data-driven approaches.
5. To demonstrate AI applications such as NLP and Generative AI.
6. To design simple AI-based solutions for real-world problems.

Course Outcomes:

1. Explain AI concepts, agents, and real-world AI pipelines.
2. Apply search algorithms and analyze their performance.
3. Illustrate knowledge representation and reasoning techniques.
4. Apply machine learning techniques to solve problems.
5. Demonstrate NLP tools, LLMs, and prompt engineering.
6. Analyze and evaluate AI applications, ethics, and system design.

Prerequisite: Discrete Mathematics, Data Structures, Basic Python Programming

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
I	**Introduction to AI & Intelligent Agents**	-Overview of AI, Machine Learning, Deep Learning, and Generative AI concepts and their relationships. -Applications of AI in Industry, -Overview of AI system development process including data collection, model building, evaluation, and deployment. -Intelligent agents: Types and structure Performance measures	7	**CO1**
		Self-learning Topics:. AI tools: Google Colab, Jupyter Notebook, Overview of popular AI frameworks	7	
II	**Problem Solving & Search Techniques**	-Problem formulation and state space representation -Study of fundamental search algorithms such as BFS, DFS, and A* for solving AI problems. -Introduction to heuristic functions and their role in improving search efficiency. -Analysis of time and space requirements of different search techniques. -Application of search algorithms such as BFS, DFS, Greedy Best First Search, and A* in solving real-world problems.	8	**CO2**
		Self-learning Topics: Implementation of search algorithms in Python Case study: Pathfinding (Google Maps concept)	8	

III	**Knowledge Representation & Reasoning**	-Knowledge representation fundamentals. -Study of basic logical representations including propositional logic and first order logic for knowledge modeling. -Understanding inference techniques such as forward chaining and backward chaining for reasoning. -Introduction to uncertainty and probabilistic reasoning in intelligent systems. -Study of Bayesian Networks for representing and reasoning under uncertainty. -Overview of knowledge graphs for structured representation of real-world relationships.	8	**CO3**
		Self-learning Topics: Simple rule-based systems Logic programming (basic Prolog concepts)	8	
IV	**Machine Learning**	-Overview of the machine learning process including data collection, data preprocessing, model training, and evaluation. -Supervised and Unsupervised learning -Study of Linear Regression and Logistic Regression Understanding decision tree algorithm for classification and decision making. -Introduction to ensemble learning using random forest technique. -Study of K-Means algorithm for grouping similar data points. -Understanding performance metrics such as accuracy, precision, and recall. -Introduction to challenges like missing data, noise, and overfitting in machine learning.	8	**CO4**
		Self-learning Topics: Dataset handling using Pandas, Mini project: Simple prediction model	8	

V	**Natural Language Processing, Large Language Model & Reinforcement Learning**	-Fundamentals of Natural Language Processing Applications of NLP. -Overview of LLMs and their role in modern AI applications. -Understanding techniques for designing effective prompts to interact with AI models. -Introduction to using AI APIs such as OpenAI and Hugging Face for building applications. -Basic concepts of Reinforcement Learning	7	**CO5**
		Self-learning Topics: Build a simple chatbot using Python, Explore APIs like OpenAI / Hugging Face	7	
VI	**Applications of AI, System Design & Ethics**	-Study of AI applications in domains such as healthcare, banking, and retail. -Introduction to Computer Vision -Introduction to Generative AI and its applications in creating text, images, and other content. -Understanding the basic principles involved in designing AI-based systems. -Ethical considerations in AI: Study of bias, fairness, and explainability in AI systems. -Overview of practical challenges faced while developing and deploying AI solutions.	7	**CO6**
		Self-learning Topics: Case studies of AI in real companies, Responsible AI practices	7	

Text Books:

1. Artificial Intelligence: A Modern Approach – Stuart Russell & Peter Norvig
2. Machine Learning – Tom Mitchell
3. Hands-On Machine Learning with Scikit-Learn & TensorFlow – Aurelien Geron
4. Artificial Intelligence and Machine Learning – Dr. Anupam Bhatia
5. Artificial Intelligence and Machine Learning Techniques in Engineering and Management – K. Srinivasa Raju & D. Nagesh Kumar

References:

1. Artificial Intelligence – Elaine Rich & Kevin Knight
2. Introduction to Machine Learning – Ethem Alpaydin
3. Deep Learning – Ian Goodfellow
4. Artificial Intelligence – Charu C. Aggarwal
5. Artificial Intelligence Foundations and Frontiers – Sulakshana Mukherjee Banerjee
6. Machine Learning Algorithms Models and Applications – Jaydip Sen

Online References:

Sr	Module	Free Online Courses
1	Module I	https://nptel.ac.in/courses/106105077 https://www.coursera.org/learn/ai-for-everyone
2	Module II	https://web.stanford.edu/class/cs221/
3	Module III	https://nptel.ac.in/courses/106106126
4	Module IV	https://www.coursera.org/learn/machine-learning
5	Module V	https://huggingface.co/learn
6	Module VI	https://ai.google/education/

Course Code	Course Name PEC-III	Teaching Scheme (Contact Hours Per Week			Teaching Scheme (Contact Hours Per Semester)					
		L	**T**	**P**	**L**	**T**	**P**	**SL**	**Notional Learning Hour**	**Total Credits (C) (Notional Learning Hour/30**
12215704	Object-Oriented Modeling and Design	3	-	-	3	-	-	45	45	3

Course Code	Course Name	Theory					Term work	Practical	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-I	IAT-II	Total					
12215704	Object-Oriented Modeling and Design	20	20	40	60	2.5	--	--	100

Rationale:

This course introduces object-oriented concepts, analysis and design methodologies used for building scalable software systems. It focuses on UML modelling, use-case driven development, implementation strategies and design patterns which are essential for modern software engineering.

Course Objectives:

1. To understand fundamental concepts and principles of object-oriented programming and modelling.
2. To learn object-oriented software development process and analysis techniques for requirement modelling.
3. To develop the ability to identify system requirements using use case modelling and scenario analysis.

4. To design software systems using object-oriented design principles and architectural approaches.
5. To understand and create different UML diagrams for modelling software systems.
6. To study concepts of object-oriented databases and their applications in modern software systems.

Course Outcomes:

1. Explain and apply fundamental concepts of object-oriented programming, including classes, objects, inheritance, polymorphism, and encapsulation.
2. Analyze software requirements using object-oriented analysis techniques and model them effectively for system development.
3. Identify and represent system requirements through use case diagrams and scenario-based modelling techniques.
4. Design software systems using object-oriented design principles, patterns, and appropriate architectural approaches.
5. Develop and interpret various UML diagrams (such as class, sequence, activity, and state diagrams) to model software systems.
6. Describe and apply concepts of object-oriented databases and evaluate their role in modern software applications.

Prerequisite: Programming using C++ / Java, Basic Software Engineering, Data Structures.

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
I	**Overview of Object-Oriented Concepts**	Overview of Object Oriented concepts: Objects and classes, abstraction, generalization and inheritance, encapsulation, multiple inheritance, aggregation, abstract classes, polymorphism, link and association, Need for object oriented approach, Exception Handling and Memory Management concepts in Object Oriented Programming	7	CO1
		Self-learning Topics: Characteristics of Object Oriented Programming Languages, Object Identity and Object Lifecycle, Advantages and limitations of Object Oriented Programming approach	7	
II	**Object Oriented Software Development Process & Analysis**	System development life cycle, object oriented Software development process model, Object Oriented Analysis, Object Modeling Technique (OMT): object model, function model, dynamic model, relationship among models, object diagrams, state diagrams, data flow diagrams, analysis	8	CO2
		Self-learning Topics: Comparison of Structured Analysis vs Object Oriented Analysis, Agile methods in OO development, Domain Modeling concepts, Role of CASE tools in OO analysis, Real-world case study of OO analysis	8	
III	**Use Case Modelling & Requirements**	Use case modelling overview, Scenario identification, Detailed Object Oriented requirements, System sequence diagrams, Inputs and outputs identification, Behavior modeling using state charts, Integrated Object Oriented models	8	CO3
		Self-learning Topics: Use Case Relationships (Include, Extend, Generalization), Use Case-driven development, Event-based modeling, Real-life use case example (ATM / Library Management System)	8	

IV	**Object Oriented Design & Testing**	Overview of System Design, Estimating Performance, Planning for software reuse, Decomposing a system into subsystems, Identifying concurrency in system design., Choosing a software control strategy, common architectural style, CRC cards, Design Patterns, Unit Testing	8	CO4
		Self-learning Topics: Design Principles (SOLID principles), Coupling and Cohesion in software design, Client-Server Architecture, Design Metrics for software quality	8	
V	**Unified Modelling Language (UML) Diagrams**	UML overview, Class diagram, Sequence diagram, Use case diagram, Collaboration diagram, State Chart diagram, Activity diagram, Component diagram, Deployment diagram.	7	CO5
		Self-learning Topics: Package Diagram, Timing Diagram, Interaction Overview Diagram, UML in Agile Development	7	
VI	**Object Oriented Databases**	Object oriented databases, Relational vs Object Oriented database, Architecture of Object Oriented database, Query language for Object Oriented database, NoSQL Database	7	CO6
		Self-learning Topics: Object Relational Databases, ORM (Object Relational Mapping) concepts, Comparison of Object Oriented databases, Applications of Object Oriented databases	7	

Text Books:

1. "Object-Oriented Analysis and Design with Applications", 3rd Edition, Pearson Education. Grady Booch, Robert A. Maksimchuk, Michael W. Engle
2. "The Unified Modeling Language (UML) User Guide", Pearson Education. Grady Booch, James Rumbaugh, Ivar Jacobson
3. Patrick Naughton and Herbert Schildt, "Java2-Complete Reference", Tata Mc-Graw Hill.

References:

1. "Object-Oriented Modeling and Design with UML", Pearson Education. Michael Blaha and James Rumbaugh
2. "Object Oriented Analysis and Design, McGraw-Hill" Atul Kahate

Online References:

Sr	Module	Free Online Courses
1	Module I	https://onlinecourses.nptel.ac.in/noc21_cs57/preview
2	Module II	https://nptel.ac.in/courses/106105182
3	Module III	https://nptel.ac.in/courses/106106216
4	Module IV	https://onlinecourses.nptel.ac.in/noc26_cs46/preview
5	Module V	https://onlinecourses.nptel.ac.in/noc26_cs46/preview
6	Module VI	https://nptel.ac.in/courses/106105185

Course Code	Course Name PEC-III	Teaching Scheme (Contact Hours Per Week			Teaching Scheme (Contact Hours Per Semester)					
		L	T	P	L	T	P	SL	Notional Learning Hour	Total Credits (C) (Notional Learning Hour/30
12215705	Digital Signal Processing	3	-	-	45			45	90	3

Course Code	Course Name PEC-III	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration(in Hrs)			
		IAT-I	IAT-II	Total					
12215705	Digital Signal Processing	20	20	40	60	2.5	--	--	100

Rationale :

Digital Signal Processing equips students with the skills to analyze and process digital signals, design filters, and implement real-time applications using DSP processors, essential for careers in communications, control, and embedded systems.

Course Objectives:

1. To introduce the fundamentals of discrete-time signals and systems, including sampling, reconstruction, and basic signal manipulations.
2. To provide an understanding of the Discrete Fourier Transform (DFT), its properties, and its applications in signal analysis.
3. To explore the need for and implementation of the Fast Fourier Transform (FFT) for efficient spectral analysis.
4. To learn the basic design and structure of FIR and IIR filters with desired frequency responses and design digital filters.
5. To understand FFT algorithms, Multi-rate signal processing techniques and finite word length effects.
6. To introduce different digital signal processors and their real world applications.

Course Outcomes:

1. Describe discrete-time signals and systems and perform basic operations such as convolution
2. and correlation.
3. Analyze discrete signals using DFT and DTFT and apply DFT properties for signal processing.
4. Implement Radix-2 FFT algorithms and demonstrate their use in spectral analysis of signals.
5. Understand the fundamentals of general-purpose and special-purpose DSP processors and their applications.
6. Utilize DSP processors and associated hardware for implementing real-time signal processing applications in areas like radar and speech processing.

Prerequisite:

Signals and Systems, Engineering Mathematics - III

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
I	**Discrete-Time Signal and Discrete-Time Systems**	Introduction to Digital Signal Processing, Sampling and Reconstruction, Standard DT Signals, Concept of Digital Frequency, Representation of DT signal using Standard DT Signals, Signal Manipulations-shifting, reversal, scaling, addition, multiplication. Classification of Discrete-Time Signals, Classification of Discrete-Systems, LTI system, Impulse Response. Linear Convolution, Circular Convolution- Emphasis on graphical method. Linear convolution using Circular Convolution.	08	CO1
		Self-learning Topics:. Software simulation - Impulse Response, Step Response, convolution, Correlation.	08	
II	**Discrete Fourier Transform & Fast Fourier Transform:**	Definition and Properties of DFT, IDFT, Circular convolution of sequences using DFT and IDFT. Filtering of long data sequences: Overlap-Save and Overlap-Add Method for computation of DFT. Fast Fourier Transforms (FFT), Radix-2 decimation in time and decimation in frequency FFT algorithms, Inverse FFT.	07	CO2
		Self-learning Topics: Trignometric fourier transform,Applications of DFT,IDFT	07	

III	**IIR Digital Filters:**	Types of IIR Filters (Low Pass, High Pass, Band Pass, Band Stop), Analog filter approximations: Butterworth, Chebyshev . Mapping of S-plane to Z-plane, impulse invariance method, bilinear transformation method, Design of IIR digital filters (Butterworth and Chebyshev-I) from Analog filters with numerical examples. Effect of Poles and Zeros on the Frequency Response of IIR filters. Position of Poles and Zeros of Low Pass, High Pass, Band Pass, Band Stop, All Pass filters	08	CO3
		Self-learning Topics: Stability of IIR Filters, Applications of IIR Filters.	08	
IV	**FIR Digital Filters**	Characteristics of FIR digital filters, Minimum Phase, Maximum Phase, Mixed Phase and Linear Phase (Type 1 to Type 4) FIR Filters. Design of FIR filters using Window techniques (Rectangular, Hamming, Hanning, Blackman), Design of FIR filters using Frequency Sampling technique, Comparison of IIR and FIR filters.	08	CO4
		Self-learning Topics: Design and Implementation of FIR Filters.	08	
V	**Frequency Domain Analysis using DTFT and Z Transform**	Introduction to DTFT. Properties of DTFT. Z transform - definition, properties of unilateral and bilateral Z Transform, Z transform of standard signals, ROC, poles and zeros of transfer function, Inverse Z transform Analysis and characterization of LTI system using Z transform, impulse and step response, causality, stability, stability of causal system	07	CO5

		Self-learning Topics: Digital filter design using pole-zero placement, Multirate signal processing basics (decimation and interpolation).	07	
VI	**DSP Processors and Applications**	Introduction to General Purpose and Special Purpose DSP processors, fixed point and floating-point DSP processor, Computer architecture for signal processing, Harvard Architecture, Pipelining, multiplier and accumulator (MAC) Special purpose DSP hardware, Applications of DSP: Radar Signal Processing and Speech Processing,	07	CO6
		Self-learning Topics: Evolution and comparison of general-purpose vs special-purpose DSP processors, Overview of popular DSP processors,Basics of real-time operating systems (RTOS) in DSP applications.	07	

Text Books:

1. J. G. Proakis, D. G. Manolakis, “Digital Signal Processing,” Pearson Education, Fourth Edition, 2007.
2. B. Venkata Ramani and M. Bhaskar, “Digital Signal Processors, Architecture, Programming and Applications,” Tata McGraw Hill, Second Edition, 2004.
3. Alan V. Oppenheim, Ronald W. Schafer, “Discrete Time Signal Processing,” Pearson Education, Third Edition, 2021.
4. A. Anand Kumar, “Digital Signal Processing”, 2nd Edition, PHI Learning Pvt. Ltd. 2014.
5. S. Salivahanan, A. Vallavaraj, and C. Gnanapriya, “Digital Signal Processing”, 2nd Edition, Tata McGraw Hill Publication, 2011.

References:

1. Sanjit K. Mitra, “Digital Signal Processing – A Computer Based Approach”, McGraw Hill Education, Fourth Edition, 2013.

2. Tarun Kumar Rawat, “Digital Signal Processing”, Oxford University Press, First Edition, 2015.

3. Digital Signal Processing - A Practical Approach, Emmanuel C. Ifeachor and Barrie W. Jervis, 2nd Edition, Pearson Education, 2009.

4. Fundamentals of Digital Signal Processing using MATLAB – Robert J. Schilling, Sandra L. Harris, b Thomson, 2007.

Online References:

Sr. No.	Website Name
1.	https://nptel.ac.in/courses/
2.	https://swayam.gov.in
3.	https://www.geeksforgeeks.org

Course Code	Course Name	Teaching Scheme (Contact Hours Per Week			Teaching Scheme (Contact Hours Per Semester)					
		L	T	P	L	T	P	SL	Notional Learning Hour	Total Credits (C) (Notional Learning Hour/30
12412509	Open Source Technology	–	–	2	–	–	30	–	30	1

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration(in Hrs)			
		Test 1	Test 2	Total					
12412509	Open Source Technology	-	-	-	-	-	25	-	25

Rationale: Open-source operating systems, particularly Unix and Linux, form the foundation of modern computing environments, servers, cloud platforms, and software-development systems. This laboratory course provides students with practical exposure to Unix architecture, installation, command-line utilities, file and user management, process and memory management, and shell scripting. Through hands-on experiments, students develop the skills required to operate, administer, and automate tasks in an open-source environment. The course also introduces text-processing and scripting tools such as grep, sed, awk, and Perl, enabling students to perform efficient system operations and solve practical computing problems

Lab Objectives:

1	To understand architecture and installation of Unix Operating System.
2	To learn Unix general purpose commands and programming in Unix editor environment.
3	To understand file system management and user management commands in Unix.
4	To understand process management and memory management commands in
5	To learn basic shell scripting.
6	To learn scripting using awk and perl languages.

Lab Outcomes: Upon successful completion of the laboratory course, students will be able to:

1	Understand the architecture and functioning of Unix.
2	Identify the Unix general purpose commands.
3	Apply Unix commands for system administrative tasks such as file system management and user management.
4	Execute Unix commands for system administrative tasks such as process management and memory management.
5	Implement basic shell scripts for different applications.
6	Implement advanced scripts using awk and perl languages and grep, sed, etc. commands for performing various tasks.

Prerequisite: Basic Operating System Concepts, Programming fundamentals, Basic Software Engineering concepts, Basic Computer Networks

Lab Syllabus:

Sr. No.	Lab Module	Detailed Content	Hours	LO Mapping
I	Introduction to Unix	Case Study: Brief History of UNIX, Unix Architecture; Installation of Unix Operating System	4	LO1
II	Basic Commands	Execution of Unix General Purpose Utility Commands like echo, clear, exit, date, time, uptime, cal, cat, tty, man, which, history, id, pwd, whoami, ping, ifconfig, pr, lp, lpr, lpstat, lpq, lprm, cancel, mail, etc. b) Working with Editor Vi/other editor	6	LO2

III	Commands for File System Management and User Management	Study of Unix file system tree structure, file and directory permissions, single-user and multiuser environment. b) Execution of File System Management Commands like ls, cd, pwd, cat, mkdir, rmdir, rm, cp, mv, chmod, wc, piping and redirection, grep, tr, echo, sort, head, tail, diff, comm, less, more, file, type, split, cmp, tar, find, vim, gzip, bzip2, unzip, locate, etc. c) Execution of User Management Commands like who, whoami, su, sudo, login, logout, exit, passwd, useradd/adduser, usermod, userdel, groupadd, groupmod, groupdel, gpasswd, chown, chage, chgrp, chfn, etc.	6	LO3
IV	Commands for Process Management and Memory Management	Execution of Process Management Commands like ps, pstree, nice, kill, pkill, killall, xkill, fg, bg, pgrep, renice, etc. b) Execution of Memory Management Commands like free, /proc/meminfo, top, htop, df, du, vmstat, demidecode, sar, pagesize, etc.	6	LO4
V	Basic Scripts	Study of Shell, Types of Shell, Variables and Operators. Execute the following scripts, at least six.	6	LO5
VI	Advanced Scripts	Execute scripts using grep, sed, awk, and Perl.	2	LO6

Text Books:

1	S. Das, *Unix Concepts and Applications*, 4th ed., McGraw Hill, 2017.
2	*R. Michael*, Mastering Unix Shell Scripting, *2nd ed., Wiley, 2008.*
3	*D. Ambawade and D. Shah, Linux Labs and Open Source Technologies, Dreamtech Press, 2014.*

References:

1. Y. Kanetkar, Unix Shell Programming, BPB Publications, 2003.
2. B. Forouzan and R. Gilberg, Unix and Shell Programming, Cengage Learning, 2003.

Suggested Software Requirements

Ubuntu, Fedora, or Debian Linux; Oracle VirtualBox or VMware; Bash Shell; Git; GitHub; Visual Studio Code; Apache; PHP; MySQL or PostgreSQL; WordPress or Moodle; and GitHub Actions.

Suggested Practical List

Sr. No.	Laboratory Experiment	Hours	LO Mapping
1	Study the brief history, features, architecture, and major components of the Unix operating system.	2	LO1
2	Install and configure a Unix/Linux operating system on a physical machine or virtual machine.	2	LO1
3	Execute Unix general-purpose utility commands: echo, clear, exit, date, time, uptime, cal, cat, tty, man, which, history, id, pwd, and whoami.	2	LO2
4	Execute Unix network, printing, and communication commands: ping, ifconfig, pr, lp, lpr, lpstat, lpq, lprm, cancel, and mail.	2	LO2
5	Create, edit, save, search, replace, copy, delete, and navigate files using the Vi/Vim editor.	2	LO2
6	Study the Unix file-system tree structure, file and directory permissions, and single-user and multiuser environments. Execute ls, cd, pwd, cat, mkdir, rmdir, rm, cp, and mv.	2	LO3
7	Execute file permissions, filters, pipes, and redirection using chmod, wc, grep, tr, echo, sort, head, tail, diff, comm, less, more, file, type, split, and cmp.	2	LO3
8	Execute file search, archiving, and compression commands: tar, find, vim, gzip, bzip2, unzip, and locate. Perform user and group management using who, whoami, su, sudo, login, logout, exit, passwd, useradd/adduser, usermod, userdel, groupadd, groupmod, groupdel, gpasswd, chown, chage, chgrp, and chfn.	2	LO3
9	Execute process-monitoring commands such as ps, pstree, pgrep, top, and htop to observe running processes and process-related information.	2	LO4
10	Execute process-control and priority-management commands: nice, renice, kill, pkill, killall, xkill, fg, and bg.	2	LO4
11	Execute memory and storage management commands: free, /proc/meminfo, top, htop, df, du, vmstat, demidecode, sar, and pagesize.	2	LO4
12	Study shells, types of shells, variables, and operators. Write shell scripts to perform arithmetic operations, calculate simple interest, find the largest among three integers, and determine whether a given year is a leap year.	2	LO5

Sr. No.	Laboratory Experiment	Hours	LO Mapping
13	Write shell scripts to print the multiplication table of a given number using a while statement, search for an element in a list, compare two strings, and print the prescribed pattern.	2	LO5
14	Write shell scripts to check whether a file or directory exists and create it when required, implement a menu-driven calculator using the case statement, and perform prescribed directory operations.	2	LO5
15	Execute the prescribed scripts using grep, egrep, sed, awk, and Perl for file analysis, word replacement, duplicate-line identification, even-number and Fibonacci generation, array sorting, and prime-number checking.	2	LO6

The laboratory journal shall include:

- Aim and objectives of the experiment
- Required software and tools
- Commands, source code, or configuration steps
- Execution results and screenshots
- Observations and analysis
- Conclusion

The term work shall also include:

- One assignment on open-source licensing, compliance, governance, or business models.
- One case study of a major open-source project.
- One open-source contribution report or integrated mini-project.

Term Work: Term work shall consist of at least 12 practical experiments from the prescribed list.

Term Work Evaluation (25): 15 Marks for Lab Experiments + 5 Marks for Attendance + 5 Marks for Assignments/ Case Study.

Course Code	Course Name (MDM-II)	Teaching Scheme (Contact Hours Per Week			Teaching Scheme (Contact Hours Per Semester)					
		L	**T**	**P**	**L**	**T**	**P**	**SL**	**Notional Learning Hour**	**Total Credits (C) (Notional Learning Hour/30**
9931158 15	Applied Cryptography	3	--	--	45	-	-	45	90	3

(1 Credit = 30 Notional Learning Hours)

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration			
		IA T-I	IAT -II	Total					
9931158 15	Applied Cryptography	20	20	40	60	2.5	--	--	100

Rationale:

With the increasing dependence on digital communication, cloud services, e-commerce, block chain, and secure online transactions, cryptography has become a foundational technology for ensuring confidentiality, integrity, authentication, and non-repudiation in modern computing systems. Applied Cryptography focuses on practical cryptographic techniques and their implementation in real-world applications. The course introduces students to classical and modern cryptographic algorithms, cryptographic protocols, authentication systems, key management, digital signatures, public key infrastructure, and emerging cryptographic applications. It equips students with the knowledge required to design and analyze secure systems used in industries, financial systems, cloud platforms, and cybersecurity domains.

Course Objectives:

1. To understand the fundamentals of cryptography and secure communication.
2. To study classical and modern symmetric encryption algorithms.
3. To learn public key cryptography and key exchange mechanisms.

4. To understand hashing algorithms, authentication, and digital signatures.
5. To analyze cryptographic protocols and their applications in network security.
6. To introduce emerging trends in applied cryptography including blockchain and post-quantum cryptography.

Course Outcomes:

Students will be able to:

1. Explain the principles and goals of cryptography in secure systems.
2. Apply symmetric and asymmetric cryptographic algorithms for secure communication.
3. Implement hashing, message authentication, and digital signature mechanisms.
4. Analyze cryptographic protocols used in modern networks and web applications.
5. Evaluate security challenges in key management and cryptographic applications.
6. Explore emerging cryptographic technologies such as blockchain and quantum-resistant cryptography.

Prerequisite: Basic knowledge of Computer Networks, Discrete Mathematics, Number Theory, Data Structures, and Operating Systems.

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
I	**Introduction to Cryptography**	Security Goals: Confidentiality, Integrity, Authentication, Non-Repudiation, Classical Cryptography Techniques: Caesar Cipher, Play fair Cipher, Hill Cipher, Vigenere Cipher, Cryptanalysis Basics, Security Services and Attacks, Mathematical Foundations: Modular Arithmetic, Prime Numbers, Euler's Theorem, Finite Fields	07	CO1
		Self-learning Topics: History of cryptography, Real-world cryptographic failures, Applications of cryptography in banking and e-commerce	07	
II	**Symmetric Key Cryptography**	Block Cipher Principles, Feistel Structure, DES, Triple DES, AES Architecture and Working, Modes of Operation (ECB, CBC, CFB, OFB, CTR), Stream Ciphers, RC4 Overview, Performance and Security Comparison of Symmetric Algorithms	08	CO2

		Self-learning Topics: AES implementation tools, Comparative study of DES and AES, Lightweight cryptography for IoT devices	08	
III	**Public Key Cryptography**	Principles of Asymmetric Cryptography, RSA Algorithm, Diffie-Hellman Key Exchange, ElGamal Cryptosystem, Elliptic Curve Cryptography (ECC), Key Distribution and Key Management Techniques, Public Key Infrastructure (PKI)	07	CO3
		Self-learning Topics: RSA vulnerabilities, ECC applications in mobile devices, Digital certificate structure and validation	07	
IV	**Hash Functions and Authentication**	Cryptographic Hash Functions, MD5, SHA Family (SHA-1, SHA-2, SHA-3), Message Authentication Codes (MAC), HMAC, Digital Signatures, Authentication Protocols, Kerberos, Secure Password Storage and Salting Techniques	08	CO4
		Self-learning Topics: Hash collision attacks, Password cracking techniques, Authentication systems used in cloud services	08	
V	**Cryptographic Protocols and Applications**	SSL/TLS Protocols, HTTPS Security, Secure Shell (SSH), IPsec, Email Security (PGP, S/MIME), Wireless Security Protocols (WEP, WPA, WPA2, WPA3), VPN Technologies, Cryptography in Cloud Computing and IoT	07	CO5
		Self-learning Topics: TLS handshake process, VPN tunneling protocols, Cryptographic applications in mobile payment systems	07	
VI	**Emerging Trends in Applied Cryptography**	Block chain Fundamentals and Cryptographic Concepts, Cryptocurrency Security Basics, Zero Knowledge Proofs, Homomorphic Encryption, Quantum Cryptography Basics, Post-Quantum Cryptography, Ethical and Legal Issues in Cryptography	08	CO6
		Self-learning Topics: Blockchain consensus mechanisms, Quantum computing impact on RSA, Cryptography in digital identity management	08	

Text Books:

1. William Stallings, Cryptography and Network Security: Principles and Practice, Pearson.
2. Bruce Schneier, Applied Cryptography, Wiley.
3. Behrouz A. Forouzan, Cryptography and Network Security, McGraw Hill.
4. Atul Kahate, Cryptography and Network Security, McGraw Hill.

References:

1. Jonathan Katz and Yehuda Lindell, Introduction to Modern Cryptography, CRC Press.
2. Christof Paar and Jan Pelzl, Understanding Cryptography, Springer.
3. Neal Koblitz, A Course in Number Theory and Cryptography, Springer.
4. Douglas R. Stinson and Maura Paterson, Cryptography: Theory and Practice, CRC Press.
5. Charlie Kaufman, Radia Perlman, and Mike Speciner, Network Security: Private Communication in a Public World, Pearson.

Online References:

Sr	**Module**	**Free Online Courses**
1	Module I	https://nptel.ac.in/courses/106106221 https://www.coursera.org/learn/crypto
2	Module II	https://nptel.ac.in/courses/106105031
3	Module III	https://nptel.ac.in/courses/106105162 https://crypto.stanford.edu/~dabo/courses/OnlineCrypto/ https://csrc.nist.gov/projects/public-key-infrastructure
4	Module IV	https://csrc.nist.gov/projects/hash-functions https://owasp.org/www-project-cheat-sheets/
5	Module V	https://www.cloudflare.com/learning/ssl/what-is-ssl/ https://www.rfc-editor.org/rfc/rfc8446
6	Module VI	https://nptel.ac.in/courses/106105220 https://quantum-computing.ibm.com/ https://csrc.nist.gov/projects/post-quantum-cryptography

SEMESTER- VI

Course Code	Course Name	Teaching Scheme (Contact Hours Per Week			Teaching Scheme (Contact Hours Per Semester)					
		L	T	P	L	T	P	SL	Notional Learning Hour	Total Credits (C) (Notional Learning Hour/30
12211601	**System Programming and Compiler Construction**	3	-	-	45	-	-	45	90	3

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration(in Hrs)			
		IAT-I	IAT-II	Total					
12211601	**System Programming and Compiler Construction**	20	20	40	60	2.5	--	--	100

Rationale :

System software forms the backbone of computing systems by enabling interaction between hardware and application software. This course provides foundational knowledge of system-level programming components such as assemblers, macro processors, loaders, linkers, and compilers. Understanding these components is essential for designing efficient and optimized software systems. With the evolution of computing platforms, modern compiler technologies such as LLVM frameworks, Just-In-Time (JIT) compilation, runtime systems, and parallel compilation techniques have become critical in high-performance computing, cloud platforms, AI accelerators, and domain-specific languages.

Course Objectives:

1. To introduce the purpose and components of system software.
2. To study assembler and macro processor design techniques.
3. To explore loaders, linkers and their practical relevance.
4. To understand compiler analysis techniques.
5. To study synthesis phase and optimization strategies.
6. To introduce modern compiler technologies, LLVM framework and runtime systems.

Course Outcomes:

1. Recall and explain system software components and assembler design using appropriate data structures
2. Apply macro processing techniques and implement basic macro expansion using parameterized and nested macros
3. Analyze the differences between various loader and linker schemes and their role in relocation and linking.
4. Evaluate parsing strategies like LL(1), SLR, and operator precedence in syntax analysis of compilers.
5. Design and implement intermediate code structures and apply optimization techniques to improve code generation.
6. Design solutions using modern compiler frameworks such as LLVM and JIT.

Prerequisite: COA, OS, TCS

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
I	**Introduction to System Software & Assemblers**	Concept of System Software, system programming goals, overview of assembler, loader, linker, compiler, interpreter, editors and debuggers. Assembly language basics, pass structure, single-pass and two-pass assembler design for X86, data structures (SYMTAB, LITTAB, POOLTAB, MOT, POT).	7	CO1
		Self-learning Topics: RISC-V architecture fundamentals , Cross-assembler and cross-compilation, Advanced ARM/X86 addressing modes,	7	

		Inline assembly in C/C++,Virtual machines and system software in cloud platforms.		
II	**Macros and Macro Processor**	Introduction, Macro definition and call, Features of Macro facility: Simple, parameterized, conditional and nested. Design of Two pass macro processor, data structures used.	7	CO2
		Self-learning Topics: Advanced C preprocessor techniques, Recursive macro expansion, Macro systems in Lisp and Rust, Automated code generation tools, Debugging macro expansions.	7	
III	**Loaders and Linkers**	Introduction, functions of loaders, Relocation and Linking concept, Different loading schemes: Relocating loader, Direct Linking Loader, Dynamic linking and loading.	7	CO3
		Self-learning Topics: ELF and PE file formats, Static vs dynamic linking comparison, shared libraries (.so/.dll), Symbol resolution mechanisms, Runtime linking concepts.	7	
IV	**Compiler – Analysis Phase**	Introduction to compilers, Phases of compilers: Syntax Analysis- Role of Context Free Grammar in Syntax analysis, Types of Parsers: Top down parser- LL (1), Bottom up parser- SR Parser, Operator precedence parser, SLR. Semantic Analysis, Syntax directed definitions.	8	CO4
		Self-learning Topics: LR (1) and LALR parsers, Grammar ambiguity resolution, Practical ANTLR implementation, Lex & YACC tools, Error recovery techniques.	8	
V	**Compiler – Synthesis Phase**	Intermediate Code Generation: Types of Intermediate codes: Syntax tree, Postfix notation, three address codes: Triples and Quadruples, indirect triple. Code Optimization: Need and sources of optimization, Code optimization techniques: Machine Dependent and Machine Independent. Code Generation: Issues in the design of code generator, code generation algorithm. Basic block and flow graph.	8	CO5

		Self-learning Topics: Detailed Self Learning Topics: Global vs local optimization. Register allocation using graph coloring, SSA form Instruction scheduling. Loop optimization and pipelining.	8	
VI	**Modern Compiler Technologies & Advanced Runtime Systems**	Modern compiler evolution, LLVM architecture and IR, Clang, JIT vs AOT compilation, runtime systems, garbage collection strategies, loop optimization, dead code elimination, Open MP basics, DSL concepts, Web Assembly, compiler security**.**	8	CO6
		Self-learning Topics: Developing custom LLVM passes.JVM and .NET JIT internals. CUDA/OpenCL compiler frameworks. Compiler optimization for AI accelerators. Web Assembly runtime engines. Secure compilation & Control Flow Integrity	8	

Text Books:

1. D. M Dhamdhere: Systems programming and Operating Systems, Tata McGraw Hill, Revised Second Edition
2. A. V. Aho, R. Shethi, Monica Lam, J.D. Ulman: Compilers Principles, Techniques and Tools, Pearson Education, Second Edition.
3. J. J. Donovan: Systems Programming Tata McGraw Hill, Edition 1991.

References:

1. John R. Levine, Tony Mason & Doug Brown, Lex & YACC, O 'Reilly publication, second Edition
2. D, M .Dhamdhere ,Compiler construction 2e, Macmillan publication, second edition.
3. Kenneth C. Louden ,Compiler construction: principles and practices, Cengage Learning.

Online References:

Sr	Free Online Courses
1	https://nptel.ac.in/courses/106105190
2	https://www.coursera.org/courses?query=system%20programming
3	http://digimat.in/nptel/courses/video/106104123/L10.html
4	https://llvm.org/docs/LangRef.html

Course Code	Course Name	Teaching Scheme (Contact Hours Per Week			Teaching Scheme (Contact Hours Per Semester)					
		L	T	P	L	T	P	SL	Notional Learning Hour	Total Credits (C) (Notional Learning Hour/30
12211602	**Information and Network Security**	3	-	-	45	-	-	45	90	3

Course Code	Course Name	Theory					Term work	Practical / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration(in			
		IAT-I	IAT-II	Total					
12211602	Information and Network Security	20	20	40	60	2.5	–	–	100

Rationale :

With the rapid growth of digital systems, cloud computing, IoT, and online transactions, security has become a critical requirement in modern computing environments. Information and Network Security focuses on protecting data, systems, and communication networks from unauthorized access, attacks, and vulnerabilities. This course provides foundational knowledge of cryptography, authentication mechanisms, network security protocols, security policies, and ethical hacking concepts. It prepares students to design secure systems and understand real-world security challenges in industry and research domains.

Course Objectives:

1. To understand fundamental concepts of information security and network security.
2. To learn principles of cryptography and encryption techniques.
3. To study authentication, access control, and digital signatures.
4. To understand various network security threats and defense mechanisms.
5. To analyze security protocols used in modern networks.
6. To introduce security management, risk assessment, and cyber laws.

Course Outcomes:

Students will be able to:

1. Explain basic principles of information security and security models.
2. Apply symmetric and asymmetric cryptographic algorithms.
3. Implement hashing, digital signatures, and key management concepts.
4. Identify common network attacks and propose countermeasures.
5. Analyze secure communication protocols such as SSL/TLS and IPsec.
6. Evaluate security policies, risk management techniques, and ethical considerations.

Prerequisite: Basic knowledge of Computer Networks, Operating Systems, Data Structures, and Mathematical Concepts.

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
I	**Introduction to Information Security**	Security Concepts: Confidentiality, Integrity, Availability (CIA Triad), Security Goals and Services, Security Attacks: Passive and Active Attacks, Security Mechanisms, Security Models, Overview of Cyber Security	07	CO1
		Self-learning Topics: History and evolution of cyber security, Types of cyber threats in real-world scenarios, Risk vs Threat vs Vulnerability	07	
II	**Cryptography Fundamentals**	Introduction to Cryptography, Symmetric Key Cryptography, Classical Encryption Techniques, Block Ciphers and Stream Ciphers, DES, AES Overview, Asymmetric Key Cryptography, RSA Algorithm,	07	CO2

		Self-learning Topics: Mathematical foundations of cryptography, Modular arithmetic and number theory basics, Comparison of AES vs DES	07	
III	**Hash Functions and Digital Signatures**	Hash Functions and Properties, MD5, SHA Family, Message Authentication Code (MAC), Digital Signatures, Public Key Infrastructure (PKI), Certificates and Certificate Authorities	07	CO3
		Self-learning Topics: Collision resistance and hash attacks, SHA-256 working mechanism, Hashing applications, Digital certificate structure	07	
IV	**Network Security Threats and Attacks**	Network Security Overview, Sniffing and Spoofing, Man-in-the-Middle Attack, Denial of Service (DoS) and Distributed DoS, Malware: Virus, Worm, Trojan, Ransomware, Firewall Concepts, Intrusion Detection and Prevention Systems (IDS/IPS)	08	CO4
		Self-learning Topics: Real-world examples of DoS attacks, ARP spoofing and DNS poisoning, Malware analysis basics, Ransomware attack lifecycle	08	
V	**Secure Network Protocols**	Secure Socket Layer (SSL/TLS), IP Security (IPsec), Secure Shell (SSH), Email Security (PGP, S/MIME), Wireless Network Security (WPA/WPA2/WPA3), Virtual Private Networks (VPN)	08	CO5
		Self-learning Topics: Working of TLS handshake, SSL certificate verification process, VPN tunneling mechanisms, IPsec architecture (AH & ESP)	08	
VI	**Security Management and Cyber Laws**	Risk Assessment and Management, Security Policies and Procedures, Ethical Hacking Basics, Introduction to Penetration Testing, Cyber Laws and IT Act, Incident Response and Disaster Recovery, Cloud Security Overview	08	CO6
		Self-learning Topics: Risk assessment methodologies, Penetration testing lifecycle, Ethical hacking phases, Cybercrime types in India	08	

Text Books:

1. William Stallings, Cryptography and Network Security: Principles and Practice, Pearson.
2. Atul Kahate, Cryptography and Network Security, McGraw Hill.
3. Charlie Kaufman, Radia Perlman, Mike Speciner, Network Security, Prentice Hall.

References:

1. Bruce Schneier, Applied Cryptography, Wiley.
2. Behrouz A. Forouzan, Cryptography and Network Security, McGraw Hill.
3. Niels Ferguson & Bruce Schneier, Practical Cryptography, Wiley.
4. Charles P. Pfleeger, Security in Computing, Pearson.

Online References:

Sr	**Module**	**Free Online Courses**
1	Module I	https://nptel.ac.in/courses/106105162 https://nptel.ac.in/courses/106106184
2	Module II	https://nptel.ac.in/courses/106105162 https://www.rsa.com/en-us/resources/rsa-algorithm
3	Module III	https://csrc.nist.gov/projects/hash-functions https://www.openssl.org/docs/ https://csrc.nist.gov/projects/public-key-infrastructure
4	Module IV	https://owasp.org/www-project-top-ten/ https://csrc.nist.gov/publications/detail/sp/800-94/final
5	Module V	https://www.cloudflare.com/learning/ssl/what-is-ssl/ https://www.rfc-editor.org/rfc/rfc8446 https://www.openssh.com/manual.html
6	Module VI	https://www.cert-in.org.in https://www.isaca.org/resources/cybersecurity https://csrc.nist.gov/projects/risk-management

Course Code	Course Name	Teaching Scheme (Contact Hours Per Week			Teaching Scheme (Contact Hours Per Semester)					
		L	T	P	L	T	P	SL	Notional Learning Hour	Total Credits (C) (Notional Learning Hour/30
12211603	Machine Learning	3	-	-	45	-	-	45	90	3

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration(in Hrs)			
		IAT-I	IAT-II	Total					
12211603	Machine Learning	20	20	40	60	2.5	--	--	100

Rationale

Machine Learning enables computer systems to learn patterns from data and make predictions or decisions without being explicitly programmed. With the rapid growth of data-driven technologies across domains such as healthcare, finance, manufacturing, cybersecurity, and smart cities, machine learning has become an essential component of intelligent systems. This course introduces the fundamental concepts of machine learning, including learning paradigms, data preprocessing techniques, supervised and unsupervised learning algorithms, model evaluation, and ensemble methods. The course aims to develop the analytical and problem-solving abilities required to build predictive models and apply machine learning techniques to real-world problems.

Course Objectives:

1. To introduce the fundamental concepts and terminology of Machine Learning.
2. To understand various learning paradigms and data preprocessing techniques.
3. To study supervised learning algorithms used for regression and classification problems.
4. To explore unsupervised learning techniques for clustering and dimensionality reduction.

5. To evaluate machine learning models using appropriate performance metrics.
6. To understand ensemble learning techniques and their applications.

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

CO1: Explain the basic concepts and applications of Machine Learning.

CO2: Apply data preprocessing and feature engineering techniques to prepare datasets for machine learning models.

CO3: Implement supervised learning algorithms for regression and classification tasks.

CO4: Apply unsupervised learning techniques for clustering and dimensionality reduction.

CO5: Evaluate machine learning models using appropriate performance metrics.

CO6: Analyze ensemble learning methods and identify suitable machine learning models for different problem domains.

Prerequisite: Relevant applied Maths and Statistics: Data Visualization, Basic programming knowledge (preferably Python).

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
I	**Introduction to Machine Learning**	Introduction to Machine Learning: Introduction to Machine Learning, Comparison of Machine learning with traditional programming, Types of ML Problems, Statistical Learning Theory, ML vs AI vs Data Science. Types of learning: Supervised, Unsupervised, and semi-supervised, reinforcement learning techniques, Important Elements of Machine Learning- Data formats. Current ML practices- like GPT, BERT.	7	CO1
		Self-learning Topics: Applications of Machine Learning in healthcare, finance, and recommendation systems. Neural Networks.	7	

II	**Data Preprocessing and Feature Engineering**	Data collection and dataset preparation, Data preprocessing and cleaning techniques, Handling missing values, Data normalization and standardization, Concept of Data leakage. Feature Engineering (interaction and polynomial features, decomposition, target encoding), Encoding categorical variables, Feature selection methods, Data visualization for exploratory data analysis. Introduction to SMOTE.	7	CO2
		Self-learning Topics: Handling imbalanced datasets, Why is data missing ?	7	
III	**Supervised Learning & Evaluation Metrics**	Regression Techniques: Linear Regression, Multiple Linear Regression, Polynomial Regression, Regularization techniques. Classification Techniques: Logistic Regression, Naïve Bayes Classifier, Decision Trees (Ginni), K-Nearest Neighbor (KNN). Concepts, Bias–Variance tradeoff, Overfitting and Underfitting, Cross-validation techniques. Model Evaluation Metrics, Confusion Matrix, Accuracy, Precision, Recall, F1-score, ROC Curve and AUC. Hyperparameter Optimization.	8	CO3
		Self-learning Topics: Comparative analysis of classification algorithms, Logistic Regression vs KNN vs Decision Tree, to model explainability tools: SHAP, LIME.	8	
IV	**Advanced Supervised Learning and Ensemble Methods**	Support Vector Machines (SVM): Linear and Non-linear kernels, Multiclass and Multilabel Classification. Ensemble Learning Techniques: Bias–Variance tradeoff revisited, Bagging, Random Forest, Boosting, AdaBoost, XGBoost.	7	CO4
		Self-learning Topics: Advanced evaluation metrics: Log Loss, MCC, Cohen's Kappa. Hyperparameter tuning: Grid Search vs Random Search vs Bayesian Optimization.	7	

V	**Unsupervised Learning**	Clustering Techniques: Introduction to clustering, K-Means clustering, Hierarchical clustering, Density-based clustering (DBSCAN), Dimensionality Reduction- Principal Component Analysis (PCA), Kernel PCA, t-SNE, Applications: Customer segmentation, anomaly detection.	8	CO5
		Self-learning Topics: Cluster validation techniques: Silhouette Score, Davies-Bouldin Index, Advanced clustering methods: Gaussian Mixture Models (GMM).	8	
VI	**Reinforcement Learning & ML in Production**	Introduction to Reinforcement Learning-Basic concepts of Reinforcement Learning, Reinforcement Learning framework, Basics of Markov Decision Process, Q-learning (overview). ML Pipeline Building, Model Serialization and Serving, Tools, Model Drift and Maintenance, and Model Documentation.	8	CO6
		Self-learning Topics: Advanced ensemble techniques, Stacking, Blending.	8	

Text Books:

1. Bishop, Christopher M., and Nasser M. Nasrabadi, "Pattern recognition and machine learning", Vol. 4.No. 4. New York: springer, 2006.
2. Ethem Alpaydin, "Introduction to Machine Learning", PHI 2nd Edition-2013

References:

1.Tom Mitchell, "Machine learning", McGraw-Hill series in Computer Science, 1997

2. Shalev-Shwartz, Shai, and Shai Ben-David, "Understanding machine learning: From theory to algorithms", Cambridge university press, 2014.

3. Jiawei Han, Michelin Kamber, and Jian Pie, "Data Mining: Concepts and Techniques", Elsevier Publishers Third Edition, ISBN: 9780123814791, 9780123814807

4. Hastie, Trevor, et al., "The elements of statistical learning: data mining, inference, and prediction", Vol. 2. New York: springer, 2009.

5. McKinney, "Python for Data Analysis ", O' Reilly media, ISBN: 978-1-449-31979-3

Online References:

Sr	Module	Free Online Courses
1	Module 1	https://onlinecourses.swayam2.ac.in/e-learning/preview/nou26_ge65
2	Module 2	https://onlinecourses.nptel.ac.in/noc26_cs74/
3	Module 3	https://onlinecourses.nptel.ac.in/noc26_cs74/
4	Module 4	https://onlinecourses.nptel.ac.in/noc26_cs74/
5	Module 5	https://onlinecourses.nptel.ac.in/noc26_cs74/
6	Module 6	https://onlinecourses.nptel.ac.in/noc26_cs74/

Course Code	Course Name	Teaching Scheme (Contact Hours Per Week			Teaching Scheme (Contact Hours Per Semester)					
		L	T	P	L	T	P	SL	Notional Learning Hour	Total Credits (C) (Notional Learning Hour/30
12212604	**Information and Network Security**	--	--	2	--	--	30	--	30	1

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Practical/ Oral	Total
		Internal assessment			End Sem. Exam			
		IAT-I	IAT-II	Total				
12212604	**Information and Network Security**	--	--	--	--	25	25	50

Lab Objectives:

1. Demonstrate how security goals (Confidentiality, Integrity, Availability) are maintained or breached by simulating active and passive network threats.

2. Implement symmetric and asymmetric encryption algorithms (such as classical ciphers, DES/AES, or RSA) to securely transmit data.

3. Execute message hashing algorithms and digital signatures to validate data integrity and non-repudiation.

4. Analyze live network traffic using packet sniffing tools to detect unauthorized access, spoofing attacks, and firewall filter actions.

5. Examine the configuration and handshake mechanisms of secure network protocols including SSL/TLS, SSH, and IPsec.

6. Evaluate host and network systems for security vulnerabilities using basic ethical hacking and penetration testing techniques.

Lab Outcomes:

1. Differentiate between active and passive network threats in a simulated environment and establish how they impact the CIA triad.

2. Develop and execute programmatic implementations of symmetric (DES/AES) and asymmetric (RSA) cryptographic techniques to secure plain text.

3. Generate and verify message authentication codes (MACs), cryptographic hashes, and digital certificates to ensure data integrity and non-repudiation.

4. Capture and filter live network packets using tools like Wireshark to locate system vulnerabilities, identify spoofing attempts, and test firewall rulesets.

5. Trace and interpret the multi-step handshake mechanisms of secure session protocols (SSL/TLS, SSH, IPsec) during data transmission.

6. Conduct basic vulnerability scanning and system assessments on a host machine to generate security status reports using standard penetration testing tools.

List of Experiments.

Sr No	List of Experiments	Hrs
01	Implementation and performance analysis of Classical Substitution Ciphers (Caesar Cipher and Monoalphabetic Cipher).	2
02	Implementation and performance analysis of Classical Transposition Ciphers (Rail Fence Cipher and Row Transposition).	2
03	Implementation of block cipher mode operations using the Data Encryption Standard (DES) via standard programming libraries.	2
04	Implementation of secure data encryption using the Advanced Encryption Standard (AES) via standard programming libraries.	2

05	Implementation of the asymmetric RSA Cryptosystem for key generation, data encryption, and decryption routines.	2
06	Implementation of the Diffie-Hellman Key Exchange algorithm to securely establish a shared secret key over an insecure channel.	2
07	Practical execution of message hashing utilities (MD5 and SHA-256) to verify data integrity and demonstrate the avalanche effect.	2
08	Creation and verification of Digital Signatures using public-key cryptography libraries to enforce non-repudiation.	2
09	Configuration of a local Certificate Authority (CA) using OpenSSL to generate, sign, and validate digital certificates.	2
10	Analysis of unencrypted application protocols (HTTP/FTP) using the Wireshark packet sniffer to isolate plaintext credentials.	2
11	Simulation and packet-level tracing of a network-based attack (such as ARP Spoofing or a localized Denial of Service).	2
12	Configuration and validation of packet filtering access rules using firewalls (such as Linux iptables or local OS firewall suites).	2
13	Capturing and parsing an active Secure Socket Layer / Transport Layer Security (SSL/TLS) session handshake to trace sub-protocols.	2
14	Deployment of an open-source Intrusion Detection System (Snort) to compile, configure, and evaluate custom signature alert rules.	2
15	Performing active network reconnaissance, target port scanning, and host vulnerability assessment utilizing Nmap and Nessus.	2

Sr No	List of Assignments / Tutorials	Hrs.
01	**Case Study on Corporate Threat Modeling:** Detailed analysis of a notable real-world enterprise cyber breach lifecycle, identifying systemic failures across the CIA Triad.	2
02	A legal and technical study of the Information Technology (IT) Act, legal remedies for regional cybercrimes, and ethical boundaries for penetration testing.	2

Text Books:

1. William Stallings, Cryptography and Network Security: Principles and Practice, Pearson.
2. Atul Kahate, Cryptography and Network Security, McGraw Hill.
3. Charlie Kaufman, Radia Perlman, Mike Speciner, Network Security, Prentice Hall.

References:

1. Bruce Schneier, Applied Cryptography, Wiley.
2. Behrouz A. Forouzan, Cryptography and Network Security, McGraw Hill.
3. Niels Ferguson & Bruce Schneier, Practical Cryptography, Wiley.
4. Charles P. Pfleeger, Security in Computing, Pearson.

Assessment:

Term Work: Term Work shall consist of all practicals based on the above list. Also, Term work Journal must include at least 2 assignments per subject.

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

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

Course Code	Course Name	Teaching Scheme (Contact Hours Per Week			Teaching Scheme (Contact Hours Per Semester)					
		L	T	P	L	T	P	SL	Notional Learning Hour	Total Credits (C) (Notional Learning Hour/30
12212605	Machine Learning Lab	--	--	2	--	--	30	--	30	1

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Practical	Total
		Internal assessment			End Sem. Exam			
		IAT-I	IAT-II	Total				
12212605	Machine Learning Lab	--	--	--	--	25	25	50

Lab Objectives:

1. To provide hands-on experience in implementing machine learning algorithms using Python.
2. To understand and apply data preprocessing and feature engineering techniques.
3. To implement supervised and unsupervised learning models on real datasets.
4. To evaluate model performance using appropriate metrics.
5. To explore ensemble learning techniques and basic concepts of reinforcement learning.
6. To develop analytical and problem-solving skills for real-world applications.

Lab Outcomes:

After successful completion of the lab, students will be able to:
LO1: Demonstrate the use of Python libraries (NumPy, Pandas, Matplotlib, Scikit-learn) for machine learning tasks.
LO2: Perform data preprocessing including handling missing values, encoding, and scaling.
LO3: Implement regression and classification algorithms for predictive modeling.
LO4: Apply clustering and dimensionality reduction techniques.
LO5: Evaluate machine learning models using standard performance metrics.
LO6: Implement ensemble learning techniques for improved model performance.

List of Experiments.

Sr.	Experiment	Hours
00	Problem Statement Selection + Dataset Download	3 Hr
01	Python Libraries — NumPy, Pandas, Matplotlib, Seaborn + Basic Dataset Exploration	3 Hr
02	Data Cleaning & Preprocessing — Missing values, outliers, duplicates on their own dataset	2 Hrs
03	Feature Engineering & Encoding — Categorical encoding, scaling, interaction features	2 Hr
04	Exploratory Data Analysis & Visualization — Statistical + graphical insights from their domain	2 Hrs
05	Simple Linear / Logistic Regression — Baseline model + evaluation metrics	2 Hr
06	Multiple & Polynomial Regression — Compare against baseline.	2 Hr
07	Decision Tree & Naïve Bayes — Implement + compare with justification.	2 Hrs
08	KNN + Hyperparameter Analysis — Effect of K on their specific problem.	2 Hr
09	SVM with Kernel Tuning — Linear vs non-linear kernel + hyperparameter tuning.	2 Hrs
10	Clustering — K-Means & Hierarchical — Segment and interpret in domain context.	2 Hrs
11	PCA for Dimensionality Reduction — Reduce features and re-evaluate model	2 Hr
12	Ensemble Methods — Random Forest / AdaBoost — Compare with all previous models.	2 Hrs
13	ML Pipeline + Model Serialization — End-to-end pipeline, joblib/pickle, prediction interface.	2 Hrs

Sr No	List of Assignments / Tutorials	Hrs.
01	Assignment 1 — Mid-Semester Report covering problem statement, preprocessing, and baseline models.	2
02	Assignment 2 — Final Report/Research Paper covering all models, comparison table, and conclusions.	2

Text Books:

- Pattern Recognition and Machine Learning by Christopher M. Bishop, Springer, 2006; Introduction to Machine Learning by Ethem Alpaydin, PHI / MIT Press (latest edition).

References:

- Machine Learning by Tom Mitchell, McGraw-Hill, 1997; The Elements of Statistical Learning by Trevor Hastie, Robert Tibshirani, and Jerome Friedman, Springer.
- Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow by Aurélien Géron.

Online Resources:

Sr	Website Name
1.	https://nptel.ac.in/courses/106106139

Assessment:

Term Work: Term Work shall consist all practical's based on the above list. Also, Term work Journal must include at least 2 assignments per subject.

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

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

Course Code	Course Name	Teaching Scheme (Contact Hours Per Week			Teaching Scheme (Contact Hours Per Semester)					
		L	T	P	L	T	P	SL	Notional Learning Hour	Total Credits (C) (Notional Learning Hour/30
12812606	**Dissertation 1**	--	--	2	--	--	30	--	30	1

Course Code	Course Name	Examination Scheme						
		Theory Marks				Term Work	Practical/ Oral	Total
		Internal assessment			End Sem. Exam			
		IAT-I	IAT-II	Total				
12812606	**Dissertation 1**	--	--	--	--	50	50	100

Course Objectives:

1. To enable students to identify, define, and analyze real-world engineering problems through systematic literature review and stakeholder consultation.
2. To develop and apply appropriate engineering knowledge, tools, and technologies for designing feasible and efficient solutions to identified problems, with a focus on addressing United Nations Sustainable Development Goals(UNSDG's), Societal challenges, and Local/Community-based issues.
3. To promote teamwork and collaborative project management skills through effective planning, task delegation, and communication within multidisciplinary teams.
4. To encourage the integration of professional ethics, social responsibility, and sustainability principles in engineering project design and implementation.
5. To strengthen written and oral communication skills through documentation, presentations, and technical reporting of the project work.
6. To foster independent learning, critical thinking, and lifelong learning aptitude through iterative problem-solving and self-evaluation during the project lifecycle.

Course Outcomes: Learner will able

1. To develop the understanding of the problem domain through extensive review of literature.
2. To Identify and analyze the problem in detail to define its scope with problem specific data.
3. To know various techniques to be implemented for the selected problem and related technical skills through feasibility analysis.

4	To design solutions for real-time problems that will positively impact society and the environment..
5	To develop clarity of presentation based on communication, teamwork and leadership skills.
6	To inculcate professional and ethical behavior.

Guidelines:

1. Project Topic Selection and Allocation:

a) Project topic selection Process to be defined and followed:
b) Students should be informed about the domain and domain experts whose guidance can be taken before selecting projects.
c) Students should be recommended to refer papers from reputed conferences/ journals like IEEE, Elsevier, ACM, Springer, Bentham Science, MDPI, Wiley etc. which are not more than 3 years old for review of literature.
d) Students can certainly take ideas from anywhere, but be sure that they should evolve them in the unique way to suit their project requirements. Students can be informed to refer to Digital India portal, SIH portal, TIFAC, Kavach, Avishkar, Idex, Ideathon or any other hackathon portal for problem selection.

2. Topics can be finalized with respect to following criterion:

a) **Topic Selection**: The topics selected should be novel in nature (Product based, Application based or Research based) or should work towards removing the lacuna in currently existing systems.
b) **Technology Used:** Use of latest technology or modern tools can be encouraged**.**
c) Students should not repeat work done previously (work done in the last three years).
d) Project work must be carried out by the group of at least 2 students and maximum 4.
e) The project work can be undertaken in a research institute or organization/Industry/any business establishment. (out-house projects)
f) The project proposal presentations can be scheduled according to the domains and should be judged by faculty who are experts in the domain.
g) Head of department and senior staff along with project coordinators will take decisions regarding final selection of projects.
h) Guide allocation should be done and students have to submit a weekly progress report to the internal guide.
i) The internal guide has to keep track of the progress of the project and also has to maintain an attendance report. This progress report can be used for awarding term work marks.
j) In case of industry/ out-house projects, visit by internal guide will be preferred and external members can be called during the presentation at various levels

3. **Project Report Format:**

At the end of semester, each group needs to prepare a project report as per the guidelines issued by the University of Mumbai.

A project report should preferably contain at least following details:

a) Abstract
b) Introduction
c) Literature Survey/ Existing system
d) Limitation Existing system or research gap
e) Problem Statement and Objective
f) Proposed System
    a. Analysis/Framework/ Algorithm
    b. Design details
    c. Methodology (your approach to solve the problem) Proposed System
g) Experimental Set up
    a. Details of Database or details about input to systems or selected data
    b. Performance Evaluation Parameters (for Validation)
    c. Software and Hardware Set up
h) Implementation Plan for Next Semester
i) Timeline Chart for Term1 and Term-II (Project Management tools can be used.)
j) References

4. Desirable:

Students can be asked to undergo some Certification course (for the technical skill set that will be useful and applicable for projects.)

5. Requirement of Term work:

Students are required to publish one Scopus indexed conference/ present in Intercollege seminars/ National level conferences.

Course Code	Course Name PEC-IV	Teaching Scheme (Contact Hours Per Week			Teaching Scheme (Contact Hours Per Semester)					
		L	T	P	L	T	P	SL	Notional Learning Hour	Total Credits (C) (Notional Learning Hour/30
12216701	POSIX Programming	3	--	-	45	-	-	45	90	3

Course Code	Course Name	Theory					Term work	Practical / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT-I	IAT-II	Total					
12216701	POSIX Programming	20	20	40	60	2.5	--	--	100

Rationale :

The POSIX Programming Laboratory provides practical exposure to UNIX/Linux system programming using standard POSIX APIs. It helps students understand core concepts such as process management, interprocess communication, multithreading, synchronization, file handling, and socket programming. The lab emphasizes hands-on implementation to bridge the gap between theory and practice, enabling students to develop efficient and reliable system-level applications. The inclusion of mini-projects, GitHub-based version control, and debugging tools enhances industry readiness and problem-solving skills.

Course Objectives:

1. To understand the fundamentals of POSIX standards and UNIX/Linux operating system architecture.
2. To develop C programs using POSIX system calls for file, process, and memory management.
3. To implement inter-process communication mechanisms using pipes, message queues, shared memory, and semaphores.
4. To design multi-threaded applications using POSIX threads (Pthreads).
5. To analyze synchronization issues such as race conditions and deadlocks.
6. To build real-time and concurrent applications in a Linux environment.

Course Outcomes:

Students will be able to:

1. Use POSIX APIs to perform file and directory operations.
2. Develop process-based programs using fork(), exec(), wait(), and signals.
3. Implement IPC mechanisms using pipes, shared memory, semaphores, and message queues.
4. Develop multi-threaded programs using POSIX threads.
5. Apply synchronization techniques to solve concurrency problems.
6. Design and test concurrent applications in Linux.

Prerequisite:

Operating Systems, C Programming, Data Structures

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
I	Introduction to POSIX & UNIX Architecture	Overview of UNIX/Linux Architecture ,History and Need for POSIX ,POSIX Standards and Compliance, System Calls Vs Library Functions ,Basic Shell Commands ,File System Structure in Linux ,Compilation using GCC	7	CO1
		Self-learning Topics: Comparison between POSIX and Windows API. , Study of different UNIX variants (Linux, BSD, macOS). , Linux kernel architecture overview. , Writing custom shell scripts for automation. , Exploring system call tracing using strace. ,Understanding ELF file format basics.	7	
II	File and Directory Management	File Descriptors ,open (), close (), read (), write (), lseek () ,File Permissions and Access Modes ,Stat (), fstat () ,Directory Handling: opendir (), readdir(), closedir() ,File Locking Mechanisms ,Error Handling using errno and perror	7	CO1
		Self-learning Topics: Advanced file control using fcntl (). ,Memory-mapped files using mmap (). ,Implementation of custom file copy utility (mini cp command). ,File system types in Linux (ext4, xfs, etc.). ,Disk I/O performance measurement. ,Implementing file encryption using system calls.	7	
III	Process Management	Process Concept and Process States ,Fork (), vfork() ,exec Family of Functions ,wait (), waitpid() ,Process Control and Termination ,Orphan and Zombie Processes ,Environment Variables	7	CO2
		Self-learning Topics:	7	

		Implementing a mini shell using fork () and exec ()., Process scheduling policies in Linux (CFS scheduler). , CPU affinity and process priority (nice, renice). , Daemon process creation., Exploring /proc file system., Process monitoring tools (top, htop, ps).		
IV	Signals and Inter-Process Communication	Signal Concepts, Signal (), sigaction(), Kill (), alarm(), pause(), Pipes (Unnamed and Named – FIFO), Message Queues, Shared Memory, Semaphores, Comparative Study of IPC Mechanisms	8	CO3
		Self-learning Topics: Advanced signal handling using sigaction()., Comparing POSIX IPC vs System V IPC., Implementing chat application using named pipes., Shared memory performance analysis., Signal masking and pending signals., Implementing inter-process synchronization using semaphores.	8	
V	POSIX Threads (Pthreads)	Thread Concepts, pthread_create(), pthread_exit(), pthread_join(), pthread_cancel(), Thread Attributes, Thread Synchronization (Mutex, Condition Variables), Thread vs Process Comparison	8	CO4
		Self-learning Topics: Thread pool implementation., Multithreaded matrix multiplication., Performance comparison: Single-thread vs Multi-thread., Thread-safe data structures., Thread attributes and scheduling policies., Deadlock demonstration using threads.	8	
VI	Synchronization & Advanced Concepts	Critical Section Problem, Race Conditions, Deadlocks (Detection & Prevention), Reader-Writer Problem, Producer-Consumer Problem, Introduction to Real-Time POSIX Extensions	8	CO5, CO6
		Self-learning Topics: Study of POSIX real-time extensions., Implementing Dining Philosophers Problem., Banker's Algorithm simulation., Real-time scheduling policies in Linux., Spinlocks vs Mutex comparison., Lock-free programming basics.	8	

Text Books:

1. W. Richard Stevens & Stephen A. Rago, **Advanced Programming in the UNIX Environment**, Addison-Wesley.
2. Michael Kerrisk, **The Linux Programming Interface**, No Starch Press.
3. Brian W. Kernighan & Dennis M. Ritchie, **The C Programming Language**, Prentice Hall.

References:

1. W. Richard Stevens, **UNIX Network Programming**, Prentice Hall.
2. Maurice J. Bach, **The Design of the UNIX Operating System**, Prentice Hall.
3. Robert Love, **Linux System Programming**, O'Reilly Media.

Course Code	Course Name PEC-IV	Teaching Scheme (Contact Hours Per Week			Teaching Scheme (Contact Hours Per Semester)					Total Credits (C) (Notional Learning Hour/30
		L	T	P	L	T	P	SL	Notional Learning Hour	
12216702	Advance Computer Network	3	-	-	45	-	-	45	90	3

Course Code	Course Name	Theory					Term work	Practical / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs.)			
		IAT-I	IAT-II	Total					
12216702	Advance Computer Network	20	20	40	60	2.5	--	--	100

Rationale:

This course provides advanced knowledge of computer networking including data link, routing, transport (TCP/UDP), and application protocols (DNS, HTTP) for efficient and scalable communication systems. It aligns with NEP 2020 by emphasizing theoretical understanding supported by case studies, simulations, and analytical learning.The course focuses on wireless, mobile, multimedia, and emerging networks (MANET, WSN) and supports United Nations Sustainable Development Goals (SDG 4: Quality Education, SDG 9: Industry, Innovation and Infrastructure, and SDG 11: Sustainable Cities and Communities) by preparing students to design reliable and future-ready network solutions. Guided by Bloom's Taxonomy, it develops skills from understanding to analysis, design, and evaluation of modern communication networks.

Course Objectives:

1. **Understand** advanced data link, wireless, and routing protocols used in modern communication networks.
2. **Analyze** transport and application layer mechanisms for reliable and efficient data communication.
3. **Explore** web technologies, DNS, and network programming concepts for building network-based applications.
4. **Examine** wireless and mobile networking principles including mobility management and cellular communication.
5. **Study** multimedia networking and overlay architectures for real-time and scalable communication.
6. **Investigate** ad-hoc and sensor networks focusing on routing strategies and energy-efficient communication.

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

1. Apply data link protocols, wireless networking concepts, and advanced routing algorithms to design efficient network solutions.
2. Analyze transport layer protocols (TCP/UDP) and evaluate their performance for reliable data transmission.
3. Develop basic network applications using socket programming and understand web-based communication (HTTP, DNS, FTP).
4. Evaluate wireless and mobile network performance, including mobility management and cellular systems.
5. Analyze multimedia networking protocols and design overlay network-based solutions such as VoIP and streaming systems.
6. Apply routing and energy-efficient techniques in ad-hoc and wireless sensor networks to solve real-world problems.

Prerequisite: Data Communication, Basic Networking Principles, Computer Networks

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
I	Data Link Protocols,	Data-link protocols: Ethernet, Token Ring and Wireless (802.11). Wireless Networks and Mobile IP: Infrastructure of Wireless Networks, Wireless LAN Technologies, IEEE 802.11 Wireless Standard, Cellular Networks, Mobile IP, Wireless Mesh Networks (WMNs), Multiple access schemes. **Routing and Internetworking:** Network–Layer Routing,Least-Cost-Pathalgorithms,Non-Least-CostPath algorithms, Intra-domain Routing Protocols, Inter-domain Routing Protocols, Congestion Control at Network Layer.	8	CO1
		Self-learning Topics:.Configure basic routing using **Cisco Packet Tracer / GNS3** Implement shortest path algorithm (Dijkstra) in Python Study and compare ,**OSPF vs RIP** using simulation Analyze Wi-Fi network parameters using mobile apps (signal strength, channels),Perform packet capture of Ethernet frames using **Wireshark,** Create a small report on real-world ISP routing (India-based ISP like Jio/Airtel).	8	
II	Transport and Application Layer Protocols:	Transport and Application Layer Protocols: Client-Server and Peer-To-Peer Application Communication, Protocols on the transport layer, reliable communication. Routing packets through a LAN and WAN. Transport Layer, Transmission Control Protocol (TCP), User Datagram Protocol (UDP), Mobile Transport Protocols, TCP Congestion Control. Principles of Network Applications	8	CO2
		Self-learning Topics: Write a basic client-server program using TCP sockets (Python/Java), Observe TCP handshake and congestion control in Wireshark, Analyze differences between HTTP and HTTPS traffic.	8	

III	Web, DNS & Network Programming	The Web and HTTP, File Transfer: FTP, Electronic Mail in the Internet, Domain Name System (DNS), P2P File Sharing, Socket Programming with TCP and UDP, building a Simple Web Server Creating simulated networks and passing packets through them using different routing techniques. Installing and using network monitoring tools.	8	CO3
		Self-learning Topics: Capture HTTP request/response packets using Wireshark, Create a basic file transfer program using FTP concepts, Simulate a network using Mininet or Cisco Packet Tracer.	8	
IV	Wireless and Mobile Networks	Wireless and Mobile Networks: Introduction, Wireless links and Network Characteristics - CDMA, Wifi: 802.11 Wireless LANS, Cellular internet access, Mobility management: Principles	7	CO4
		Self-learning Topics: Measure Wi-Fi signal strength at different locations, Study mobile network types (3G/4G/5G) using your smartphone, Analyze handoff behavior while moving between networks, Compare Wi-Fi vs mobile data performance (speed test analysis), Study different wireless standards (802.11 variants) and prepare comparison chart, Case study on 5G deployment in India	7	
V	Multimedia Networking & Overlay Networks	Multimedia Networking: Multimedia networking applications, streaming stored video, Voice-over-IP, Protocols for real-time conversational applications.	7	CO5
		Self-learning Topics: Study how VoIP works using apps (e.g., WhatsApp call analysis), Compare streaming quality under different network speeds, Observe RTP/RTCP packets using Wireshark (if possible), Study CDN working by accessing same website from different locations, Prepare a report on Netflix/YouTube streaming architecture.	7	
VI	Ad-hoc and Sensor Networks	**Mobile A-Hoc Networks:** Overview of Wireless Ad-Hoc Networks, Routing in Ad-Hoc Networks, Routing Protocols for Ad-Hoc Networks – **Wireless Sensor Networks:** Sensor Networks and Protocol Structures, Communication Energy Model,Clustering Protocols, Routing Protocols	7	CO6

		Self-learning Topics: Simulate MANET routing using NS2/NS3 (basic level) Study real-life sensor network applications (smart city, agriculture) Create a small IoT demo using Arduino/Raspberry Pi (if available) Analyze energy consumption concepts in WSN through case study Compare routing protocols (AODV, DSR) theoretically Mini-project: Design a simple smart sensor-based system	7	

Text Books:

1. Computer Networking: A Top-Down Approach, James F. Kurose, Keith W. Ross,8th Edition, Pearson.
2. Computer and Communication Networks, Nader F. Mir, Pearson Education
3. High-Speed Networks and Internets: Performance and Quality of Service, William Stallings,2nd Edition, Pearson

References:

1. Data Communications and Networking, Behrouz A. Forouzan,5th Edition, McGraw-Hill
2. Computer Networks, Andrew S. Tanenbaum, David J. Wetherall,5th Edition, Pearson.
3. An Engineering Approach to Computer Networking, S. Keshav, Pearson.
4. Internetworking with TCP/IP Volume 1, Douglas E. Come,6th Edition, Pearson.

Online References:

Sr.	**Module**	**Free Online Courses**
1	Module I	https://see.stanford.edu/Course/CS107 https://ocw.mit.edu/collections/introductory-programming/ https://see.stanford.edu/Course/CS106A Principles of programming Languages, https://nptel.ac.in/courses/106102067
2	Module II	
3	Module III	
4	Module IV	Learn You Haskell For Great Good, https://learnyouahaskell.com/chapters https://www.coursera.org/learn/scala-functional-programming

5	Module V	Online Prolog Learning Resources, https://swish.swi-prolog.org/example/examples.swinb https://www.coursera.org/learn/logic-introduction https://www.coursera.org/learn/an-introduction-to-logic-in-computer-science https://ocw.mit.edu/courses/6-001-structure-and-interpretation-of-computer-programs-spring-2005/resources/8a-logic-programming-part-1/
6	Module VI	Book Companion Resource : Dart Apprentice: Beyond the Basics (First Edition): Object-Oriented Programming, Concurrency & More https://github.com/kodecocodes/dabb-materials/tree/editions/1.0 https://see.stanford.edu/Course/CS107 https://ocw.mit.edu/courses/6-001-structure-and-interpretation-of-computer-programs-spring-2005/resources/8a-logic-programming-part-1/

Course Code	Course Name PEC-IV	Teaching Scheme (Contact Hours Per Week			Teaching Scheme (Contact Hours Per Semester)					
		L	T	P	L	T	P	SL	Notional Learning Hour	Total Credits (C) (Notional Learning Hour/30
12216703	Natural Language Processing	3	-	-	45	-	-	45	90	3

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration(in Hrs)			
		IAT-I	IAT-II	Total					
12216703	Natural Language Processing	20	20	40	60	2.5	--	--	100

Rationale :

Natural Language Processing (NLP), a core area of Artificial Intelligence, enables machines to understand and process human language. With advancements in Deep Learning, modern NLP uses neural and transformer-based models for improved accuracy. This course covers both classical and modern techniques to build real-world applications like chatbots and translation systems. It also addresses challenges such as ambiguity, bias, and ethical concerns, preparing students for industry-relevant NLP solutions.

Course Objectives:

1. To understand fundamental concepts and evolution of Natural Language Processing including modern deep learning approaches.
2. To analyze different levels of language processing: Morphology, Syntax, Semantics, Pragmatics, and Discourse.
3. To design and implement statistical and neural language models for text processing.

4. To apply advanced NLP techniques including word embeddings and transformer-based models.
5. To develop real-world NLP applications such as chatbots, sentiment analysis, and machine translation.
6. To evaluate, optimize, and address ethical challenges in modern NLP systems.

Course Outcomes:

1. Explain fundamental and advanced concepts of NLP including neural approaches.
2. Apply statistical and neural language models for text representation and processing.
3. Implement POS tagging and parsing using both classical and machine learning techniques.
4. Analyze semantic representations using embeddings and disambiguation techniques.
5. Design discourse-level models including coreference resolution and contextual understanding.
6. Develop and evaluate real-world NLP applications using modern frameworks and transformer models.

Prerequisite: Theory of Computer Science, System Programming &Compiler Construction

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
I	**Introduction to NLP**	Origin & History, Stages of NLP, Ambiguity, NLP pipeline, Applications, Evolution from Rule-based → Statistical → Neural NLP, Introduction to Generative AI & LLMs, Challenges (bias, ethics)	7	CO1
		Self-learning Topics: NLP tools for Indian languages, Overview of LLMs, Prompt engineering basics, Noisy channel model, Edit distance, Implementation of embeddings, Text preprocessing using Python libraries	7	
II	Word Level Analysis	Tokenization, Stemming, Lemmatization, Morphology, Regular Expressions, FST, N-grams & limitations, Smoothing, Perplexity, Word Embeddings (Word2Vec, GloVe, FastText), Subword Tokenization (BPE, WordPiece), Neural Language Models	8	CO2

		Self-learning Topics: Noisy channel model, Edit distance, Implementation of embeddings, Text preprocessing using Python libraries	8	
III	Syntax Analysis	POS Tagging (Rule-based, HMM, CRF), Parsing (Top-down, Bottom-up, CYK, PCFG), Dependency Parsing, Neural POS tagging (BiLSTM), Intro to Transformer encoder	7	CO3
		Self-learning Topics: Parser evaluation techniques, Treebanks for Indian languages, Syntax parsing using libraries	7	
IV	Semantic Analysis	Meaning representation, Lexical semantics, WordNet, WSD (Lesk, Naïve Bayes), Contextual Embeddings (BERT concept), Sentence embeddings, Transformer-based semantics	8	CO4
		Self-learning Topics: Distributional semantics, Topic modeling (LDA), Pretrained embeddings usage	8	
V	Pragmatics & Discourse	Discourse processing, Reference & Anaphora resolution, Coherence, Coreference resolution using ML, Dialogue systems basics, Context handling in chatbots	7	CO5
		Self-learning Topics: Discourse segmentation, Conversational AI design, Chatbot frameworks	7	
VI	Applications of NLP	Machine Translation, Sentiment Analysis, Information Retrieval, Question Answering, Chatbots & Conversational AI, Transformer-based applications, Intro to tools (SpaCy, Hugging Face)	8	CO6
		Self-learning Topics: LSTM, RNN basics, Transformer models (BERT, GPT concepts)	8	

Text Books:

1. Daniel Jurafsky and James H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*, 3rd Edition
2. Jacob Eisenstein, *Introduction to Natural Language Processing*, MIT Press, 2019
3. Delip Rao and Brian McMahan, *Natural Language Processing with PyTorch*, O'Reilly Media, 2019

References:

1. Lewis Tunstall, Leandro von Werra, Thomas Wolf, *Natural Language Processing with Transformers*, O'Reilly Media, 2022
2. Denis Rothman, *Transformers for Natural Language Processing*, Packt Publishing, 2022
3. Shashank Mohan Jain, *Introduction to Transformers for NLP*, Apress, 2022

Online References:

Sr. No.	**Website Name**
1.	https://onlinecourses.nptel.ac.in/noc23_cs45/preview
2.	https://huggingface.co/learn/nlp-course/chapter1/1
3.	https://www.edx.org/learn/natural-language-processing

Course Code	Course Name PEC-IV	Teaching Scheme (Contact Hours Per Week			Teaching Scheme (Contact Hours Per Semester)					
		L	T	P	L	T	P	SL	Notional Learning Hour	Total Credits (C) (Notional Learning Hour/30
12216704	Parallel Algorithms	3	-	-	45	-	-	45	90	3

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration(in Hrs)			
		IAT-I	IAT-II	Total					
12216704	Parallel Algorithms	20	20	40	60	2.5	--	--	100

Rationale:

The syllabus aims to develop the ability to design, analyze, and implement algorithms that efficiently utilize multiple processors to solve large and complex computational problems.

The rationale for offering Parallel Algorithms alongside Computer Graphics and Virtual Reality lies in the increasing computational demands of modern graphical systems and immersive environments. Computer graphics involve complex tasks such as rendering, animation, shading, and real-time simulations, all of which require high-performance computing. Parallel algorithms enable these tasks to be executed efficiently by distributing computations across multiple processors, significantly improving speed and scalability. This synergy helps students understand how large graphical datasets and real-time VR applications are managed in practice.

Course Objectives:

1. To understand the fundamental concepts of computer graphics and virtual reality.
2. To learn algorithms for generating graphical primitives such as lines, circles, and polygons.
3. To understand and apply 2D and 3D transformations, viewing, and projection techniques.
4. To study illumination models, shading techniques, and hidden surface removal methods.
5. To understand the concepts of animation and virtual reality systems and their applications.
6. To develop basic graphical applications using standard graphics tools and APIs.

Course Outcomes: Learners will be able to:

1. Understand different parallel processing approaches and platforms involved in Achieving Performance.
2. Understand design Issues and limitations in Parallel Algorithms.
3. Understand and enable matrix multiplication and solving linear systems.
4. Analyze and optimize performance parameters.
5. Understand and enabled combinational search methods for parallel algorithms.
6. Learn to programming using message passing paradigm using opensource APIs, Design and implement parallel programs for multicore systems using OpenMP, MPI, and introductory GPU frameworks

Prerequisite: Computer Network, Advance operating system, Microprocessor

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
I	**Introduction to Parallel Algorithms**	History. Parallel architecture, simple parallel arrays, processor Array, Parallel architecture, multiprocessor. Parallel algorithm design Elements, task/channel models. Fosters design methodology	07	CO1
		Self-learning Topics:. Flynn's Taxonomy of Parallel Computer Architecture, Basics of OpenMP / MPI for Parallel Programming.	07	

II	**Parallel Algorithm Design**	Introduction Boundary value problems: - partition, communication, mapping analysis , finding the maximum, N-body problem, adding data inputs, Ford algorithm	07	CO2
		Self-learning Topics:. Parallel Prefix Sum (Scan) Algorithm,Parallel Sorting Algorithms (Bitonic Sort / Parallel Merge Sort)	07	
III	**Matrix Multiplication**	Introduction, Sequential matrix multiplication: - Iterative Row-oriented algorithm, recursive blockoriented Algorithm,Row-wise block stripped parallel algorithm, Cannon Algorithm	08	CO3
		Self-learning Topics: Strassen Algorithm for Matrix Multiplication ,Fox Algorithm for Parallel Matrix Multiplication	08	
IV	**Solving linear System**	Introduction, terminology, back substitution ,sequential algorithm, row-orientation parallel algorithm, Gaussian elimination,-sequential parallel, row-orientation ,column -orientation algorithm, Parallel Gaussian elimination (row-oriented and column-oriented), introduction to hybrid programming models (MPI + OpenMP) for solving linear systems, basic introduction to GPU computing for matrix operations, and a brief overview of parallel sorting (quick sort, bucket sort)	08	CO4
		Self-learning Topics: LU Decomposition Method Parallel Merge Sort Algorithm	08	
V	**Combinational Search**	Introduction, Divide and conquer, parallel back track search ,Branch and bound, searching game trees, parallel alpha-beta search :-parallel algorithm search, parallel binary tree, parallel graph tree ,technologies, Firewalls and Routers,	08	CO5
		Self-learning Topics: Minimax Algorithm for Game Tree Search ,Intrusion Detection System (IDS) in Network Security	08	

VI	**Programming using Message Passing Paradigm**	Principles, building blocks, MPI, Overlapping communication and computation, collective communication operations, Composite synchronization constructs, OpenMP Threading Building blocks; An Overview of Memory Allocators, Parallel programming model, combining MPI and OpenMP, Shared memory programming Installation and Configuration.	07	CO6
		Self-learning Topics: Basics of CUDA Programming , P threads (POSIX Threads) for Parallel Programming	07	

Text Books:

1. Michael J. Quinn, "Parallel Programming in C with MPI and OpenMP", McGraw-Hill

2. Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar , "Introduction to Parallel Computing", Pearson Education, Second Edition, 2007.

References:

1. Laurence T. Yang, Minyi Guo, "High- Performance Computing: Paradigm and Infrastructure" Wiley, 2006.

2. Kai Hwang, Naresh Jotwani, "Advanced Computer Architecture: Parallelism, Scalability, Programmability", McGraw Hill, Second Edition, 2010.

Online References:

Sr	**Module**	**Free Online Courses**
1	Module I	https://cse.iitkgp.ac.in/~debdeep/courses_iitkgp/PAlgo21 https://www.ques10.com/p/36530/explain-the-various-types-of-parallel-programming-/
2	Module II	
3	Module III	
4	Module IV	https://www.spiceworks.com/tech/iot/articles/what-is-parallel-processing/
5	Module V	https://www.coursera.org/learn/scala-parallel-programming
6	Module VI	https://www.coursera.org/learn/scala-parallel-programming

Course Code	Course Name PEC-IV	Teaching Scheme (Contact Hours Per Week			Teaching Scheme (Contact Hours Per Semester)					
		L	T	P	L	T	P	SL	Notional Learning Hour	Total Credits (C) (Notional Learning Hour/30
12216705	**Image Processing**	3	-	-	45	-	-	45	90	3

Course Code	Course Name	Theory					Term work	Practical / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration (in Hrs)			
		IAT -I	IAT -II	Total					
12216705	**Image Processing**	20	20	40	60	2.5	--	--	100

Rationale:

The **Image Processing** course provides fundamental knowledge of digital image processing techniques and their practical applications. It covers essential concepts such as image acquisition, enhancement, restoration, segmentation, and compression, enabling students to analyze and process visual data effectively.

Course Objectives:

1. To introduce the fundamental concepts and principles of digital image processing.
2. To develop the ability to apply image enhancement and restoration techniques for improving image quality.
3. To understand and implement image segmentation and feature extraction methods.
4. To familiarize students with image compression and recognition techniques.
5. To expose learners to real-world applications of image processing in various domains.

6. To build analytical and problem-solving skills for designing image processing solutions.

Course Outcomes:

1. Know and understand the basics and fundamentals of digital image processing, such as digitization, sampling, quantization, and 2D-transforms.
2. Operate on images using the techniques of smoothing, sharpening and enhancement.
3. Understand the restoration concepts and filtering techniques.
4. Learn the basics of segmentation, features extraction, compression and recognition methods for color models.
5. Comprehend image compression concepts.
6. To demonstrate real time image and video analytics applications.

Prerequisite:

Linear Algebra (matrices, vectors), Probability & Statistics, Continuous & discrete signals, C / Python / MATLAB basics, Sensors and imaging devices

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
I	**Digital Image Fundamentals**	Steps in Digital Image Processing – Components – Elements of Visual Perception – Image Sensing and Acquisition – Image Sampling and Quantization – Relationships between pixels - Color image fundamentals - RGB, HSI models, Two-dimensional mathematical preliminaries, 2D transforms - DFT, DCT.	7	**CO1**
		Self-learning Topics: Introduction to Digital Image Processing tools**,** Image file formats, Human visual system in detail, Applications of DFT and DCT in real-world systems	7	
II	**IMAGE ENHANCEMENT**	Spatial Domain: Gray level transformations – Histogram processing – Basics of Spatial Filtering– Smoothing and Sharpening Spatial Filtering, Frequency Domain: Introduction to Fourier Transform– Smoothing and Sharpening frequency domain filters – Ideal, Butterworth and Gaussian filters, Homomorphic filtering, Color image enhancement.	8	**CO2**

		Self-learning Topics: Contrast stretching techniques, Histogram equalization, Edge enhancement techniques, Noise reduction techniques in real images	8	
III	**IMAGE RESTORATION**	Image Restoration - degradation model, Properties, Noise models – Mean Filters – Order Statistics – Adaptive filters – Band reject Filters – Band pass Filters – Notch Filters – Optimum Notch Filtering – Inverse Filtering – Wiener filtering	8	**CO3**
		Self-learning Topics: Types of noise (Gaussian, Salt & Pepper, Speckle), Blind vs Non-blind deconvolution, Motion blur removal techniques, Deep learning-based image de-noising (intro)	8	
IV	**IMAGE SEGMENTATION**	Edge detection, Edge linking via Hough transform – Thresholding - Region based segmentation – Region growing – Region splitting and merging – Morphological processing- erosion and dilation, Segmentation by morphological watersheds – basic concepts – Dam construction – Watershed segmentation algorithm.	7	**CO4**
		Self-learning Topics: K-means clustering for segmentation, Graph-based segmentation methods, Deep learning segmentation (CNN, U-Net basics)	7	
V	**IMAGE COMPRESSION AND RECOGNITION**	Need for data compression, Huffman, Run Length Encoding, Shift codes, Arithmetic coding, JPEG standard, MPEG. Boundary representation, Boundary description, Fourier Descriptor, Regional Descriptors – Topological feature, Texture - Patterns and Pattern classes - Recognition based on matching.	7	**CO5**
		Self-learning Topics: Lossy vs Lossless compression comparison, PNG compression techniques, Wavelet-based compression (JPEG2000 basics), Feature extraction techniques (SIFT, SURF – intro)	7	

VI	**APPLICATIONS AND CASE STUDIES**	Industrial- Transportation & Travel- Remote Sensing - Surveillance. IoT Video Analytics Architectures, Medical Image Processing Applications (X-Ray, CT Scan), Emerging Applications - Autonomous vehicles and computer vision, Augmented Reality (AR) and Virtual Reality (VR), Facial recognition and biometrics, Agriculture (crop monitoring, disease detection), Case Study-Face detection system. Introduction to Deep Learning in Image Processing	8	**CO6**
		Self-learning Topics: Smart manufacturing (Industry 4.0), Vehicle detection and tracking, scan preprocessing techniques, Crop disease detection using images	8	

Text Books:

1. Rafael C. Gonzalez, Richard E. Woods, 'Digital Image Processing', Pearson, Third Edition,2010.
2. Anil K. Jain, 'Fundamentals of Digital Image Processing', Pearson, 2002.
3. Wilhelm, Mark J.Burge, "Digital Image Processing An Algorithmic Introduction", Springer; 3rd edition, 2022.

References:

1. Kenneth R. Castleman, 'Digital Image Processing', Pearson, 2006.
2. Rafael C. Gonzalez, Richard E. Woods, Steven Eddins, 'Digital Image Processing using MATLAB', Pearson Education, Inc., 2011.
3. D,E. Dudgeon and RM. Mersereau, 'Multidimensional Digital Signal Processing', Prentice Hall Professional Technical Reference, 1990.
4. William K. Pratt, 'Digital Image Processing', John Wiley, New York, 2002
5. Milan Sonka, Vaclav Hlavac, Roger Boyle, "Image Processing, Analysis, and MachineVision", 4nd edition, Thomson Learning, 2013.
6. A.K. Jain, "Fundamentals of Digital Image Processing", PHI, New Delhi, 2015.
7. Rick Szelisk, "Computer Vision: Algorithms and Applications", Springer 2011.

Course Code	Course Name	Teaching Scheme (Contact Hours Per Week			Teaching Scheme (Contact Hours Per Semester)					Total Credits (C) (Notional Learning Hour/30
		L	T	P	L	T	P	SL	Notional Learning Hour	
98451612	**Professional Communication & Ethics**	2	-	-	30	-	-	30	60	2

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration(in Hrs)			
		Test 1	Test 2	Avg.					
98451612	Professional Communication and Ethics	-	-	-	-	-	25	-	25

Rationale :

The syllabus is designed to help students face the corporate requirement with respect to interpersonal skills, resume drafting, presentation giving and to crack the interviews.

Course Objectives:

1) To develop students interpersonal and professional communication skills.
2) To train in group discussion techniques and collaborative communication for selection processes and teamwork.
3) To prepare effective resumes and perform confidently in interviews.
4) To train students to answer confidently and cracked HR interview.
5) To build competence in drafting notices, agendas and minutes of the meetings.
6) To enhance the ability to deliver structured presentations and prepare professional reports.

Course Outcomes:

CO1: Explain the concepts and importance of interpersonal skills in professional environments.

CO2: Describe the structure, types, and evaluation criteria of group discussions.
CO3: Explain the principles and components of effective resume writing.
CO4: Describe interview processes, techniques, and professional etiquette.
CO5: Explain formats and principles of formal communication including notices and meetings.
CO6: Describe the structure and techniques of presentations and report writing.

Prerequisite: Professional Communication & Ethics

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
I	**Interpersonal Skills**	Definition, importance, and scope of interpersonal skills, Components of interpersonal skills: Emotional Intelligence (self-awareness, self-regulation, motivation, empathy, social skills), Leadership and teamwork concepts, Communication in teams: barriers and effectiveness, Empathy and assertiveness: meaning and significance, Conflict management: types, causes, and resolution strategies	5	CO1
		Self-Learning Topics: Self-assessment of interpersonal skills via SWOT analysis	5	
2.	**Group Discussion**	Purpose of Group Discussion in recruitment and teamwork, Types of GD: Topic-based, case-based, abstract, activity-based, Technical vs non-technical GD, Structure of GD: initiation, discussion flow, conclusion, Evaluation parameters: clarity, content, communication skills, leadership, GD etiquette: turn taking, listening, body language, summarizing, Do's and Don'ts in GD.	5	CO2
		Self-Learning Topics: Latest GD topics via (Current affairs, social issues, technology)	5	
3.	**Resume building**	Purpose and importance of resume and CV, Types of resumes: chronological, functional, combination, Key sections of a resume: Objective/Summary, Education, Skills, Projects and achievements, Principles of effective resume writing: clarity, conciseness, relevance, Resume customization for different job roles.	5	CO3

		Self-Learning Topics: Industry specific resume formats (IT, Core, Engineering, Managements).	5	
4.	**HR / Personal Interviews**	Meaning and types of interviews (HR, technical, behavioural), Common HR interview questions and expected responses, STAR method for answering behavioural questions, Interview etiquette: dress code, body language, communication style, Handling stress questions and tricky situations, Basic interview preparation strategies.	5	CO4
		Self-Learning Topics: Frequently asked HR and behavioral interview questions.	5	
5.	**Notice, Agenda and Meeting**	Features of formal English: tone, clarity, precision, Types of notices : circulars, warnings, announcements, Format and structure of notices, Agenda of meetings: meaning, elements, and format, Minutes of meetings: purpose, structure, and recording decisions, Importance of professional documentation.	5	CO5
		Self-Learning Topics: Study of real meeting, agendas and minutes (Corporate samples).		
6.	**General Topic Presentation & Report Making**	Principles of effective presentations, Structure of presentation: introduction, body, conclusion, Audience analysis and communication objectives, Presentation techniques: voice modulation, pacing, transitions, Slide design principles: simplicity, bullet points, visual clarity	5	CO6
		Self-Learning Topics : Learning Different types of reports (Technical, survey, analytical)	5	

Sr. No.	**Documents required to be submitted for internal orals:**
1.	Reflection report on GD.
2.	Resume drafted in latest format.

3.	Corporate meeting – Frame Notice, Agenda, and minutes of the meeting.
4.	Short report format and a case study for the same.

References:

- Communication Skill for Engineers & Scientists (2nd Edition) by Sangeeta Sharma & Mohan N. Mishra.
- Art of Effective communication in group discussion and interview by O. P. Singh

Online References:

Sr	**Free Online Courses**
1	LPU Blog: Comprehensive GD guide with techniques, frameworks, phrases, do's/don'ts, and topic types (campus to corporate): https://www.lpu.in/blog/from-campus-to-corporate-the-ultimate-guide-to-master group-discussion/
2	Tech Interview Handbook: ATS-friendly resume guide for software engineers (templates, keywords, structure, tools): https://www.techinterviewhandbook.org/resume/

Term work marks – (Total 25 marks)

Group Discussion - 10 marks

Mock PI - 5 marks

Mock Meeting and Roleplay -5 marks

Attendance - 5 marks

Course Code	Course Name (MDM-III)	Teaching Scheme (Contact Hours Per Week			Teaching Scheme (Contact Hours Per Semester)					
		L	T	P	L	T	P	SL	Notional Learning Hour	Total Credits (C) (Notional Learning Hour/30
993116815	Decentralized Technology	3	--	-	45	-	-	45	90	3

(1 Credit = 30 Notional Learning Hours)

Course Code	Course Name	Theory					Term work	Pract / Oral	Total
		Internal Assessment			End Sem Exam	Exam Duration(in			
		IAT-I	IAT-II	Total					
993116815	Decentralized Technology	20	20	40	60	2.5	--	--	100

Rationale:

The rapid evolution of distributed computing, blockchain systems, decentralized finance (DeFi), Web3 technologies, and peer-to-peer networks has transformed the way digital systems operate. Decentralized Technology focuses on systems where control, data, and decision-making are distributed across multiple nodes instead of centralized authorities. The course introduces the principles of blockchain, consensus mechanisms, smart contracts, decentralized applications (DApps), distributed storage, cryptocurrencies, and Web3 ecosystems. It enables students to understand the architecture, security, scalability, and practical implementation of decentralized systems used in industries such as finance, supply chain, healthcare, and digital identity management.

Course Objectives:

1. To understand the fundamentals of decentralized systems and distributed networks.
2. To study blockchain architecture and consensus mechanisms.
3. To learn cryptocurrencies, wallets, and transaction management.
4. To understand smart contracts and decentralized application development.
5. To analyze decentralized storage, security, and scalability challenges.
6. To introduce emerging applications of Web3 and decentralized technologies.

Course Outcomes:

Students will be able to:

1. Explain the principles and architecture of decentralized systems.
2. Analyze block chain technologies and consensus protocols.
3. Implement cryptocurrency transactions and wallet mechanisms.
4. Develop and deploy basic smart contracts and decentralized applications.
5. Evaluate security, scalability, and privacy challenges in decentralized systems.
6. Explore modern applications of Web3, DeFi, NFTs, and decentralized governance.

Prerequisite:

Basic knowledge of Computer Networks, Operating Systems, Cryptography, Data Structures, and Database Management Systems.

DETAILED SYLLABUS:

Sr. No.	Name of Module	Detailed Content	Hours	CO Mapping
I	**Introduction to Decentralized Systems**	Centralized vs Decentralized Systems, Distributed Computing Basics, Peer-to-Peer Networks, Byzantine Generals Problem, Fundamentals of Blockchain Technology, Components of Blockchain, Types of Blockchains: Public, Private, Consortium, Hybrid	07	CO1
		Self-learning Topics: Evolution of decentralized systems, Real-world blockchain use cases, History of Bitcoin and Ethereum	07	
II	**Blockchain Architecture and Consensus Mechanisms**	Blockchain Data Structure, Blocks and Transactions, Merkle Trees, Consensus Algorithms: Proof of Work (PoW), Proof of Stake (PoS), Delegated PoS, Practical Byzantine Fault Tolerance (PBFT), Mining Process, Forks and Chain Management	08	CO2
		Self-learning Topics: Energy consumption in blockchain, Comparative study of consensus algorithms, Blockchain scalability trilemma	08	
III	**Cryptocurrencies and Wallets**	Bitcoin Architecture, Ethereum Overview, Cryptocurrency Transactions, Digital Wallets and Key Management, UTXO Model, Gas Mechanism in Ethereum, Cryptocurrency Exchanges, Token Standards (ERC-20, ERC-721), Stablecoins	07	CO3

		Self-learning Topics: Cryptocurrency regulations in India, Wallet security techniques, NFT ecosystems and marketplaces	07	
IV	**Smart Contracts and DApps**	Introduction to Smart Contracts, Solidity Programming Basics, Ethereum Virtual Machine (EVM), Smart Contract Deployment, Decentralized Applications (DApps), Web3.js Basics, Testing and Debugging Smart Contracts, Security Vulnerabilities in Smart Contracts	08	CO4
		Self-learning Topics: DAO concepts, Smart contract auditing tools, Real-world DApp architectures	08	
V	**Decentralized Storage and Security**	InterPlanetary File System (IPFS), Decentralized Storage Concepts, Blockchain Security Threats, 51% Attack, Sybil Attack, Double Spending, Privacy and Anonymity, Zero Knowledge Proofs, Secure Identity Management in Blockchain	07	CO5
		Self-learning Topics: Decentralized cloud storage platforms, Blockchain for cybersecurity, Privacy-preserving blockchain systems	07	
VI	**Emerging Trends and Applications**	Decentralized Finance (DeFi), Non-Fungible Tokens (NFTs), Web3 Ecosystem, Decentralized Autonomous Organizations (DAO), Blockchain in Supply Chain and Healthcare, Central Bank Digital Currency (CBDC), Quantum Computing Impact on Blockchain, Future of Decentralized Technologies	08	CO6
		Self-learning Topics: Metaverse and blockchain integration, Green blockchain initiatives, Cross-chain interoperability	08	

Text Books:

1. Imran Bashir, Mastering Blockchain, Packt Publishing.
2. Andreas M. Antonopoulos, Mastering Bitcoin, O'Reilly.
3. Andreas M. Antonopoulos and Gavin Wood, Mastering Ethereum, O'Reilly.
4. Melanie Swan, Blockchain: Blueprint for a New Economy, O'Reilly.

References:

1. Arshdeep Bahga and Vijay Madisetti, Blockchain Applications: A Hands-On Approach, VPT.
2. Narayan Prusty, Building Blockchain Projects, Packt Publishing.
3. Roger Wattenhofer, The Science of the Blockchain, CreateSpace.

4. Alex Leverington, Ethereum Programming, Packt Publishing.

Online References:

Sr	**Module**	**Free Online Courses**
1	Module I	https://nptel.ac.in/courses/106105184 https://www.ibm.com/topics/blockchain
2	Module II	https://nptel.ac.in/courses/106105220
3	Module III	https://www.coursera.org/learn/cryptocurrency https://ethereum.org/en/developers/docs/ https://www.coinbase.com/learn/crypto-basics
4	Module IV	https://soliditylang.org/ https://web3js.readthedocs.io/
5	Module V	https://consensys.net/knowledge-base/blockchain/ https://z.cash/technology/zksnarks/
6	Module VI	https://ethereum.org/en/web3/