COURSE STRUCTURE (R23) –B. TECH IT
(Applicable from the academic year 2023-24 and onwards)

B. Tech – I Year I Semester

S.No.	Category	Title	L	T	P	Credits
1	R23BS01	Linear Algebra & Calculus	3	0	0	3
2	R23BS03	Engineering Physics	3	0	0	3
3	R23HS01	Communicative English	2	0	0	2
4	R23ES01	Basic Civil & Mechanical Engineering	3	0	0	3
5	R23ES07	Introduction to Programming	3	0	0	3
6	R23HS01	Communicative English Lab	0	0	2	1
7	R23BS03	Engineering Physics Lab	0	0	2	1
8	R23ES02	Engineering Workshop	0	0	3	1.5
9	R23ES06	IT Workshop	0	0	2	1
10	R23ES07	Computer Programming Lab	0	0	3	1.5
11	R23MC01	Health and Wellness, Yoga and Sports	0	0	1	0.5
Total						**20.5**

B. Tech – I Year II Semester

S.N.	Category	Title	L/D	T	P	Credits
1.	R23BS02	Differential Equations and Vector calculus	3	0	0	3
2.	R23BS05	Chemistry	3	0	0	3
3.	R23ES03	Engineering Graphics	1	0	4	3
4.	R23ES04	Basic Electrical &Electronics Engineering	3	0	0	3
5.	R23PC04	Data Structures	3	0	0	3
6.	R23BS05	Chemistry Lab	0	0	2	1
7.	R23ES05	Electrical &Electronics Engineering workshop	0	0	3	1.5
8.	R23PC04	Data Structures Lab	0	0	3	1.5
9.	R23MC02	NSS/NCC/Scouts &Guides/Community Service	0	0	1	0.5
Total						**19.5**

B.Tech. II Year I Semester

S.No.	Category	Title	L	T	P	Credits
1	BS&H	Mathematical Foundations of Computer Science	3	0	0	3
2	BS&H	Universal Human Values– Understanding Harmony	2	1	0	3
3	Engineering Science	Digital Logic & Computer Organization	3	0	0	3
4	Professional Core	Software Engineering	3	0	0	3
5	Professional Core	Object Oriented Programming Through Java	3	0	0	3
6	Professional Core	CASE Tools Lab	0	0	3	1.5
7	Professional Core	Object Oriented Programming Through Java Lab	0	0	3	1.5
8	Skill Enhancement course	Python Programming	0	1	2	2
9	Audit Course	Environmental Science	2	0	0	-
Total			**16**	**2**	**8**	**20**

B.Tech. II Year II Semester

S.No.	**Category**	**Title**	**L**	**T**	**P**	**Credits**
1	Management Course- I	Managerial Economics and Financial Analysis	2	0	0	2
2	Engineering Science/ Basic Science	Probability & Statistics	3	0	0	3
3	Professional Core	Operating Systems	3	0	0	3
4	Professional Core	Database Management Systems	3	0	0	3
5	Professional Core	Design and Analysis of Algorithms	3	0	0	3
6	Professional Core	Operating Systems Lab	0	0	3	1.5
7	Professional Core	Database Management Systems Lab	0	0	3	1.5
8	Skill Enhancement Course	Django Framework	0	1	2	2
9	BS&H	Design Thinking & Innovation	1	0	2	2
Total			**15**	**1**	**10**	**21**
Mandatory Community Service Project Internship of 08 weeks duration during summer vacation						

B.Tech. – III Year I Semester

S.No.	Category	Title	L	T	P	Credits
1	Professional Core	Advanced Java	3	0	0	3
2	Professional Core	Computer Networks	3	0	0	3
3	Professional Core	Automata Theory & Compiler Design	3	0	0	3
4	Professional Elective-I	1. Object Oriented Analysis and Design 2. Cyber Security 3. Artificial Intelligence 4. Microprocessors & Microcontrollers 5. Data Warehousing & Data Mining 6. 12-week MOOC Swayam/ NPTEL course recommended by the BoS	3	0	0	3
5	Open Elective- I	**Open elective offered by other department**	3	0	0	3
6	Professional Core	Advanced Java Lab	0	0	3	1.5
7	Professional Core	Computer Networks Lab	0	0	3	1.5
8	Skill Enhancement course	Full Stack Development 1	0	1	2	2
9	Engineering Science	User Interface Design using Flutter / SWAYAM Plus - Android Application Development (with Flutter)	0	0	2	1
10	Evaluation of Community Service Internship		-	-	-	2
	Total		15	1	10	23
MC	Minor Course (Student may select from the same specialized minors pool)		3	0	0	3
MC	Minor Course through SWAYAM/NPTEL (minimum 12 week, 3 credit course)		3	0	0	3
HC	Honors Course (Student may select from the same honors pool)		3	0	0	3

B.Tech. III Year II Semester

S.No.	Category	Title	L	T	P	Credits
1	Professional Core	Cloud Computing	3	0	0	3
2	Professional Core	Cryptography & Network Security	3	0	0	3
3	Professional Core	Machine Learning	3	0	0	3
4	Professional Elective-II	1. Software Testing Methodologies 2. Augmented Reality & Virtual Reality 3. DevOps 4. Generative AI 5. 12-week MOOC Swayam/NPTEL course recommended by the BoS	3	0	0	3
5	Professional Elective-III	1. Software Project Management 2. Mobile Adhoc Networks 3. Natural Language Processing 4. Distributed Operating System 5. 12-week MOOC Swayam/NPTEL course recommended by the BoS	3	0	0	3
6	Open Elective – II	**Open elective offered by other department**	3	0	0	3
7	Professional Core	Cloud Computing Lab	0	0	3	1.5
8	Professional Core	Machine Learning Lab	0	0	3	1.5
9	Skill Enhancement course	Soft skills / / SWAYAM Plus - 21st Century Employability Skills	0	1	2	2
10	Audit Course	Technical Paper Writing & IPR	2	0	0	-
Total			**20**	**1**	**08**	**23**
Mandatory Industry Internship / Mini Project of 08 weeks duration during summer vacation						
MC	Minor Course (Student may select from the same specialized minors pool)		3	0	3	4.5
HC	Honors Course (Student may select from the same honors pool)		3	0	0	3

* Under Industry Internship interested students can pursue SWAYAM Plus courses viz., Hands-on Masterclass on Data Analytics OR Artificial Intelligence for Real-World Application

B.Tech. IV Year I Semester

S.No.	Category	Title	L	T	P	Credits
1	Professional Core	Internet of Things	3	0	0	3
2	Management Course-II	Human Resources & Project Management	2	0	0	2
3	Professional Elective-IV	1. Software Architecture & Design Pattern 2. Deep Learning 3. Computer Vision 4. Block chain Technology 5. 12-week MOOC Swayam/NPTEL course recommended by the BoS	3	0	0	3
4	Professional Elective-V	1. Agile methodologies 2. Big Data Analytics 3. Mobile Computing 4. Cyber Physical Systems 5. 12-week MOOC Swayam/NPTEL course recommended by the BoS	3	0	0	3
5	Open Elective-III	**Open elective offered by other department**	3	0	0	3
6	Open Elective-IV	**Open elective offered by other department**	3	0	0	3
7	Skill Enhancement Course	Mean Stack Technologies/ SWAYAM Plus - Certificate program in Prompt Engineering and ChatGPT	0	1	2	2
8	Audit Course	Constitution of India	2	0	0	-
9	Internship	Evaluation of Industry Internship / Mini Project	-	-	-	2
Total			**19**	**1**	**02**	**21**
MC	Minor Course (Student may select from the same specialized minors pool)		3	0	0	3
HC	Honors Course (Student may select from the same honors pool)		3	0	0	3

B.Tech. IV Year II Semester

S.No.	Category	Title	L	T	P	Credits
1	Internship & Project Work	Full semester Internship / Project Work	0	0	24	12

Note: ***Student need to do at least ONE MOOC/NPTEL Course (of 3 credits out of 160 credits) to meet the mandatory requirement (11th criteria, as per R23 Regulations); they are allowed to register one semester in advance***

Open Electives, offered to other department students:

Open Elective I: Principles of Operating Systems/ Computer Organization and Architecture

Open Elective II: Principles of Database Management Systems

Open Elective III: Object Oriented Programming Through Java
Open Elective IV: Principles of Software Engineering /Computer Networks

Minor Engineering

Note:

1. *To obtain Minor Engineering, student needs to obtain 18 credits by successfully completing any of the following courses in the concern stream.*
2. *During Minor/Honors Course selection, there should not be any overlapping with Regular/Major/OPEN Electives*

Minor in IT

1. Principles of Database Management Systems
2. Principles of Software Engineering
3. Advanced Data Structures & Algorithm Analysis
4. Principles of Operating Systems

Any of the following 12 Week 3 credit NPTEL MOOC Courses

5. Artificial Intelligence: Knowledge Representation and Reasoning
6. Computer Networks and Internet Protocol
7. Machine Learning and Deep Learning - Fundamentals and Applications
8. The Joy of Computing using Python
9. Discrete Mathematics for CS
10. Human Computer Interaction (In English)
11. Data Analytics with Python
12. Foundations of Cyber Physical Systems

B.Tech. – III Year I Semester

III Year I Semester	Advanced Java	L	T	P	C
		3	0	0	3

Course Objectives:

1. To enable students to understand the fundamentals of JDBC and implement basic to advanced database operations using Java.
2. To provide a comprehensive understanding of J2EE architecture and web application development using servlets and JSP.
3. To explore the lifecycle and components of servlets and their integration within web applications, including session management and filters.
4. To introduce students to JSP architecture, scripting elements, expression language, and JSTL for dynamic content generation.
5. To provide hands-on experience in using Java-based web frameworks such as Spring MVC for building scalable enterprise web applications with proper transaction management.

Unit I:

JDBC Programming: JDBC Architecture, Types of JDBC Drivers, Introduction to major JDBC Classes and Interface, Creating simple JDBC Application, Types of Statement (Statement Interface, PreparedStatement, CallableStatement), Exploring ResultSet Operations, Batch Updates in JDBC, Creating CRUD Application, Using Rowsets Objects, Managing Database Transaction.

Unit II:

J2EE and Web Development: J2EE Architecture Types, J2EE Containers, Types of Servers in J2EE Application, HTTP Protocols and API, Request Processing in Web Application, Web Application Structure, Web Containers and Web Architecture Models.

Unit III:

Servlet API and Overview: Servlet Introduction, Servlet Life Cycle(SLC), Types of Servlet, Servlet Configuration with Deployment Descriptor, Working with ServletContext and ServletConfig Object, Attributes in Servelt,, Response and Redirection using Request Dispacher and using sendRedirect Method, Filter API, Manipulating Responses using Filter API, Session Tracking: using Cookies, HTTPSession, Hidden Form Fields and URL Rewriting,Types of Servlet Event: ContextLevel and SessionLevel.

Unit IV:

Java Server Pages(JSP): Introduction to JSP , Comparison with Servlet, JSP Architecture, JSP: Life Cycle, Scripting Elements, Directives, Action Tags, Implicit Objects, Expression Language(EL), JSP Standard Tag Libraries(JSTL), Custom Tag, Session Management, Exception Handling, CRUD Application

Unit V:

Java Web Frameworks: Spring MVC Spring: Introduction, Architecture, Spring MVC Module, Life Cycle of Bean Factory, Explore: Constructor Injection, Dependency Injection, Inner Beans, Aliases in Bean, Bean Scopes, Spring Annotations, Spring AOP Module, Spring DAO, Database Transaction Management, CRUD Operation using DAO and Spring API.

Course Outcomes:

After completing this course, the students will be able to:

1. Develop JDBC-based applications using different types of statements and manage transactions efficiently.
2. Design and structure enterprise-level web applications using J2EE architecture, containers, and HTTP protocols.
3. Implement servlets to handle request processing, session tracking, and filtering mechanisms in dynamic web applications.
4. Create interactive and dynamic web pages using JSP, JSTL, EL, and manage server-side scripting effectively.

5. Build enterprise applications using the Spring framework, apply dependency injection, and perform CRUD operations with Spring DAO and transaction management.

Text Books:

1. Black Book "Java server programming" J2EE, 1st ed., Dream Tech Publishers, 2008.
2. Complete Reference J2EE, James Keogh, McGraw Hill publication
3. Professional Java Server Programming, Subrahmanyam Allamaraju, Cedric Buest, Wiley Publication
4. Spring in Action, 3rd edition , Craig walls, Manning Publication

Reference Books:

1. Core Java, Volume II: Advanced Features, Cay Horstmann, Gary Cornell Pearson Publication
2. JDBC™ API Tutorial and Reference, Third Edition, Maydene Fisher, Jon Ellis, Jonathan Bruce, Addison Wesley
3. Beginning JSP, JSF and Tomcat, Giulio Zambon, Apress

III Year I Semester	Computer Networks	L	T	P	C
		3	0	0	3

Course Objectives:

1. To provide insight about networks, topologies, and the key concepts.
2. To gain comprehensive knowledge about the layered communication architectures (OSI and TCP/IP) and its functionalities.
3. To understand the principles, key protocols, design issues, and significance of each layers in ISO and TCP/IP.
4. To know the basic concepts of network services and various network applications.

UNIT I: Introduction: Network Types, LAN, MAN, WAN, Network Topologies Reference models- The OSI Reference Model- the TCP/IP Reference Model - A Comparison of the OSI and TCP/IP Reference Models, OSI Vs TCP/IP.
Physical Layer –Introduction to Guided Media- Twisted-pair cable, Coaxial cable and Fiber optic cable and introduction about unguided media.

UNIT II: Data link layer: Design issues, **Framing**: fixed size framing, variable size framing, flow control, error control, error detection and correction codes, CRC, Checksum: idea, one's complement internet checksum, services provided to Network Layer, **Elementary Data Link Layer protocols**: simplex protocol, Simplex stop and wait, Simplex protocol for Noisy Channel.
Sliding window protocol: One bit, Go back N, Selective repeat-Stop and wait protocol, Data link layer in HDLC, Point to point protocol (PPP)

UNIT – III: Media Access Control: Random Access: ALOHA, Carrier sense multiple access (CSMA), CSMA with Collision Detection, CSMA with Collision Avoidance, **Controlled Access:** Reservation, Polling, Token Passing, **Channelization:** frequency division multiple Access(FDMA), time division multiple access(TDMA), code division multiple access(CDMA).
Wired LANs: Ethernet, Ethernet Protocol, Standard Ethernet, Fast Ethernet(100 Mbps), Gigabit Ethernet, 10 Gigabit Ethernet.

UNIT – IV: The Network Layer Design Issues – Store and Forward Packet Switching-Services Provided to the Transport layer- Implementation of Connectionless Service-Implementation of Connection Oriented Service- Comparison of Virtual Circuit and Datagram Networks,
Routing Algorithms-The Optimality principle-Shortest path, Flooding, Distance vector, Link state, Hierarchical, Congestion Control algorithms-General principles of congestion control, Congestion prevention polices, Approaches to Congestion Control-Traffic Aware Routing-Admission Control-Traffic Throttling-Load Shedding. Traffic Control Algorithm-Leaky bucket & Token bucket.
Internet Working: How networks differ- How networks can be connected- Tunnelling, internetwork routing-, Fragmentation, network layer in the internet – IP protocols-IP Version 4 protocol-IPV4 Header Format, IP addresses, Class full Addressing, CIDR, Subnets-IP Version 6-The main IPV6 header, Transition from IPV4 to IPV6, Comparison of IPV4 & IPV6.

UNIT –V: The Transport Layer: Transport layer protocols: Introduction-services- port number-User data gram protocol-User datagram-UDP services-UDP applications-Transmission control protocol: TCP services- TCP features- Segment- A TCP connection-windows in TCP- flow control-Error control, Congestion control in TCP.
Application Layer –- World Wide Web: HTTP, Electronic mail-Architecture- web based mail- email security- TELENET-local versus remote Logging-Domain Name System.
Course Outcomes (COs):

1. Explain the structure and functions of various types of computer networks, network topologies, and compare the OSI and TCP/IP reference models.

2. Analyze the functions of the data link layer including framing, error detection and correction, and implement protocols such as Stop-and-Wait, Go-Back-N, and Selective Repeat.
3. Evaluate multiple media access control techniques and Ethernet technologies, including CSMA/CD, FDMA, TDMA, and Gigabit Ethernet standards.
4. Apply routing and congestion control algorithms and distinguish between IPv4 and IPv6 addressing schemes and their roles in internetworking.
5. Describe the operations and services of the transport and application layers, including protocols like TCP, UDP, HTTP, DNS, and email systems.

Text Books:

1. Computer Networksm, Andrew S Tanenbaum, Fifth Edition. Pearson Education/PHI
2. Data Communications and Networks, Behrouz A. Forouzan, Fifth Edition TMH.

References Books:

1. Data Communications and Networks- Achut S Godbole, AtulKahate
2. Computer Networks, Mayank Dave, CENGAGE

III Year I Semester	Automata Theory & Compiler Design	L	T	P	C
		3	0	0	3

Course Objectives:

1. Introduce the notion of formal languages and grammars
2. Design of Grammars, FAs and PDAs
3. To become familiar with the underlying theory and methods used in compiler design
4. To Introduce the parsing techniques, code optimization techniques and generate code

UNIT – I: Regular Expressions, Languages and Finite Automata

Formal Languages and the Chomsky Hierarchy, Regular Expressions and Regular Languages, Algebraic Laws for Regular Expressions, Applications of Regular Expressions, Abstract model of Finite Automaton, Transition Tables and Transition Graphs, Deterministic Finite Automata (DFA), Nondeterministic Finite Automata (NFA), Converting NFA to DFA, Finite Automata with ϵ transitions (NFA-ϵ), Converting NFA-ϵ to NFA/DFA, Minimization of Finite Automata, Equivalence of FA and Regular Expressions.

UNIT-II: Context Free Grammars and Push Down Automata:

Context Free Grammars (CFG) and Context Free Languages (CFL), Design of CFGs, Leftmost and Rightmost Derivations, Parse Trees, Applications of CFGs, Ambiguity in Grammars and Languages, Push Down Automata (PDA), The Language of a PDA, Equivalence of PDAs and CFGs.

UNIT-III: Lexical Analysis and Top-Down Parsing

The structure of a compiler, Role of lexical analyzer, Input Buffering, Specification of tokens, Recognition of tokens, The Lexical Analyser Generator –LEX.
Introduction to Syntax Analysis, Eliminating ambiguity and left recursion from a CFG, Recursive Decent Parsing, LL(1) Grammars, Nonrecursive Predictive Parsing .

UNIT-IV: Bottom-Up Parsing and Syntax Directed Translation

Shift-Reduce Parsing, Simple LR parsing, Canonical LR(1) Parsing, LALR Parsing, Parser Generators
Syntax Directed Definitions, Evaluation Orders for SDDs, Syntax Directed Translation Schemes

UNIT-V: Intermediate Code Generation, Code Generation and Optimization:

Three address code, Types and Declarations, Translation of Expressions, Type Checking, Control Flow, Issues in the design of a Code Generator, The Target Language, A simple Code GeneratorBasic Blocks and Flow Graphs, Optimization of Basic Blocks, Peephole Optimization

Course Outcomes (COs):

1. Demonstrate understanding of regular languages, regular expressions, and the construction and minimization of finite automata for language recognition.
2. Construct context-free grammars (CFGs) and analyze their properties using derivations, parse trees, and pushdown automata.
3. Apply lexical analysis techniques using tools like LEX to recognize tokens and perform scanning in the compilation process.
4. Analyze and implement top-down and bottom-up parsing techniques, including LL(1), SLR, LALR, and LR(1) parsers, and apply syntax-directed translation schemes.
5. Generate intermediate code using three-address code, perform code generation, and apply basic block optimization techniques for performance enhancement.

Textbooks:

1. Introduction to Automata Theory, Languages and Computation, J.E.Hopcroft, R.Motwani and J.D.Ullman, 3rd Edition, Pearson, 2008.
2. Compilers Principles, Techniques and Tools, 2nd Edition, Alfred V.Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman, Pearson

Reference Books:

1. Introduction to Languages and The Theory of Computation, John C. Martin, McGraw Hill.
2. Theory of Computer Science-Automata, Languages and Computation, K.L.P.Mishra and N.Chandrasekaran, 3rd Edition, PHI, 2007
3. Compiler Construction‖, K.V.N. Sunitha, Pearson, 2013
4. Compiler Design, SandeepSaxena, Rajkumar Singh Rathore, S.Chand publication

III Year I Semester	Object Oriented Analysis and Design (Professional Elective-I)	L	T	P	C
		3	0	0	3

Course Objectives: The main objective is the students to

1. Become familiar with all phases of OOAD.
2. Master the main features of the UML.
3. Master the main concepts of Object Technologies and how to apply them at work and develop the ability to analyze and solve challenging problem in various domains.
4. Learn the Object design Principles and understand how to apply them towards Implementation.

UNIT I:

Introduction: The Structure of Complex systems, The Inherent Complexity of Software, Attributes of Complex System, Organized and Disorganized Complexity, Bringing Order to Chaos, Designing Complex Systems. **Case Study:** System Architecture: Satellite-Based Navigation

UNIT II:

Introduction to UML: Importance of modeling, principles of modeling, object oriented modeling, conceptual model of the UML, Architecture, and Software Development Life Cycle. **Basic Structural Modeling:** Classes, Relationships, common Mechanisms, and diagrams. **Case Study:** Control System: Traffic Management.

UNIT III:

Class & Object Diagrams: Terms, concepts, modeling techniques for Class & Object Diagrams. **Advanced Structural Modeling:** Advanced classes, advanced relationships, Interfaces, Types and Roles, Packages. **Case Study:** AI: Cryptanalysis.

UNIT IV:

Basic Behavioral Modeling-I: Interactions, Interaction diagrams Use cases, Use case Diagrams, Activity Diagrams. **Case Study:** Web Application: Vacation Tracking System

UNIT V:

Advanced Behavioral Modeling: Events and signals, state machines, processes and Threads, time and space, state chart diagrams. **Architectural Modeling:** Component, Deployment, Component diagrams and Deployment diagrams. **Case Study:** Weather Forecasting

Text Books:

1. Grady BOOCH, Robert A. Maksimchuk, Michael W. ENGLE, Bobbi J. Young, Jim Conallen, Kellia Houston , “Object- Oriented Analysis and Design with Applications”, 3rd edition, 2013, PEARSON.
2. Grady Booch, James Rumbaugh, Ivar Jacobson: The Unified Modeling Language User Guide, Pearson Education.

Course Outcomes (COs):

1. Understand the inherent complexity of software systems and apply principles for designing structured, organized complex systems.
2. Demonstrate knowledge of object-oriented modeling and the Unified Modeling Language (UML) to represent software systems effectively across the software development life cycle.
3. Develop and interpret class and object diagrams using appropriate modeling techniques to define system structure and object interactions.
4. Create and analyze behavioral models including use case, activity, and interaction diagrams to capture dynamic aspects of system functionality.
5. Design architectural models using component and deployment diagrams, and model advanced behavioral features such as state machines and concurrent processes.

Reference Books:

1. Meilir Page-Jones: Fundamentals of Object Oriented Design in UML, Pearson Education.
2. Pascal Roques: Modeling Software Systems Using UML2, WILEY- Dreamtech India Pvt. Ltd.
3. Atul Kahate: Object Oriented Analysis & Design, The McGraw-Hill Companies.
4. Appling UML and Patterns: An introduction to Object – Oriented Analysis and Design and Unified Process, Craig Larman, Pearson Education.

III Year I Semester	Cyber Security (Professional Elective-I)	L	T	P	C
		3	0	0	3

Course Objectives:
The aim of the course is to
1. identify security risks and take preventive steps
2. understand the forensics fundamentals
3. understand the evidence capturing process
4. understand the preservation of digital evidence

UNIT I: Introduction to Cybercrime: Introduction, Cybercrime: Definition and Origins of the Word, Cybercrime and Information Security, Cybercriminals, Classifications of Cybercrime, Cyber stalking, Cyber cafe and Cybercrimes, Botnets. Attack Vector, Proliferation of Mobile and Wireless Devices, Security Challenges Posed by Mobile Devices, Attack son Mobile/Cell Phones, Network and Computer Attacks.

UNIT II:
Tools and Methods : Proxy Servers and Anonymizers, Phishing, Password Cracking, Key loggers and Spywares, VirusandWorms,TrojanHorsesandBackdoors,Steganography,Sniffers,Spoofing, Session Hijacking Buffer over flow, DoS and DDoS Attacks, SQL Injection, Buffer Overflow, Attacks on Wireless Networks, Identity Theft (ID Theft), Foot Printing and Social Engineering, Port Scanning, Enumeration.

UNIT III:
Cyber Crime Investigation: Introduction, Investigation Tools, eDiscovery, Digital Evidence Collection, Evidence Preservation, E-Mail Investigation, E-Mail Tracking, IP Tracking, E-Mail Recovery, Hands on Case Studies. Encryption and Decryption Methods, Search and Seizure of Computers, Recovering Deleted Evidences, Password Cracking.

UNIT IV:
Computer Forensics and Investigations: Understanding Computer Forensics, Preparing for Computer Investigations. Current Computer Forensics Tools: Evaluating Computer Forensics Tools, Computer Forensics Software Tools, Computer Forensics Hardware Tools, Validating and Testing Forensics Software, Face, Iris and Fingerprint Recognition, Audio Video Analysis, Windows System Forensics, Linux System Forensics, Graphics and Network Forensics, E-mail Investigations, Cell Phone and Mobile Device Forensics.

UNIT V:
Cyber Crime Legal Perspectives: Introduction, Cybercrime and the Legal Landscape around the World, The Indian IT Act, Challenges to Indian Law and Cyber crime Scenario in India,ConsequencesofNotAddressingtheWeaknessinInformationTechnologyAct,Digital SignaturesandtheIndianITAct,AmendmentstotheIndianITAct,CybercrimeandPunishment,Cyberlaw,TechnologyandStudents:IndianScenario.

Course Outcomes (COs):
1. Understand and classify various forms of cybercrime, identify cybercriminal behavior, and analyze security threats related to mobile and wireless devices.
2. Explain and evaluate the tools and techniques used in cyberattacks such as phishing, spoofing, malware, session hijacking, SQL injection, and social engineering.
3. Apply methods for cybercrime investigation, including digital evidence collection, email and IP tracking, data recovery, and encryption techniques.
4. Demonstrate the use of modern computer forensics tools and techniques for analyzing evidence across platforms including mobile devices, networks, and operating systems.
5. Interpret and apply national and international cyber laws, especially the Indian IT Act and its amendments, to understand legal responses to cybercrime.

Text Books:

1. Sun it Belapure Nina Godbole "Cyber Security: Understanding Cyber Crimes, Computer Forensics and Legal Perspectives", WILEY, 2011.
2. Nelson Phillips and Enfinger Steuart, "Computer Forensics and Investigations", Cengage Learning, New Delhi, 2009.

Reference Books:

1. Michael T. Simpson, Kent Backman and James E. Corley, "Hands on Ethical Hacking and Network Defence", Cengage, 2019.
2. Computer Forensics, Computer Crime Investigation by John R. Vacca, Firewall Media, New Delhi.
3. Alfred Basta, Nadine Basta,Mary Brown and Ravinder Kumar "Cyber Security and Cyber Laws" , Cengage,2018.

E-Resources:

1. CERT-In Guidelines- http://www.cert-in.org.in/
2. https://www.coursera.org/learn/introduction-cybersecurity-cyber-attacks [Online Course]
3. https://computersecurity.stanford.edu/free-online-videos [Free Online Videos]
4. Nickolai Zeldovich. 6.858 Computer Systems Security. Fall 2014. Massachusetts Institute of Technology: MIT OpenCourseWare, https://ocw.mit.edu License:Creative CommonsBY-NC-SA.

III Year I Semester	Artificial Intelligence (Professional Elective-I)	L	T	P	C
		3	0	0	3

Pre-requisite:

1. Knowledge in Computer Programming.
2. A course on "Mathematical Foundations of Computer Science".
3. Background in linear algebra, data structures and algorithms, and probability.

Course Objectives:

1. The student should be made to study the concepts of Artificial Intelligence.
2. The student should be made to learn the methods of solving problems using Artificial Intelligence.
3. The student should be made to introduce the concepts of Expert Systems.
4. To understand the applications of AI, namely game playing, theorem proving, and machine learning.
5. To learn different knowledge representation techniques

UNIT - I

Introduction: AI problems, foundation of AI and history of AI intelligent agents: Agents and Environments, the concept of rationality, the nature of environments, structure of agents, problem solving agents, problem formulation.

UNIT - II

Searching- Searching for solutions, uniformed search strategies – Breadth first search, depth first Search. Search with partial information (Heuristic search) Hill climbing, A* ,AO* Algorithms, Problem reduction, Game Playing-Adversial search, Games, mini-max algorithm, optimal decisions in multiplayer games, Problem in Game playing, Alpha-Beta pruning, Evaluation functions.

UNIT - III

Representation of Knowledge: Knowledge representation issues, predicate logic- logic programming, semantic nets- frames and inheritance, constraint propagation, representing knowledge using rules, rules based deduction systems. Reasoning under uncertainty, review of probability, Bayes' probabilistic interferences and dempstershafer theory.

UNIT - IV

Logic concepts: First order logic. Inference in first order logic, propositional vs. first order inference, unification & lifts forward chaining, Backward chaining, Resolution, Learning from observation Inductive learning, Decision trees, Explanation based learning, Statistical Learning methods, Reinforcement Learning.

UNIT - V

Expert Systems: Architecture of expert systems, Roles of expert systems – Knowledge Acquisition Meta knowledge Heuristics. Typical expert systems – MYCIN, DART, XCON: Expert systems shells.

Course Outcomes (COs):

1. Understand the fundamental concepts, history, and structure of AI systems, including intelligent agents and problem-solving approaches.
2. Apply various search algorithms such as BFS, DFS, A*, and game-playing strategies like Minimax and Alpha-Beta pruning to solve AI problems.
3. Represent and reason with knowledge using predicate logic, semantic networks, frames, and rule-based systems under both certain and uncertain conditions.
4. Perform inference using first-order logic and implement learning techniques including decision trees, inductive learning, and reinforcement learning.
5. Analyze the architecture and components of expert systems and evaluate their applications using real-world examples like MYCIN, DART, and XCON.

Textbooks:

1. S. Russel and P. Norvig, "Artificial Intelligence – A Modern Approach", SecondEdition, Pearson Education.
2. Kevin Night and Elaine Rich, Nair B., "Artificial Intelligence (SIE)", Mc Graw Hill

Reference Books:

1. David Poole, Alan Mackworth, Randy Goebel,"Computational Intelligence: a logical approach", Oxford University Press.
2. G. Luger, "Artificial Intelligence: Structures and Strategies for complex problemsolving", Fourth Edition, Pearson Education.
3. J. Nilsson, "Artificial Intelligence: A new Synthesis", Elsevier Publishers.
4. Artificial Intelligence, SarojKaushik, CENGAGE Learning.

Online Learning Resources:

1. https://ai.google/
2. https://swayam.gov.in/nd1_noc19_me71/preview

III Year I Semester	Microprocessors & Microcontrollers (Professional Elective-I)	L	T	P	C
		3	0	0	3

Course Objectives:

1. To introduce fundamental architectural concepts of microprocessors and microcontrollers.
2. To impart knowledge on addressing modes and instruction set of 8086 and 8051
3. To introduce assembly language programming concepts
4. To explain memory and I/O interfacing with 8086 and 8051
5. To introduce16 bit and 32 bit microcontrollers.

UNIT I:

8086 Architecture: Main features, pin diagram/description, 8086 microprocessor family, internal architecture, bus interfacing unit, execution unit, interrupts and interrupt response, 8086 system timing, minimum mode and maximum mode configuration.

UNIT II:

8086 Programming: Program development steps, instructions, addressing modes, assembler directives, writing simple programs with an assembler, assembly language program development tools.

UNIT III:

8086 Interfacing: Semiconductor memories interfacing (RAM, ROM), Intel 8255 programmable peripheral interface, Interfacing switches and LEDS, Interfacing seven segment displays, software and hardware interrupt applications, Intel 8251 USART architecture and interfacing, Intel 8237a DMA controller, stepper motor, A/D and D/A converters, Need for 8259 programmable interrupt controllers.

UNIT IV:

Microcontroller, Architecture of 8051, Special Function Registers(SFRs), I/O Pins Ports and Circuits, Instruction set, Addressing modes, Assembly language programming.

UNIT V:

Interfacing Microcontroller, Programming 8051 Timers, Serial Port Programming, Interrupts Programming, LCD & Keyboard Interfacing, ADC, DAC & Sensor Interfacing, External Memory Interface, Stepper Motor and Waveform generation, Comparison of Microprocessor, Microcontroller, PIC and ARM processors

Course Outcomes (COs):

1. Describe the architecture, functioning, and modes of operation of the 8086 microprocessor and its system timing.
2. Develop assembly language programs using 8086 instruction set, addressing modes, and assembler directives with appropriate development tools.
3. Interface 8086 with peripheral devices such as memory, switches, LEDs, displays, and I/O controllers like 8255, 8251, 8237, and 8259.
4. Explain the internal architecture of the 8051 microcontroller and perform assembly language programming using its instruction set and addressing modes.
5. Interface 8051 with external devices (timers, serial ports, LCD, sensors, ADC/DAC, motors) and compare microprocessors and microcontrollers with PIC and ARM architectures.

Textbooks:

1. Microprocessors and Interfacing – Programming and Hardware by Douglas V Hall, SSSP Rao, Tata McGraw Hill Education Private Limited, 3rd Edition,1994.
2. K M Bhurchandi, A K Ray, Advanced Microprocessors and Peripherals, 3rdedition, McGraw Hill Education, 2017.
3. Raj Kamal, Microcontrollers: Architecture, Programming, Interfacing and System Design, 2nd edition, Pearson, 2012.

Reference Books:

1. Ramesh S Gaonkar, Microprocessor Architecture Programming and Applications with the 8085, 6th edition, Penram International Publishing, 2013.
2. Kenneth J. Ayala, The 8051 Microcontroller, 3rd edition, Cengage Learning, 2004.

III Year I Semester	Data Warehousing & Data Mining (Professional Elective-I)	L	T	P	C
		3	0	0	3

Pre-requisites: Data Structures, Algorithms, Probability & Statistics, Data Base Management Systems

Course Objectives: The main objective of the course is to

1. Introduce basic concepts and techniques of data warehousing and data mining
2. Examine the types of the data to be mined and apply pre-processing methods on raw data
3. Discover interesting patterns, analyze supervised and unsupervised models and estimate the accuracy of the algorithms.

UNIT–I: Data Warehousing and Online Analytical Processing: Basic concepts, Data Warehouse Modeling: Data Cube and OLAP, Data Warehouse Design and Usage, Data Warehouse Implementation, Cloud Data Warehouse, Data Mining and Patten Mining, Technologies, Applications, Major issues, Data Objects & Attribute Types, Basic Statistical Descriptions of Data, Data Visualization, Measuring Data Similarity and Dissimilarity. (Text Book- 1)

UNIT II: Data Preprocessing: An Overview, Data Cleaning, Data Integration, Data Reduction, Data Transformation and Data Discretization. (Text Book- 1)

UNIT–III: Classification: Basic Concepts, General Approach to solving a classification problem, Decision Tree Induction: Attribute Selection Measures, Tree Pruning, Scalability and Decision Tree Induction, Visual Mining for Decision Tree Induction, Bayesian Classification Methods: Bayes Theorem, Naïve Bayes Classification, Rule-Based Classification, Model Evaluation and Selection. (Text Book- 2)

UNIT–IV: Association Analysis: Problem Definition, Frequent Itemset Generation, Rule Generation: Confident Based Pruning, Rule Generation in Apriori Algorithm, Compact Representation of frequent item sets, FP-Growth Algorithm. (Text Book- 2)

UNIT–V: Cluster Analysis: Overview, Basics and Importance of Cluster Analysis, Clustering techniques, Different Types of Clusters; K-means: The Basic K-means Algorithm, K-means Additional Issues, Bi-secting K Means, Agglomerative Hierarchical Clustering: Basic Agglomerative Hierarchical Clustering Algorithm DBSCAN: Traditional Density Center-Based Approach, DBSCAN Algorithm, Strengths and Weaknesses. (Text Book- 2)

Course Outcomes (COs):

1. Understand the fundamental concepts of data warehousing, OLAP, and data mining, including data modeling, cloud data warehouses, and data visualization techniques.
2. Apply data preprocessing techniques such as data cleaning, integration, reduction, transformation, and discretization to prepare data for mining.
3. Analyze and implement classification methods, including decision trees, Bayesian classifiers, and rule-based classification, along with model evaluation strategies.
4. Perform association rule mining using algorithms like Apriori and FP-Growth, and understand methods for generating and pruning frequent itemsets.
5. Explore various clustering techniques such as K-means, hierarchical clustering, and DBSCAN, and evaluate their strengths and weaknesses in practical applications.

Text Books:

1. Data Mining concepts and Techniques, 3rd edition, Jiawei Han, Michel Kamber, Elsevier, 2011.
2. Introduction to Data Mining: Pang-Ning Tan & Michael Steinbach, Vipin Kumar, Pearson, 2012.

Reference Books:

1. Data Mining: VikramPudi and P. Radha Krishna, Oxford Publisher.
2. Data Mining Techniques, Arun K Pujari, 3rd edition, Universities Press,2013.

3. (NPTEL course by Prof.PabitraMitra) http://onlinecourses.nptel.ac.in/noc17_mg24/preview
4. http://www.saedsayad.com/data_mining_map.htm

III Year I Semester	Open elective offered by other department	L	T	P	C
		3	0	0	3

III Year I Semester	Advanced Java Lab	L	T	P	C
		0	0	3	1.5

Course Objectives: the main objectives of the course are

- To make use of Servlet and JSP API in the process of enterprise application deployment.
- Implement components such as JSTL
- Distinguish Application Server, Web Container, JDBC
- Design and Development of web application having collaboration of Servlets, JSPs, Spring

Lab should cover the following concepts:

- JDBC programming
- J2EE and Web development
- Servlets
- Java Server Pages
- Java Web Frameworks

Sample List of Experiments:

1. Write a JDBC application which will interact with Database and perform the following task.
    a. Create Student Table with RollNo, Name, and Address field and insert few records.
    b. Using Statement Object display the content of Record.
    c. Using Statement Object Insert Two Record.
    d. Using Statement Object Update One Record.
    e. Using Statement Object Delete One Record.
    f. Using Statement Object display the content of Record.

2. Write a JDBC application which will interact with Database and perform the following task.
    a. Create Student Table with Roll No, Name, and Address field and insert few records.
    b. Using Prepared Statement Object display the content of Record.
    c. Using Prepared Statement Object Insert Two Record.
    d. Using Prepared Statement Object Update One Record.
    e. Using Prepared Statement Object Delete One Record.
    f. Using Prepared Statement Object display the content of Record

3. Write a JDBC application which will interact with Database and perform the following task.
    a. Create a store procedure which will insert one record into employee table.
    b. Create a store procedure which will retrieve salary for given employee id.
    c. Write a java application which will call the above procedure and display appropriate information on screen

4. Design a JDBC application which will demonstrate Scrollable ResultSet functionality.

5. Design a JDBC application which will demonstrate Updatable ResultSet functionality.

6. Write down the Program for testing the Servlet and study deployment descriptor.

7. Write down the program for testing the include action for servlet collaboration.
8. Create login form and perform state management using Cookies, HttpSession and URL Rewriting.

9. Write down the Program which displays the simple JSP file
10. Write down the program in which input the two numbers in an html file and then display the addition in JSP file.
11. Perform Database Access through JSP.
12. Write down a program which demonstrates the core tag of JSTL.
13. Write down a program which demonstrates the Format tag of JSTL.
14. Write down a program which demonstrates the Function tag of JSTL.
15. Write down a program which demonstrates the SQL tag of JSTL.
16. Study and Implement MVC using Spring Framework
17. Using Spring Template manage Database and Transaction.

Course Outcomes (COs):

1. Develop JDBC applications to perform CRUD operations, manage database transactions, and utilize advanced JDBC features like stored procedures and scrollable/updatable Result Sets.
2. Design and implement Servlets and JSPs for dynamic web content generation, including handling user sessions and request dispatching.
3. Apply JSP Standard Tag Libraries (JSTL) and expression language for efficient web page development and database interaction.
4. Demonstrate state management techniques in web applications using Cookies, Http Session, and URL rewriting to maintain user sessions.
5. Build and deploy Java Web applications following the MVC architecture using Spring Framework, including transaction management and database integration with Spring Template.

III Year I Semester	Computer Networks Lab	L	T	P	C
		0	0	3	1.5

Course Objectives:
Learn basic concepts of computer networking and acquire practical notions of protocols with the emphasis on TCP/IP. A lab provides a practical approach to Ethernet/Internet networking: networks are assembled, and experiments are made to understand the layered architecture and how do some important protocols work

List of Experiments:
1. Study of Network devices in detail and connect the computers in Local Area Network.
2. Write a Program to implement the data link layer farming methods such as
   i) Character stuffing ii) bit stuffing.
3. Write a Program to implement data link layer farming method checksum.
4. Write a program for Hamming Code generation for error detection and correction.
5. Write a Program to implement on a data set of characters the three CRC polynomials – CRC 12, CRC 16 and CRC CCIP.
6. Write a Program to implement Sliding window protocol for Goback N.
7. Write a Program to implement Sliding window protocol for Selective repeat.
8. Write a Program to implement Stop and Wait Protocol.
9. Write a program for congestion control using leaky bucket algorithm
10. Write a Program to implement Dijkstra's algorithm to compute the Shortest path through a graph.
11. Write a Program to implement Distance vector routing algorithm by obtaining routing table at each node (Take an example subnet graph with weights indicating delay between nodes).
12. Write a Program to implement Broadcast tree by taking subnet of hosts.
13. Wire shark
    i. Packet Capture Using Wire shark
    ii. Starting Wire shark
    iii. Viewing Captured Traffic
    iv. Analysis and Statistics & Filters.
14. How to run Nmap scan
15. Operating System Detection using Nmap
16. Do the following using NS2 Simulator
    i. NS2 Simulator-Introduction
    ii. Simulate to Find the Number of Packets Dropped
    iii. Simulate to Find the Number of Packets Dropped by TCP/UDP
    iv. Simulate to Find the Number of Packets Dropped due to Congestion
v. Simulate to Compare Data Rate& Throughput.

Course Outcomes (COs):

1. Understand the operation of networking devices and configure basic LANs for effective communication.
2. Learn and apply data link layer techniques for framing, error detection, and correction.
3. Develop reliable data transmission protocols including Stop-and-Wait, Go-Back-N, and Selective Repeat.
4. Apply routing algorithms and congestion control methods to ensure efficient network performance.
5. Utilize network analysis tools for traffic monitoring, security evaluation, and protocol performance assessment.

III Year I Semester	Full Stack Development (Skill Enhancement Course)	L	T	P	C
		0	1	2	2

Course Objectives:
The main objectives of the course are to
- Make use of HTML elements and their attributes for designing static web pages
- Build a web page by applying appropriate CSS styles to HTML elements
- Experiment with JavaScript to develop dynamic web pages and validate forms

Experiments covering the Topics:
- Lists, Links and Images
- HTML Tables, Forms and Frames
- HTML 5 and Cascading Style Sheets, Types of CSS
- Selector forms
- CSS with Color, Background, Font, Text and CSS Box Model
- Applying JavaScript - internal and external, I/O, Type Conversion
- JavaScript Conditional Statements and Loops, Pre-defined and User-defined Objects
- JavaScript Functions and Events

Sample Experiments:

1. Lists, Links and Images
a. Write a HTML program, to explain the working of lists.
Note: It should have an ordered list, unordered list, nested lists and ordered list in an unordered list and definition lists.
b. Write a HTML program, to explain the working of hyperlinks using <a> tag and href, target Attributes.
c. Create a HTML document that has your image and your friend's image with a specific height and width. Also when clicked on the images it should navigate to their respective profiles.
d. Write a HTML program, in such a way that, rather than placing large images on a page, the preferred technique is to use thumbnails by setting the height and width parameters to something like to 100*100 pixels. Each thumbnail image is also a link to a full sized version of the image. Create an image gallery using this technique

2. HTML Tables, Forms and Frames
a. Write a HTML program, to explain the working of tables. (use tags: <table>, <tr>, <th>, <td> and attributes: border, rowspan, colspan)
b. Write a HTML program, to explain the working of tables by preparing a timetable. (Note: Use <caption> tag to set the caption to the table & also use cell spacing, cell padding, border, rowspan, colspan etc.).
c. Write a HTML program, to explain the working of forms by designing Registration form. (Note: Include text field, password field, number field, date of birth field, checkboxes, radio buttons, list boxes using <select>&<option> tags, <text area> and two buttons ie: submit and reset. Use tables to provide a better view).
d. Write a HTML program, to explain the working of frames, such that page is to be divided into 3 parts on either direction. (Note: first frame image, second frame paragraph, third frame hyperlink. And also make sure of using "no frame" attribute such that frames to be fixed).

3. HTML 5 and Cascading Style Sheets, Types of CSS
a. Write a HTML program, that makes use of <article>, <aside>, <figure>, <figcaption>, <footer>, <header>, <main>, <nav>, <section>, , tags.
b. Write a HTML program, to embed audio and video into HTML web page.
c. Write a program to apply different types (or levels of styles or style specification formats) - inline, internal, external styles to HTML elements. (identify selector, property and value).

4. Selector forms

a. Write a program to apply different types of selector forms
   i. Simple selector (element, id, class, group, universal)
   ii. Combinator selector (descendant, child, adjacent sibling, general sibling)
   iii. Pseudo-class selector
   iv. Pseudo-element selector
   v. Attribute selector

5. CSS with Color, Background, Font, Text and CSS Box Model

a. Write a program to demonstrate the various ways you can reference a color in CSS.
b. Write a CSS rule that places a background image halfway down the page, tilting it horizontally. The image should remain in place when the user scrolls up or down.
c. Write a program using the following terms related to CSS font and text:
   i. font-size ii. font-weight iii. font-style
   iv. text-decoration v. text-transformation vi. text-alignment
d. Write a program, to explain the importance of CSS Box model using
   i. Content ii. Border iii. Margin iv. padding

6. Applying JavaScript - internal and external, I/O, Type Conversion

a. Write a program to embed internal and external JavaScript in a web page.
b. Write a program to explain the different ways for displaying output.
c. Write a program to explain the different ways for taking input.
d. Create a webpage which uses prompt dialogue box to ask a voter for his name and age. Display the information in table format along with either the voter can vote or not

7. JavaScript Pre-defined and User-defined Objects

a. Write a program using document object properties and methods.
b. Write a program using window object properties and methods.
c. Write a program using array object properties and methods.
d. Write a program using math object properties and methods.
e. Write a program using string object properties and methods.
f. Write a program using regex object properties and methods.
g. Write a program using date object properties and methods.
h. Write a program to explain user-defined object by using properties, methods, accessors, constructors and display.

8. JavaScript Conditional Statements and Loops

a. Write a program which asks the user to enter three integers, obtains the numbers from the user and outputs HTML text that displays the larger number followed by the words "LARGER NUMBER" in an information message dialog. If the numbers are equal, output HTML text as "EQUAL NUMBERS".
b. Write a program to display week days using switch case.
c. Write a program to print 1 to 10 numbers using for, while and do-while loops.
d. Write aprogram to print data in object using for-in, for-each and for-of loops
e. Develop a program to determine whether a given number is an 'ARMSTRONG NUMBER' or not. [Eg: 153 is an Armstrong number, since sum of the cube of the digits is equal to the number i.e.,13 + 53+ 33 = 153]
f. Write a program to display the denomination of the amount deposited in the bank in terms of 100's, 50's, 20's, 10's, 5's, 2's & 1's. (Eg: If deposited amount is Rs.163, the output should be 1-100's, 1-50's, 1- 10's, 1-2's & 1-1's)

9. Javascript Functions and Events

a. Design a appropriate function should be called to display
   i. Factorial of that number
   ii. Fibonacci series up to that number
   iii. Prime numbers up to that number
   iv. Is it palindrome or not

b. Design a HTML having a text box and four buttons named Factorial, Fibonacci, Prime, and Palindrome. When a button is pressed an appropriate function should be called to display
   i. Factorial of that number
   ii. Fibonacci series up to that number
   iii. Prime numbers up to that number
   iv. Is it palindrome or not

c. Write a program to validate the following fields in a registration page
   i. Name (start with alphabet and followed by alphanumeric and the length should not be less than 6 characters)
   ii. Mobile (only numbers and length 10 digits)
   iii. E-mail (should contain format like xxxxxxx@xxxxxx.xxx)

Course Outcomes (COs):

1. Develop well-structured and semantically rich web pages using HTML5 elements, including lists, tables, forms, frames, and multimedia integration.
2. Apply CSS styling techniques effectively using various selector types and properties related to colors, backgrounds, fonts, text, and the CSS box model to create visually appealing web layouts.
3. Implement client-side scripting using JavaScript for dynamic webpage behavior, including input/output handling, type conversions, control flow statements, loops, and event-driven programming.
4. Utilize JavaScript built-in objects and user-defined objects to design interactive web applications with proper data manipulation and validation techniques.
5. Design and validate web forms using JavaScript functions and regular expressions to ensure data integrity and enhance user experience in web applications.

Text Books:

1. Programming the World Wide Web, 7th Edition, Robet W Sebesta, Pearson, 2013.
2. Web Programming with HTML5, CSS and JavaScript, John Dean, Jones & Bartlett Learning, 2019 (Chapters 1-11).
3. Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node, Vasan Subramanian, 2nd edition, APress, O'Reilly.

Web Links:

1. https://www.w3schools.com/html
2. https://www.w3schools.com/css

https://www.w3schools.com/js/

III Year I Semester	User Interface Design using Flutter	L	T	P	C
		0	0	2	1

Course Objectives:

- Learns to Implement Flutter Widgets and Layouts
- Understands Responsive UI Design and with Navigation in Flutter
- Knowledge on Widges and customize widgets for specific UI elements, Themes
- Understand to include animation apart from fetching data

List of Experiments:

Students need to implement the following experiments

1. a) Install Flutter and Dart SDK.
   b) Write a simple Dart program to understand the language basics.
2. a) Explore various Flutter widgets (Text, Image, Container, etc.).
   b) Implement different layout structures using Row, Column, and Stack widgets.
3. a) Design a responsive UI that adapts to different screen sizes.
   b) Implement media queries and breakpoints for responsiveness.
4. a) Set up navigation between different screens using Navigator.
   b) Implement navigation with named routes.
5. a) Learn about stateful and stateless widgets.
   b) Implement state management using set State and Provider.
6. a) Create custom widgets for specific UI elements.
   b) Apply styling using themes and custom styles.
7. a) Design a form with various input fields.
   b) Implement form validation and error handling.
8. a) Add animations to UI elements using Flutter's animation framework.
   b) Experiment with different types of animations (fade, slide, etc.).
9. a) Fetch data from a REST API.
   b) Display the fetched data in a meaningful way in the UI.
10. a) Write unit tests for UI components.
    b) Use Flutter's debugging tools to identify and fix issues.

Course Outcomes (COs):

1. Demonstrate proficiency in Dart programming and effectively use Flutter widgets and layouts to build functional user interfaces.
2. Design and implement responsive Flutter applications that adapt seamlessly across various screen sizes and devices using media queries and breakpoints.
3. Implement robust navigation and state management techniques, including named routes and Provider, to manage app flow and data state efficiently.
4. Develop customized widgets with appropriate styling and themes, enhancing the UI/UX of Flutter applications.
5. Integrate animations and REST API data fetching into Flutter apps, and apply testing and debugging practices to ensure quality and reliability.

Text Books:

1. Marco L. Napoli, Beginning Flutter: A Hands-on Guide to App Development.
2. Rap Payne, Beginning App Development with Flutter: Create Cross-Platform Mobile Apps 1st Edition, Apres
3. Richard Rose, Flutter & Dart Cookbook, Developing Full stack Applications for the Cloud, Oreilly.

III Year II Semester	Cloud Computing	L	T	P	C
		3	0	0	3

Course Objectives:

1. To explain the evolving utility computing model called cloud computing.
2. To introduce the various levels of services offered by cloud.
3. To discuss the fundamentals of cloud enabling technologies such as distributed computing, service-oriented architecture and virtualization.
4. To emphasize the security and other challenges in cloud computing.
5. To introduce the advanced concepts such as containers, serverless computing and cloud-centric Internet of Things.

UNIT -I: Introduction to Cloud Computing Fundamentals
Cloud computing at a glance, defining a cloud, cloud computing reference model, types of services (IaaS, PaaS, SaaS), cloud deployment models (public, private, hybrid), utility computing, cloud computing characteristics and benefits, cloud service providers (Amazon Web Services, Microsoft Azure, Google AppEngine).

UNIT-II: Cloud Enabling Technologies
Ubiquitous Internet, parallel and distributed computing, elements of parallel computing, hardware architectures for parallel computing (SISD, SIMD, MISD, MIMD), elements of distributed computing, Inter-process communication, technologies for distributed computing, remote procedure calls (RPC), service-oriented architecture (SOA), Web services, virtualization.

UNIT-III: Virtualization and Containers
Characteristics of virtualized environments, taxonomy of virtualization techniques, virtualization and cloud Computing, pros and cons of virtualization, technology examples (XEN, VMware), building blocks of containers, container platforms (LXC, Docker), container orchestration, Docker Swarm and Kubernetes, public cloud VM (e.g. Amazon EC2) and container (e.g. Amazon Elastic Container Service) offerings.

UNIT-IV: Cloud computing challenges
Economics of the cloud, cloud interoperability and standards, scalability and fault tolerance, energy efficiency in clouds, federated clouds, cloud computing security, fundamentals of computer security, cloud security architecture, cloud shared responsibility model, security in cloud deployment models.

UNIT -V: Advanced concepts in cloud computing
Serverless computing, Function-as-a-Service, serverless computing architecture, public cloud (e.g. AWS Lambda) and open-source (e.g. OpenFaaS) serverless platforms, Internet of Things (IoT), applications, cloud-centric IoT and layers, edge and fog computing, DevOps, infrastructure-as-code, quantum cloud computing.

Course Outcomes

1. Understand the fundamentals of cloud computing, including service models (IaaS, PaaS, SaaS), deployment models, and key benefits offered by cloud providers.
2. Explain enabling technologies behind cloud computing such as distributed and parallel computing, virtualization, service-oriented architecture, and web services.
3. Analyze virtualization techniques and container technologies (e.g., Docker, Kubernetes) and their role in cloud infrastructure and deployment.
4. Identify and evaluate challenges in cloud computing related to economics, interoperability, scalability, fault tolerance, and security, including cloud security architecture and models.
5. Explore advanced cloud concepts such as serverless computing, IoT integration with cloud, edge/fog computing, DevOps practices, and emerging technologies like quantum cloud computing.

Text Books:

1. Mastering Cloud Computing, 2nd edition, Rajkumar Buyya, Christian Vecchiola, Thamarai Selvi, Shivananda Poojara, Satish N. Srirama, Mc Graw Hill, 2024.
2. Distributed and Cloud Computing, Kai Hwang, Geoffery C. Fox, Jack J. Dongarra, Elsevier, 2012.

Reference Books:

1. Cloud Computing, Theory and Practice, Dan C Marinescu, 2nd edition, MK Elsevier, 2018.
2. Essentials of cloud Computing, K. Chandrasekhran, CRC press, 2014.
3. Online documentation and tutorials from cloud service providers (e.g., AWS, Azure, GCP)

III Year II Semester	Cryptography & Network Security	L	T	P	C
		3	0	0	3

Course Objectives:
The main objectives of this course are to explore the working principles and utilities of various cryptographic algorithms including secret key cryptography, hashes and message digests, public key algorithms, design issues and working principles of various authentication protocols and various secure communication standards including Kerberos, IPsec, and SSL/TLS.

UNIT I:
Basic Principles : Security Goals, Cryptographic Attacks, Services and Mechanisms, Mathematics of Cryptography- integer arithmetic, modular arithmetic, matrices, linear conguence.

UNIT II:
Symmetric Encryption: Mathematics of Symmetric Key Cryptography-algebraic structures, GF(2^n) Fields, Introduction to Modern Symmetric Key Ciphers-modern block ciphers, modern stream ciphers, Data Encryption Standard- DES structure, DES analysis, Security of DES, Multiple DES, Advanced Encryption Standard-transformations, key expansions, AES ciphers, Analysis of AES.

UNIT III:
Asymmetric Encryption: Mathematics of Asymmetric Key Cryptography-primes, primality testing, factorization, CRT, Asymmetric Key Cryptography- RSA crypto system, Rabin cryptosystem, Elgamal Crypto system, ECC

UNIT IV:
Data Integrity, Digital Signature Schemes & Key Management : Message Integrity and Message Authentication-message integrity, Random Oracle model, Message authentication, Cryptographic Hash Functions-whirlpool, SHA-512, Digital Signature- process, services, attacks, schemes, applications, Key Management-symmetric key distribution, Kerberos.

UNIT V:
Network Security-I: Security at application layer: PGP and S/MIME, Security at the Transport Layer: SSL and TLS, **Network Security-II :** Security at the Network Layer: IPSec-two modes, two security protocols, security association, IKE, ISAKMP, System Security-users, trust, trusted systems, buffer overflow, malicious software, worms, viruses, IDS, Firewalls.

Course Outcomes

1. Understand the fundamental principles of cryptography, including security goals, cryptographic attacks, and essential mathematical foundations such as modular arithmetic and integer arithmetic.
2. Explain symmetric key cryptography concepts and algorithms, including DES, multiple DES, and AES, along with their mathematical foundations and security analyses.
3. Analyze asymmetric key cryptography methods, including RSA, Rabin, ElGamal, and Elliptic Curve Cryptography (ECC), supported by relevant number theory concepts like primality testing and factorization.
4. Understand data integrity, digital signatures, cryptographic hash functions, and key management techniques including Kerberos for secure communication.
5. Evaluate network security mechanisms across different layers, including application layer security (PGP, S/MIME), transport layer security (SSL/TLS), network layer security (IPSec), and system security topics such as firewalls, IDS, and malware defense.

Text Books:

1. Cryptography and Network Security, 3rd Edition Behrouz A Forouzan, Deb deep Mukhopadhyay, McGraw Hill,2015
2. Cryptography and Network Security,4th Edition, William Stallings, (6e) Pearson,2006
3. Everyday Cryptography, 1st Edition, Keith M.Martin, Oxford,2016

Reference Books:

1. Network Security and Cryptography, 1st Edition, Bernard Meneges, Cengage Learning,2018

III Year II Semester	Machine Learning	L	T	P	C
		3	0	0	3

Course Objectives:
The objectives of the course is to

- Define machine learning and its different types (supervised and unsupervised) and understand their applications.
- Apply supervised learning algorithms including decision trees and k-nearest neighbours (k-NN).
- Implement unsupervised learning techniques, such as K-means clustering.

UNIT-I:
Introduction to Machine Learning: Evolution of Machine Learning, Paradigms for ML, Learning by Rote, Learning by Induction, Reinforcement Learning, Types of Data, Matching, Stages in Machine Learning, Data Acquisition, Feature Engineering, Data Representation, Model Selection, Model Learning, Model Evaluation, Model Prediction, Search and Learning, Data Sets.

UNIT-II:
Nearest Neighbor-Based Models: Introduction to Proximity Measures, Distance Measures, Non-Metric Similarity Functions, Proximity Between Binary Patterns, Different Classification Algorithms Based on the Distance Measures ,K-Nearest Neighbor Classifier, Radius Distance Nearest Neighbor Algorithm, KNN Regression, Performance of Classifiers, Performance of Regression Algorithms.

UNIT-III:
Models Based on Decision Trees: Decision Trees for Classification, Impurity Measures, Properties, Regression Based on Decision Trees, Bias–Variance Trade-off, Random Forests for Classification and Regression. The Bayes Classifier: Introduction to the Bayes Classifier, Bayes' Rule and Inference, The Bayes Classifier and its Optimality, Multi-Class Classification, Class Conditional Independence and Naive Bayes Classifier (NBC)

UNIT-IV:
Linear Discriminants for Machine Learning: Introduction to Linear Discriminants, Linear Discriminants for Classification, Perceptron Classifier, Perceptron Learning Algorithm, Support Vector Machines, Linearly Non-Separable Case, Non-linear SVM, Kernel Trick, Logistic Regression, Linear Regression, Multi-Layer Perceptrons (MLPs), Backpropagation for Training an MLP.

UNIT-V:
Clustering : Introduction to Clustering, Partitioning of Data, Matrix Factorization, Clustering of Patterns, Divisive Clustering, Agglomerative Clustering, Partitional Clustering, K-Means Clustering, Soft Partitioning, Soft Clustering, Fuzzy C-Means Clustering, Rough Clustering, Rough K-Means Clustering Algorithm, Expectation Maximization-Based Clustering, Spectral Clustering.

Course Outcomes

1. Understand the fundamental concepts and evolution of machine learning, including various learning paradigms, data types, and key stages involved in building machine learning models.
2. Apply nearest neighbor-based models and proximity measures for classification and regression problems, and evaluate the performance of these algorithms effectively.
3. Analyze decision tree-based models, including impurity measures, bias-variance trade-off, and ensemble methods such as Random Forests, as well as implement and understand Bayes classifiers and Naive Bayes algorithms.
4. Develop proficiency in linear discriminant methods including Perceptron, Support Vector Machines (SVM), Logistic Regression, and Multi-Layer Perceptrons (MLPs) with training techniques like backpropagation.

5. Explore clustering techniques such as K-Means, fuzzy clustering, hierarchical clustering, and spectral clustering to effectively partition and analyze unlabeled data.

Text Books:

1. "Machine Learning Theory and Practice", M N Murthy, V S Ananthanarayana, Universities Press (India), 2024

Reference Books:

1. "Machine Learning", Tom M. Mitchell, McGraw-Hill Publication, 2017
2. "Machine Learning in Action",Peter Harrington, DreamTech
3. "Introduction to Data Mining", Pang-Ning Tan, Michel Stenbach, Vipin Kumar, 7th Edition, 2019.

III Year II Semester	Software Testing Methodologies (Professional Elective-II)	L	T	P	C
		3	0	0	3

Course Objectives

- To provide knowledge of the concepts in software testing such as testing process, criteria, strategies, and methodologies.
- To develop skills in software test automation and management using the latest tools.

UNIT - I

Introduction: Purpose of testing, Dichotomies, model for testing, consequences of bugs, taxonomy of bugs Flow graphs and Path testing: Basics concepts of path testing, predicates, path predicates and achievable paths, path sensitizing, path instrumentation, application of path testing.

UNIT - II

Transaction Flow Testing: transaction flows, transaction flow testing techniques.
Data Flow testing: Basics of data flow testing, strategies in data flow testing, application of data flow testing.
Domain Testing: domains and paths, Nice & ugly domains, domain testing, domains and interfaces testing, domain and interface testing, domains and testability.

UNIT - III

Paths, Path products and Regular expressions: path products & path expression, reduction procedure, applications, regular expressions & flow anomaly detection.
Logic Based Testing: overview, decision tables, path expressions, kv charts, specifications.

UNIT - IV

State, State Graphs and Transition testing: state graphs, good & bad state graphs, state testing, Testability tips.

UNIT - V

Graph Matrices and Application: Motivational overview, matrix of graph, relations, power of a matrix, node reduction algorithm, building tools. (Student should be given an exposure to a tool like Jmeter/selenium/soapUI/Catalon).

Course Outcomes

1. Understand the fundamentals and purpose of software testing, including bug taxonomy, testing models, and techniques such as flow graphs and path testing for effective test case design.
2. Apply transaction flow, data flow, and domain testing methodologies to identify potential faults and improve software quality through systematic test coverage.
3. Analyze paths, path products, and regular expressions to detect flow anomalies and use logic-based testing techniques such as decision tables and KV charts for comprehensive test specification.
4. Develop skills in state and transition testing by modeling software behavior with state graphs and applying testability principles to validate dynamic system states.
5. Utilize graph matrices and associated algorithms for test case generation and gain practical experience with popular testing tools like JMeter, Selenium, SoapUI, or Katalon for automated test

Text Books:

1. Software Testing techniques - Baris Beizer, Dreamtech, second edition.
2. Software Testing Tools – Dr. K. V. K. K. Prasad, Dreamtech.

Reference Books:

1. The craft of software testing - Brian Marick, Pearson Education.
2. Software Testing Techniques – SPD(Oreille)
3. Software Testing in the Real World – Edward Kit, Pearson.
4. Effective methods of Software Testing, Perry, John Wiley.
5. Art of Software Testing – Meyers, John Wiley.

III Year II Semester	Augmented Reality & Virtual Reality (Professional Elective-II)	L	T	P	C
		3	0	0	3

Course Objectives:

- Provide a foundation to the fast growing field of AR and make the students aware of the various AR concepts.
- To give historical and modern overviews and perspectives on virtual reality. It describes the fundamentals of sensation, perception, technical and engineering aspects of virtual reality systems.

UNIT - I

Introduction to Augmented Reality: Augmented Reality - Defining augmented reality, history of augmented reality, Examples, Related fields

Displays: Multimodal Displays, Visual Perception, Requirements and Characteristics, Spatial Display Model, Visual Displays

Tracking: Tracking, Calibration, and Registration, Coordinate Systems, Characteristics of Tracking Technology, Stationary Tracking Systems, Mobile Sensors

UNIT - II

Computer Vision for Augmented Reality: Marker Tracking, Multiple-Camera Infrared Tracking, Natural Feature Tracking by Detection, Outdoor Tracking.

Interaction: Output Modalities, Input Modalities, Tangible Interfaces, Virtual User Interfaces on Real Surfaces, Augmented Paper, Multi-view Interfaces, Haptic Interaction

Software Architectures: AR Application Requirements, Software Engineering Requirements, Distributed Object Systems, Dataflow, Scene Graphs

UNIT - III

Introduction to Virtual Reality: Defining Virtual Reality, History of VR, Human Physiology and Perception

The Geometry of Virtual Worlds: Geometric Models, Axis-Angle Representations of Rotation, Viewing Transformations

Light and Optics: Basic Behavior of Light, Lenses, Optical Aberrations, The Human Eye, Cameras, Displays

UNIT - IV

The Physiology of Human Vision: From the Cornea to Photoreceptors, From Photoreceptors to the Visual Cortex, Eye Movements, Implications for VR

Visual Perception: Visual Perception - Perception of Depth, Perception of Motion, **Perception of Color Visual Rendering:** Visual Rendering -Ray Tracing and Shading Models, Rasterization, Correcting Optical Distortions, Improving Latency and Frame Rates, Immersive Photos and Videos

UNIT - V

Motion in Real and Virtual Worlds: Velocities and Accelerations, The Vestibular System, Physics in the Virtual World, Mismatched Motion and Vection

Interaction: Motor Programs and Remapping, Locomotion, Social Interaction

Audio: The Physics of Sound, The Physiology of Human Hearing, Auditory Perception, Auditory Rendering

Course Outcomes:

1. Understand the fundamental concepts, evolution, and applications of AR and VR.

2. Learn AR displays, tracking systems, calibration methods, and software architecture.

3. Apply computer vision techniques and design effective AR input/output interfaces.

4. Analyze visual perception, rendering, and latency factors in immersive VR environments.

5. Explore motion dynamics, vection, and auditory design for interactive VR experiences.

Text Books:

1. Augmented Reality: Principles & Practice by Schmalstieg / Hollerer, Pearson Education India;First edition (12 October 2016),ISBN-10: 9332578494
2. Virtual Reality, Steven M. LaValle, Cambridge University Press, 2016

Reference Books:

1. Allan Fowler-AR Game Developmentl, 1st Edition, A press Publications, 2018, ISBN 978-1484236178
2. Understanding Virtual Reality: Interface, Application and Design, William R Sherman and Alan B Craig, (The Morgan Kaufmann Series in Computer Graphics)". Morgan Kaufmann Publishers, San Francisco, CA, 2002
3. Developing Virtual Reality Applications: Foundations of Effective Design, Alan B Craig, William R Sherman and Jeffrey D Will, Morgan Kaufmann, 2009
4. Designing for Mixed Reality, Kharis O'Connell Published by O'Reilly Media, Inc., 2016, ISBN:9781491962381
5. Sanni Siltanen- Theory and applications of marker-based augmented reality. Julkaisija – Utgivare Publisher. 2012. ISBN 978-951-38-7449-0
6. Gerard Jounghyun Kim, "Designing Virtual Systems: The Structured Approach", 2005

III Year II Semester	DevOps (Professional Elective-II)	L	T	P	C
		3	0	0	3

Course Objectives:
The main objectives of this course are to:

- Describe the agile relationship between development and IT operations.
- Understand the skill sets and high-functioning teams involved in DevOps and related methods to reach a continuous delivery capability.
- Implement automated system update and DevOps lifecycle.

UNIT-I
Introduction to DevOps: Introduction to SDLC, Agile Model. Introduction to Devops. DevOps Features, DevOps Architecture, DevOps Lifecycle, Understanding Workflow and principles, Introduction to DevOps tools, Build Automation, Delivery Automation, Understanding Code Quality, Automation of CI/ CD. Release management, Scrum, Kanban, delivery pipeline, bottlenecks, examples

UNIT-II
Source Code Management (GIT): The need for source code control, The history of source code management, Roles and code, source code management system and migrations. What is Version Control and GIT, GIT Installation, GIT features, GIT workflow, working with remote repository, GIT commands, GIT branching, GIT staging and collaboration. UNIT TESTING - CODE COVERAGE: Junit, nUnit & Code Coverage with Sonar Qube, SonarQube - Code Quality Analysis.

UNIT-III
Build Automation - Continuous Integration (CI): Build Automation, What is CI Why Cl is Required, CI tools, Introduction to Jenkins (With Architecture), jenkins workflow, jenkins master slave architecture, Jenkins Pipelines, PIPELINE BASICS - Jenkins Master, Node, Agent, and Executor Freestyle Projects & Pipelines, Jenkins for Continuous Integration, Create and Manage Builds, User Management in Jenkins Schedule Builds, Launch Builds on Slave Nodes.

UNIT-IV
Continuous Delivery (CD): Importance of Continuous Delivery, CONTINUOUS DEPLOYMENT CD Flow, Containerization with Docker: Introduction to Docker, Docker installation, Docker commands, Images & Containers, DockerFile, Running containers, Working with containers and publish to Docker Hub.
Testing Tools: Introduction to Selenium and its features, JavaScript testing.

UNIT-V
Configuration Management - ANSIBLE: Introduction to Ansible, Ansible tasks, Roles, Jinja templating, Vaults, Deployments using Ansible.
CONTAINERIZATION USING KUBERNETES(OPENSHIFT): Introduction to Kubernetes Namespace & Resources, CI/CD - On OCP, BC, DC & ConfigMaps, Deploying Apps on Openshift Container Pods. Introduction to Puppet master and Chef.
Course Outcomes:
1. Understand DevOps principles, lifecycle, architecture, and the role of agile methodologies in CI/CD workflows.
2. Apply version control and source code management using Git along with code quality tools like JUnit and SonarQube.
3. Demonstrate build automation and continuous integration using Jenkins pipelines and distributed build systems.
4. Learn continuous delivery processes and containerization using Docker with automated testing through Selenium.
5. Implement configuration management with Ansible and container orchestration using Kubernetes, OpenShift, Puppet, and Chef.

Text Books:

1. Joyner, Joseph., Devops for Beginners: Devops Software Development Method Guide for Software Developers and It Professionals, 1st Edition Mihails Konoplows, 2015.
2. Alisson Machado de Menezes., Hands-on DevOps with Linux,1st Edition, BPB Publications, India, 2021.

Reference Books:

1. Len Bass, Ingo Weber, Liming Zhu. DevOps: A Software Architect's Perspective. Addison Wesley; ISBN-10
2. Gene Kim Je Humble, Patrick Debois, John Willis. The DevOps Handbook, 1st Edition, IT Revolution Press, 2016.
3. Verona, Joakim Practical DevOps, 1st Edition, Packt Publishing, 2016.
4. Joakim Verona. Practical Devops, Ingram short title; 2nd edition (2018). ISBN10: 1788392574
5. Deepak Gaikwad, Viral Thakkar. DevOps Tools from Practitioner's Viewpoint. Wiley publications. ISBN: 9788126579952

III Year II Semester	Generative AI (Professional Elective-II)	L	T	P	C
		3	0	0	3

Course Objectives:
- Understand the basics of Generative AI.
- Know the basics of Text Generation.
- Understand the process of generating videos.
- Know about GAN and its variants.

UNIT I :
Introduction To Gen Ai: Historical Overview of Generative modelling, Difference between Gen AI and Discriminative Modeling, Importance of generative models in AI and Machine Learning, Types of Generative models, GANs, VAEs, autoregressive models and Vector quantized Diffusion models, Understanding of probabilistic modeling and generative process, Challenges of Generative Modeling, Future of Gen AI, Ethical Aspects of AI, Responsible AI, Use Cases.

UNIT II:
Generative Models For Text: Language Models Basics, Building blocks of Language models, Transformer Architecture, Encoder and Decoder, Attention mechanisms, Generation of Text, Models like BERT and GPT models, Generation of Text, Autoencoding, Regression Models, Exploring ChatGPT, Prompt Engineering: Designing Prompts, Revising Prompts using Reinforcement Learning from Human Feedback (RLHF), Retrieval Augmented Generation, Multimodal LLM, Issues of LLM like hallucination.

UNIT III:
Generation of Images: Introduction to Generative Adversarial Networks, Adversarial Training Process, Nash Equilibrium, Variational Autoencoders, Encoder-Decoder Architectures, Stable Diffusion Models, Introduction to Transformer-based Image Generation, CLIP, Visual Transformers ViT- Dall-E2 and Dall-E3, GPT-4V, Issues of Image Generation models like Mode Collapse and Stability.

UNIT IV:
Generation of Painting, Music, and Play: Variants of GAN, Types of GAN, Cyclic GAN, Using Cyclic GAN to Generate Paintings, Neural Style Transfer, Style Transfer, Music Generating RNN, MuseGAN, Autonomous agents, Deep Q Algorithm, Actor-critic Network.

UNIT V:
Open Source Models And Programming Frameworks: Training and Fine tuning of Generative models, GPT 4 All, Transfer learning and Pretrained models, Training vision models, Google Copilot, Programming LLM, LangChain, Open Source Models, Llama, Programming for TimeSformer, Deployment, Hugging Face.

Course Outcomes
1. Understand generative models like GANs, VAEs, and autoregressive models with ethical considerations.
2. Apply transformer-based models such as BERT and GPT for text generation using prompt engineering and RLHF.
3. Explore image generation techniques using GANs, VAEs, and transformer models like DALL-E.
4. Create art and music with GANs, neural style transfer, and RNN-based models like MuseGAN.
5. Train, fine-tune, and deploy generative models using tools like Hugging Face and Google Copilot.

Text Books:
1. Denis Rothman, “Transformers for Natural Language Processing and Computer Vision”, Third Edition , Packt Books, 2024

Reference Books:

1. David Foster, ”Generative Deep Learning”, O’Reily Books, 2024.
2. Altaf Rehmani, “Generative AI for Everyone”, BlueRose One, 2024.

III Year II Semester	Software Project Management (Professional Elective-III)	L	T	P	C
		3	0	0	3

Course Objectives:
At the end of the course, the student shall be able to:

- To describe and determine the purpose and importance of project management from the perspectives of planning, tracking and completion of project
- To compare and differentiate organization structures and project structures
- To implement a project to manage project schedule, expenses and resources with the application of suitable project management tools

UNIT-I:
Conventional Software Management: The waterfall model, conventional software Management performance.
Evolution of Software Economics: Software Economics, pragmatic software cost estimation.
Improving Software Economics: Reducing Software product size, improving software processes, improving team effectiveness, improving automation, Achieving required quality, peer inspections.
The old way and the new: The principles of conventional software Engineering, principles of modern software management, transitioning to an iterative process.

UNIT-II:
Life cycle phases: Engineering and production stages, inception, Elaboration, construction, transition phases.
Artifacts of the process: The artifact sets, Management artifacts, Engineering artifacts, programmatic artifacts.

UNIT- III:
Model based software architectures: A Management perspective and technical perspective.
Work Flows of the process: Software process workflows, Iteration workflows.
Checkpoints of the process: Major mile stones, Minor Milestones, Periodic status assessments.
Iterative Process Planning: Work breakdown structures, planning guidelines, cost and schedule estimating, Iteration planning process, Pragmatic planning.

UNIT- IV:
Project Organizations and Responsibilities: Line-of-Business Organizations, Project Organizations, evolution of Organizations.
Process Automation: Automation Building blocks, The Project Environment.
Project Control and Process instrumentation: The seven core Metrics, Management indicators, quality indicators, life cycle expectations, pragmatic Software Metrics, Metrics automation.

UNIT-V:
Agile Methodology, ADAPTing to Scrum, Patterns for Adopting Scrum, Iterating towards Agility. **Fundamentals of DevOps**: Architecture, Deployments, Orchestration, Need, Instance of applications, DevOps delivery pipeline, DevOps eco system. DevOps adoption in projects: Technology aspects, Agiling capabilities, Tool stack implementation, People aspect, processes
Course Outcomes (COs)

1. Understand software management models, software economics, and cost estimation techniques.
2. Explore the software development life cycle and key project artifacts.

3. Learn iterative process planning, work breakdown structures, and project scheduling.
4. Understand project organization, process automation, and control metrics.
5. Gain knowledge of Agile, Scrum, and DevOps practices for software projects.

Text Books:

1. Software Project Management, Walker Royce, PEA, 2005.
2. Succeeding with Agile: Software Development Using Scrum, Mike Cohn, Addison Wesley.
3. The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations, Gene Kim , John Willis , Patrick Debois , Jez Humb,1st Edition, O'Reilly publications, 2016.

Reference Books:

1. Software Project Management, Bob Hughes,3/e, Mike Cotterell, TMH
2. Software Project Management, Joel Henry, PEA
3. Software Project Management in practice, Pankaj Jalote, PEA, 2005,
4. Effective Software Project Management, Robert K.Wysocki, Wiley,2006.
5. Project Management in IT, Kathy Schwalbe, Cengage

III Year II Semester	Mobile Adhoc Networks (Professional Elective-III)	L	T	P	C
		3	0	0	3

Course Objectives:
From the course the student will learn
- Architect sensor networks for various application setups.
- Devise appropriate data dissemination protocols and model links cost.
- Understanding of the fundamental concepts of wireless sensor networks and has a basic knowledge of the various protocols at various layers.
- Evaluate the performance of sensor networks and identify bottlenecks.

UNIT I: Introduction to Ad Hoc Wireless Networks- Cellular and Ad Hoc Wireless Networks, Characteristics of MANETs, Applications of MANETs, Issues and Challenges of MANETs, Ad Hoc Wireless Internet, MAC protocols for Ad hoc Wireless Networks-Issues, Design Goals and Classifications of the MAC Protocols.

UNIT II: Routing Protocols for Ad Hoc Wireless Networks- Issues in Designing a Routing Protocol, Classifications of Routing Protocols, Topology-based versus Position-based Approaches, Issues and design goals of a Transport layer protocol, Classification of Transport layer solutions, TCP over Ad hoc Wireless Networks, Solutions for TCP over Ad Hoc Wireless Networks, Other Transport layer protocols.

UNIT III: Security protocols for Ad hoc Wireless Networks- Security in Ad hoc Wireless Networks, Network Security Requirements, Issues and Challenges in Security Provisioning, Network Security Attacks, Key Management, Secure Routing in Ad hoc Wireless Networks, Cooperation in MANETs, Intrusion Detection Systems.
UNIT IV: Basics of Wireless Sensors and Applications- The Mica Mote, Sensing and Communication Range, Design Issues, Energy Consumption, Clustering of Sensors, Applications, Data Retrieval in Sensor Networks-Classification of WSNs, MAC layer, Routing layer, Transport layer, High-level application layer support, Adapting to the inherent dynamic nature of WSNs.

UNIT V: Security in WSNs- Security in WSNs, Key Management in WSNs, Secure Data Aggregation in WSNs, Sensor Network Hardware-Components of Sensor Mote, Sensor Network Operating Systems–TinyOS, LA-TinyOS, SOS, RETOS, Imperative Language-nesC, **Dataflow Style Language**-TinyGALS, Node-Level Simulators, NS-2 and its sensor network extension, TOSSIM.
Course Outcomes:
1. Understand MANET fundamentals, characteristics, and MAC protocol design.
2. Learn routing and transport protocols with a focus on TCP in ad hoc networks.
3. Explore security threats, key management, and secure routing in MANETs.
4. Understand components, energy usage, and data retrieval in Wireless Sensor Networks.
5. Learn security challenges, key management, and secure data aggregation in WSNs.

Text Books:
1. Ad Hoc Wireless Networks – Architectures and Protocols, 1st edition, C. Siva Ram Murthy, B. S. Murthy, Pearson Education, 2004
2. Ad Hoc and Sensor Networks – Theory and Applications, 2nd edition *Carlos Corderio Dharma P.Aggarwal,* World Scientific Publications / Cambridge University Press, March 2006

Reference Books:
1. Wireless Sensor Networks: An Information Processing Approach, 1st edition, *Feng Zhao, Leonidas Guibas*, Elsevier Science imprint, Morgan Kauffman Publishers, 2005, rp2009
2. Wireless Ad hoc Mobile Wireless Networks – Principles, Protocols and Applications, 1st edition, Subir Kumar Sarkar, et al., Auerbach Publications, Taylor & Francis Group, 2008

3. Ad hoc Networking, 1st edition,*Charles E.Perkins*, Pearson Education, 2001
4. Wireless Ad hoc Networking, 1st edition, *Shih-Lin Wu, Yu-Chee Tseng,* Auerbach Publications, Taylor & Francis Group, 2007
5. Wireless Sensor Networks – Principles and Practice, 1st edition, Fei Hu, Xiaojun Cao, An Auerbach book, CRC Press, Taylor & Francis Group, 2010

III Year II Semester	Natural Language Processing (Professional Elective-III)	L	T	P	C
		3	0	0	3

Course Objectives:

This course introduces the fundamental concepts and techniques of natural language processing (NLP).

- Students will gain an in-depth understanding of the computational properties of natural languages and the commonly used algorithms for processing linguistic information.
- The course examines NLP models and algorithms using both the traditional symbolic and the more recent statistical approaches.
- Enable students to be capable to describe the application based on natural language processing and to show the points of syntactic, semantic and pragmatic processing.

UNIT I:

INTRODUCTION: Origins and challenges of NLP – Language Modeling: Grammar-based LM, Statistical LM – Regular Expressions, Finite-State Automata – English Morphology, Transducers for lexicon and rules, Tokenization, Detecting and Correcting Spelling Errors, Minimum Edit Distance.

UNIT II:

WORD LEVEL ANALYSIS: Unsmoothed N-grams, Evaluating N-grams, Smoothing, Interpolation and Backoff – Word Classes, Part- of-Speech Tagging, Rule-based, Stochastic and Transformation-based tagging, Issues in PoS tagging – Hidden Markov and Maximum Entropy models.

UNIT III:

SYNTACTIC ANALYSIS: Context-Free Grammars, Grammar rules for English, Treebanks, Normal Forms for grammar – Dependency Grammar – Syntactic Parsing, Ambiguity, Dynamic Programming parsing – Shallow parsing Probabilistic CFG, Probabilistic CYK, Probabilistic Lexicalized CFGs – Feature structures, Unification of feature structures

UNIT IV:

SEMANTICS AND PRAGMATICS: Requirements for representation, First-Order Logic, Description Logics – Syntax-Driven Semantic analysis, Semantic attachments – Word Senses, Relations between Senses, Thematic Roles, selectional restrictions – Word Sense Disambiguation, WSD using Supervised, Dictionary & Thesaurus, Bootstrapping methods – Word Similarity using Thesaurus and Distributional methods.

UNIT V:

DISCOURSE ANALYSIS AND LEXICAL RESOURCES: Discourse segmentation, Coherence – Reference Phenomena, Anaphora Resolution using Hobbs and Centering Algorithm – Coreference Resolution – Resources: Porter Stemmer, Lemmatizer, Penn Treebank, Brill's Tagger, WordNet, PropBank, FrameNet, Brown Corpus, British National Corpus (BNC).

Course Outcomes:

1. Understand language models, tokenization, and finite-state automata.
2. Perform word-level analysis using N-grams, smoothing, and POS tagging.
3. Learn syntactic analysis with grammars, parsing, and probabilistic models.
4. Study semantics, word sense disambiguation, and semantic interpretation.
5. Apply discourse analysis and use lexical resources like WordNet and FrameNet.

Text Books:

1. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech, 2nd Edition, Daniel Jurafsky, James H. Martin -Pearson Publication,2024.
2. Natural Language Processing with Python, First Edition, Steven Bird, Ewan Klein and Edward Loper, OReilly Media,2009.

Reference Books:

1. Language Processing with Java and Ling Pipe Cookbook, 1st Edition, Breck Baldwin, Atlantic Publisher, 2015.
2. Natural Language Processing with Java, 2nd Edition, Richard M Reese, OReilly Media,2015.
3. Handbook of Natural Language Processing, Second, NitinIndurkhya and Fred J. Damerau, Chapman and Hall/CRC Press, 2010.Edition
4. Natural Language Processing and Information Retrieval, 3rd Edition, TanveerSiddiqui, U.S. Tiwary, Oxford University Press,2008.

III Year II Semester	Distributed Operating System (Professional Elective-III)	L	T	P	C
		3	0	0	3

Course Objectives:
The main objective of the course is to introduce design issues and different message passing techniques in DOS, distributed systems, RPC implementation and its performance in DOS, distributed shared memory and resource management, distributed file systems and evaluate the performance in terms of fault tolerance, file replication as major factors

Unit I:
Fundamentals:
What are Distributed Computing Systems? Evolution of Distributed Computing System; Distributed Computing System Models; what is Distributed Operating System? Issues in Designing a Distributed Operating System; Introduction to Distributed Computing Environment (DCE).
Message Passing:
Introduction, Desirable features of a Good Message Passing System, Issues in PC by Message Passing, Synchronization, Buffering, Multi-datagram Messages, Encoding and Decoding of Message Data, Process Addressing, Failure Handling, Group Communication, Case Study: 4.3 BSD UNIX IPC Mechanism.

Unit II:Remote Procedure Calls:
Introduction, The RPC Model, Transparency of RPC, Implementing RPC Mechanism, Stub Generation, RPC Messages, Marshaling Arguments and Results, Server Management, Parameter-Passing Semantics, Call Semantics, Communication Protocols for RPCs, Complicated RPCs, Client-Server Binding, Exception Handling, Security, Some Special Types of RPCs, RPC in Heterogeneous Environments, Lightweight RPC, Optimization for Better Performance, Case Studies: Sun RPC

Unit III: Distributed Shared Memory:
Introduction, General Architecture of DSM systems, Design and Implementation Issues of DSM, Granularity, Structure of Shared Memory Space, Consistency Models, Replacement Strategy, Thrashing, Other approaches to DSM, Heterogeneous DSM, Advantages of DSM. Synchronization: Introduction, Clock Synchronization, Event Ordering, Mutual Exclusion, Dead Lock, Election Algorithms

Unit IV: Resource Management:
Introduction, Desirable Features of a Good Global Scheduling Algorithm, Task Assignment Approach, Load – Balancing Approach, Load – Sharing Approach Process Management: Introduction, Process Migration, Threads.

Unit V: Distributed File Systems:
Introduction, Desirable Features of a Good Distributed File System, File models, File–Accessing Models, File – Sharing Semantics, File – Caching Schemes, File Replication, Fault Tolerance, Atomic Transactions and Design Principles.

Course Outcomes:

1. Understand Distributed Operating Systems, their evolution, models, and design issues.
2. Learn message passing, synchronization, buffering, and group communication in distributed systems.
3. Understand Remote Procedure Call (RPC) concepts, implementation, and security.
4. Study Distributed Shared Memory (DSM) architecture, consistency, and synchronization.
5. Explore resource management, global scheduling, load balancing, and distributed file systems.

Text books

1. Pradeep. K. Sinha: Distributed Operating Systems: Concepts and Design, PHI, 2007.

Reference Books:

1. Andrew S. Tanenbaum: Distributed Operating Systems, Pearson Education, 2013.
2. Ajay D. Kshemkalyani and MukeshSinghal, Distributed Computing: Principles, Algorithms and Systems, Cambridge University Press, 2008
3. SunitaMahajan, Seema Shan, "Distributed Computing", Oxford University Press,2015

		L	T	P	C
III Year II Semester	**Open elective offered by other department** **Open Elective – II**	3	0	0	3

III Year II Semester	Cloud Computing Lab	L	T	P	C
		0	0	3	1.5

Course Objectives:

- To introduce the various levels of services offered by cloud.
- To give practical knowledge about working with virtualization and containers.
- To introduce the advanced concepts such as serverless computing and cloud simulation.

List of Experiments:

1. Lab on web services
2. Lab on IPC, messagaging, publish/subscribe
3. Install VirtualBox/VMware Workstation with different flavours of Linux or windows OS on top of windows8 or above.
4. Install a C compiler in the virtual machine created using VirtualBox and execute Simple Programs.
5. Create an Amazon EC2 instance and set up a web-server on the instance and associate an IP address with the instance. In the process, create a security group allowing access to port 80 on the instance.
   OR
6. Do the same with OpenStack
7. Install Google App Engine. Create a hello world app and other simple web applications using python/java.
8. Start a Docker container and set up a web-server (e.g. apache2 or Python based Flask micro web framework) on the instance. Map the host directory as a data volume for the container.
9. Find a procedure to transfer the files from one virtual machine to another virtual machine. Similarly, from one container to another container.
10. Find a procedure to launch virtual machine using trystack (Online Openstack Demo Version)
11. Install Hadoop single node cluster and run simple applications like word count.
12. Utilize OpenFaaS – Serverless computing framework and demonstrate basic event driven function invocation.
13. Simulate a cloud scenario using CloudSim and run a scheduling algorithm that is not present in CloudSim.

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

1. Demonstrate various service types, delivery models and technologies of a cloud computing environment.
2. Distinguish the services based on virtual machines and containers in the cloud offerings.
3. Assess the challenges associated with a cloud-based application.
4. Discuss advanced cloud concepts such as serverless computing and cloud simulation.
5. Examine various programming paradigms suitable to solve real world and scientific problems using cloud services.

Text Books:

1. Mastering Cloud Computing, 2nd edition, Rajkumar Buyya, Christian Vecchiola, Thamarai Selvi, Shivananda Poojara, Satish N. Srirama, McGraw Hill, 2024.
2. Distributed and Cloud Computing, Kai Hwang, Geoffery C. Fox, Jack J. Dongarra, Elsevier, 2012.

Reference Books:

1. Cloud Computing, Theory and Practice, Dan C Marinescu, 2nd edition, MK Elsevier, 2018.
2. Cloud Computing: Principles and Paradigms by Rajkumar Buyya, James Broberg and Andrzej M. Goscinski, Wiley, 2011.
3. Online documentation and tutorials from cloud service providers (e.g. AWS, Google App Engine)
4. Docker, Reference documentation, https://docs.docker.com/reference/
5. OpenFaaS, Serverless Functions Made Simple, https://docs.openfaas.com/

III Year II Semester	Machine Learning Lab	L	T	P	C
		0	0	3	1.5

Course Objectives:

- To learn about computing central tendency measures and Data preprocessing techniques
- To learn about classification and regression algorithms
- To apply different clustering algorithms for a problem.

Software Required: Python/R/Weka

Lab should cover the concepts studied in the course work, sample list of Experiments:

1. Compute Central Tendency Measures: Mean, Median, Mode Measure of Dispersion: Variance, Standard Deviation.
2. Apply the following Pre-processing techniques for a given dataset.
   a. Attribute selection b. Handling Missing Values c. Discretization
   d. Elimination of Outliers
3. Apply KNN algorithm for classification and regression
4. Demonstrate decision tree algorithm for a classification problem and perform parameter tuning for better results
5. Demonstrate decision tree algorithm for a regression problem
6. Apply Random Forest algorithm for classification and regression
7. Demonstrate Naïve Bayes Classification algorithm.
8. Apply Support Vector algorithm for classification
9. Demonstrate simple linear regression algorithm for a regression problem
10. Apply Logistic regression algorithm for a classification problem
11. Demonstrate Multi-layer Perceptron algorithm for a classification problem
12. Implement the K-means algorithm and apply it to the data you selected. Evaluate performance by measuring the sum of the Euclidean distance of each example from its class center. Test the performance of the algorithm as a function of the parameters K.
13. Demonstrate the use of Fuzzy C-Means Clustering
14. Demonstrate the use of Expectation Maximization based clustering algorithm

Course outcomes:

1. Analyze datasets using statistical measures of central tendency and dispersion.
2. Apply data preprocessing techniques for cleaning and preparing datasets.
3. Implement and evaluate machine learning algorithms for classification and regression.
4. Apply clustering techniques like K-Means, Fuzzy C-Means, and EM for data grouping.
5. Develop and optimize neural networks using Multi-layer Perceptron for classification.

III Year II Semester	Soft skills (Skill Enhancement Course)	L	T	P	C
		0	1	2	2

Course Objectives:

- To equip the students with the skills to effectively communicate in English
- To train the students in interview skills, group discussions and presentation skills
- To motivate the students to develop confidence
- To enhance the students' interpersonal skills
- To improve the students' writing skills

UNIT – I

Analytical Thinking & Listening Skills: Self-Introduction, Shaping Young Minds - A Talk by Azim Premji (Listening Activity), Self – Analysis, Developing Positive Attitude, Perception.
Communication Skills: Verbal Communication; Non Verbal Communication (Body Language)

UNIT – II

Self-Management Skills: Anger Management, Stress Management, Time Management, Six Thinking Hats, Team Building, Leadership Qualities
Etiquette: Social Etiquette, Business Etiquette, Telephone Etiquette, Dining Etiquette

UNIT – III

Standard Operation Methods : Basic Grammars, Tenses, Prepositions, Pronunciation, Letter Writing; Note Making, Note Taking, Minutes Preparation, Email & Letter Writing

UNIT-IV

Job-Oriented Skills: Group Discussion, Mock Group Discussions, Resume Preparation, Interview Skills, Mock Interviews

UNIT-V

Interpersonal relationships: Introduction, Importance, Types, Uses, Factors affecting interpersonal relationships, Accommodating different styles, Consequences of interpersonal relationships

Course outcomes:

1. Enhance analytical thinking and communication skills, including body language and active listening.
2. Apply self-management techniques like stress, anger, and time management with leadership skills.
3. Develop professional writing, documentation, grammar, and pronunciation abilities.
4. Build job readiness through group discussions, resume writing, and interview preparation.
5. Strengthen interpersonal skills and manage relationships in diverse professional settings.

Text books:

1. Barun K. Mitra, Personality Development and Soft Skills, Oxford University Press, 2011.
2. S.P. Dhanavel, English and Soft Skills, Orient Blackswan, 2010.

Reference books:

1. R.S.Aggarwal, A Modern Approach to Verbal & Non-Verbal Reasoning, S.Chand& Company Ltd., 2018.
2. Raman, Meenakshi& Sharma, Sangeeta, Technical Communication Principles and Practice, Oxford University Press, 2011.

E-resources:

1. https://swayam-plus.swayam2.ac.in/courses/course-details?id=P_CAMBR_01

III Year II Semester	Technical Paper Writing & IPR	L	T	P	C
		2	0	0	-

Course Objective: The course will explain the basic related to writing the technical reports and understanding the concepts related to formatting and structuring the report. This will help students to comprehend the concept of proofreading, proposals and practice

Unit I:
Introduction: An introduction to writing technical reports, technical sentences formation, using transitions to join sentences, Using tenses for technical writing.
Planning and Structuring: Planning the report, identifying reader(s), Voice, Formatting and structuring the report, Sections of a technical report, Minutes of meeting writing.

Unit II:
Drafting report and design issues: The use of drafts, Illustrations and graphics.
Final edits: Grammar, spelling, readability and writing in plain English: Writing in plain English, Jargon and final layout issues, Spelling, punctuation and Grammar, Padding, Paragraphs, Ambiguity.

Unit III:
Proofreading and summaries: Proofreading, summaries, Activities on summaries.
Presenting final reports: Printed presentation, Verbal presentation skills, Introduction to proposals and practice.

Unit IV: Using word processor:
Adding a Table of Contents, Updating the Table of Contents, Deleting the Table of Contents, Adding an Index, Creating an Outline, Adding Comments, Tracking Changes, Viewing Changes, Additions, and Comments, Accepting and Rejecting Changes , Working with Footnotes and Endnotes, Inserting citations and Bibliography, Comparing Documents, Combining Documents, Mark documents final and make them read only., Password protect Microsoft Word documents., Using Macros,

Unit V:
Nature of Intellectual Property: Patents, Designs, Trade and Copyright. Process of **Patenting and Development:** technological research, innovation, patenting, development. International Scenario: International cooperation on Intellectual Property
Course outcomes:

1. Master technical writing to create clear, concise, and well-structured reports.
2. Enhance drafting, editing, and use of illustrations for improving technical documents.
3. Develop proofreading and summarizing skills for effective reporting and presentations.
4. Gain proficiency in advanced word processing tools for document creation and security.
5. Understand intellectual property rights, patents, and the global patenting process.

Text Books:

1. Kompal Bansal & Parshit Bansal, “Fundamentals of IPR for Beginner’s”, 1st Ed., BS Publications, 2016.
2. William S. Pfeiffer and Kaye A. Adkins, “Technical Communication: A Practical Approach”, Pearson.
3. Ramappa,T., “Intellectual Property Rights Under WTO”, 2nd Ed., S Chand, 2015.

Reference Books:

1. Adrian Wallwork , English for Writing Research Papers, Springer New York Dordrecht Heidelberg London, 2011.
2. Day R, How to Write and Publish a Scientific Paper, Cambridge University Press(2006)

E-resources:

1. https://www.udemy.com/course/reportwriting/
2. https://www.udemy.com/course/professional-business-english-and-technical-report-writing/
3. https://www.udemy.com/course/betterbusinesswriting/

B.Tech. IV Year I Semester

IV Year I Semester	Internet of Things	L	T	P	C
		3	0	0	3

Course Objectives:

From the course the student will learn

- the application areas of IOT
- the revolution of Internet in Mobile Devices, Cloud & Sensor Networks
- building blocks of Internet of Things and characteristics

UNIT I:

Predecessors of IoT: Introduction, Wireless Sensor Networks, Machine-to-Machine Communications, Cyber Physical Systems

Emergence of IoT: Introduction, Evolution of IoT, Enabling IoT and the Complex Interdependence of Technologies, IoT Networking Components, Addressing Strategies in IoT

UNIT II:

IoT Sensing and Actuation: Introduction, Sensors, Sensor Characteristics, Sensorial Deviations, Sensing Types, Sensing Considerations, Actuators, Actuator Types, Actuator Characteristics

IoT Processing Topologies and Types: Data Format, Importance of Processing in IoT, Processing Topologies, IoT Device Design and Selection Considerations, Processing Offloading.

UNIT III:

IoT Connectivity Technologies: Introduction, IEEE 802.15.4, Zigbee, Thread, ISA100.11A, WirelessHART, RFID, NFC,DASH7, Z-Wave, Weightless, Sigfox, LoRa, NB-IT, Wi-Fi, Bluetooth

IoT Communication Technologies: Introduction, Infrastructure Protocols, Discovery Protocols, Data Protocols, Identification Protocols, Device Management, Semantic Protocols.

UNIT IV:

IoT Interoperability: Introduction, Standards, Frameworks

Fog Computing and Its Applications: Introduction, View of Fog Computing Architecture, Fog Computing in IoT, Selected Applications of Fog Computing

UNIT V:

Paradigms, Challenges, and the Future: Introduction, Evolution of New IoT Paradigms, Challenges Associated with IoT, Emerging Pillars of IoT

IoT Case Studies: Agricultural IoT, Vehicular IoT

Course outcomes:

1. Understand IoT evolution, components, and enabling technologies like WSN and M2M.
2. Explore IoT sensing, actuation devices, and data processing methods.
3. Learn IoT connectivity options and communication protocols like Zigbee, LoRa, and Wi-Fi.
4. Understand IoT standards, interoperability, and the role of Fog Computing.
5. Analyze IoT paradigms and real-world applications in agriculture and smart vehicles.

Text Books:

6. Introduction to IoT, Sudip Misra, Anandarup Mukhaerjee, Arjit Roy, Cambridge University Press, 2021

7. Internet of Things: Architecture, Design Principles and Applications, Rajkamal, McGraw Hill Higher Education

Reference Books:

1. Fog and Edge Computing: Principles and Paradigms, Rajkumar Buyya (Editor), Satish narayana Srirama (Editor) , ISBN: 978-1-119-52498-4, January 2019
2. Getting Started with the Internet of Things, CunoPfister , Oreilly

IV Year I Semester	Human Resources & Project Management	L	T	P	C
		3	0	0	3

Course Objectives:

1. To impart knowledge of the fundamental concepts and functions of human resource management in organizations.
2. To familiarize students with the recruitment and selection processes and the importance of training and development.
3. To understand the significance of performance management, compensation strategies, and employee motivation.
4. To explore industrial relations, grievance handling, and labor legislation.
5. To analyze the role of HR in contemporary issues such as globalization, diversity, and technological advancement.

UNIT – I:

Human Resource Management – Definition, Nature, Scope and Objectives – Functions and Importance of HRM – Role of HR Manager – HR Policies – Evolution of HRM – Challenges of HRM in the 21st Century – Strategic Human Resource Management – HRM vs Personnel Management.

UNIT – II:

Human Resource Planning – Process and Importance – Job Analysis and Design – Recruitment: Sources and Process – Selection: Steps and Methods – Placement and Induction – Trends in Talent Acquisition – Outsourcing and Employer Branding – Use of Technology in Recruitment.

UNIT – III:

Training and Development – Objectives and Importance – Types of Training – Training Methods – Training Needs Assessment – Evaluation of Training Programs – Career Planning and Development – Promotion, Transfer, and Separation – Succession Planning – Mentoring and Coaching.

UNIT – IV:

Performance Management – Concept, Objectives, and Process – Performance Appraisal Methods – Limitations and Errors – Compensation Management – Components of Compensation – Incentives and Benefits – Employee Motivation – Theories of Motivation – Job Satisfaction and Employee Engagement.

UNIT – V:

Industrial Relations – Concept, Importance, and Parties – Trade Unions – Collective Bargaining – Grievance Handling Procedure – Employee Discipline – Labor Legislation in India – Factories Act, Industrial Disputes Act, and Workmen Compensation Act – Recent Trends in HRM – E-HRM, HR Analytics, Global HRM, Diversity and Inclusion.

Course Outcomes:

After successful completion of the course, the student will be able to:

1. Demonstrate understanding of human resource management concepts and their application in the corporate world.
2. Explain and apply the processes of recruitment, selection, training, and employee development.
3. Evaluate performance management systems and design suitable compensation packages.
4. Analyze industrial disputes and labor relations and handle employee grievances effectively.

5. Discuss emerging HRM trends and practices in the context of globalization and technology.

Text Books:

1. K. Aswathappa, **"Human Resource Management: Text and Cases"**, McGraw-Hill Education, 8th Edition.
2. V.S.P. Rao, **"Human Resource Management"**, Excel Books, 3rd Edition.
3. Gary Dessler, **"Human Resource Management"**, Pearson Education, 15th Edition.

Reference Books:

1. Michael Armstrong, **"A Handbook of Human Resource Management Practice"**, Kogan Page, 13th Edition.
2. Decenzo and Robbins, **"Fundamentals of Human Resource Management"**, Wiley, 11th Edition.
3. P. Subba Rao, **"Essentials of Human Resource Management and Industrial Relations"**, Himalaya Publishing House, Revised Edition.

IV Year I Semester	Software Architecture & Design Patterns (Professional Elective-IV)	L	T	P	C
		3	0	0	3

Course objectives:

1. Explain the fundamentals of design patterns and object-oriented design concepts.
2. Analyze system requirements and identify key classes and relationships for software design.
3. Understand and apply structural design patterns to solve common design problems.
4. Design interactive systems using the Model-View-Controller (MVC) architecture.
5. Develop distributed object-oriented applications using technologies like Java RMI and Web services.

UNIT - I

Introduction: What is a design pattern? Describing design patterns, the catalog of design pattern, organizing the catalog, how design patterns solve design problems, how to select a design pattern, how to use a design pattern What is object oriented development? key concepts of object oriented design other related concepts, benefits and drawbacks of the paradigm

UNIT – II

Analysis a System: Overview of the analysis phase, stage 1 gathering the requirements functional requirements specification, defining conceptual classes and relationships, using the knowledge of the domain Design and Implementation, discussions and further reading

UNIT – III

Design Pattern Catalog: Structural patterns, Adapter, bridge, composite, decorator, facade, flyweight, proxy.

UNIT – IV

Interactive systems and the MVC architecture: Introduction The MVC architectural pattern, analyzing a simple drawing program designing the system, designing of the subsystems, getting into implementation, implementing undo operation drawing incomplete items, adding a new feature pattern based solutions

UNIT – V

Designing with Distributed Objects: Client server system, java remote method invocation, implementing an object oriented system on the web, Web services (SOAP, Restful), Enterprise Service Bus.

Course Outcomes

1. Understand the basic concepts to identify state behavior of real world objects
2. Apply Object Oriented Analysis and Design concepts to solve complex problems
3. Construct various UML models using the appropriate notation for specific problem context
4. Design models to Show the importance of systems analysis and design in solving complex problems using case studies
5. Study of Pattern Oriented approach for real world problems

Text Books:

1. Object oriented analysis, design and implementation, brahma dathan, sarnath rammath , universities press,2013
2. Design patterns, Erich Gamma, Richard helan , Ralph johman , john vlissides, PEARSON Publication,2013

Reference Books:

1. Frank Bachmann, Regine Meunier , Hans Rohnert “Pattern Oriented Software Architecture” Volume 1, 1996.
2. William J Brown et al., "Anti Patterns: Refactoring Software, Architectures and Projects in Crisis", John Wiley, 1998

IV Year I Semester	Deep Learning (Professional Elective-IV)	L	T	P	C
		3	0	0	3

Course Objectives: The main objective of the course is to make students:

- Learn deep learning methods for working with sequential data,
- Learn deep recurrent and memory networks,
- Learn deep Turing machines,
- Apply such deep learning mechanisms to various learning problems.
- Know the open issues in deep learning, and have a grasp of the current research directions.

UNIT I:

Fundamentals of Deep Learning: Artificial Intelligence, History of Machine learning: Probabilistic Modeling, Early Neural Networks, Kernel Methods, Random forests and Gradient Boosting Machines, **Fundamentals of Machine Learning:** Four Branches of Machine Learning, Evaluating Machine learning Models. **[Text Book 2]**

UNIT II:Introducing Deep Learning: Biological and Machine Vision, Human and Machine Language, Artificial Neural Networks, Training Deep Networks, Improving Deep Networks. **[Text Book3]**

UNIT III:Neural Networks: Anatomy of Neural Network, Introduction to Keras: Keras, TensorFlow, Theano and CNTK, Setting up Deep Learning Workstation, Classifying Movie Reviews: Binary Classification, Classifying newswires: Multiclass Classification. **[Text Book 2]**

UNIT IV:

Convolutional Neural Networks: Nerual Network and Representation Learing, Convolutional Layers, Multichannel Convolution Operation, **Recurrent Neural Networks:** Introduction to RNN, RNN Code, PyTorch Tensors: Deep Learning with PyTorch, CNN in PyTorch. LSTM, Attention Mechanism **[Text Book 3]**

UNIT V:

Interactive Applications of Deep Learning: Machine Vision, Natural Language processing, Generative Adversial Networks, Deep Reinforcement Learning. **[Text Book 1]**

Deep Learning Research: Autoencoders**,** Deep Generative Models: Boltzmann Machines Restricted Boltzmann Machines, Deep Belief Networks. **[Text Book 1]**

Course outcomes:

1. **Understand** the fundamentals and history of machine learning and deep learning concepts.
2. **Explain** the architecture and training of artificial neural networks and deep neural networks.
3. **Apply** deep learning frameworks like Keras, TensorFlow, and PyTorch for building neural network models.
4. **Develop** and implement convolutional and recurrent neural networks for various applications.
5. **Analyze** advanced deep learning topics including GANs, reinforcement learning, and generative models.

Text Books:

1. Deep Learning- Ian Goodfellow, Yoshua Bengio and Aaron Courvile, MIT Press, 2016
2. Deep Learning with Python - Francois Chollet, Released December 2017, Publisher(s): Manning Publications, ISBN: 9781617294433
3. Deep Learning Illustrated: A Visual, Interactive Guide to Artificial Intelligence - Jon Krohn, Grant Beyleveld, Aglaé Bassens, Released September 2019, Publisher(s): Addison-Wesley Professional, ISBN: 9780135116821
4. Deep Learning from Scratch - Seth Weidman, Released September 2019, Publisher(s): O'Reilly Media, Inc., ISBN: 9781492041412

Reference Books:

1. Artificial Neural Networks, Yegnanarayana, B., PHI Learning Pvt. Ltd, 2009.
2. Matrix Computations, Golub, G., H., and Van Loan,C.,F, JHU Press,2013.
3. Neural Networks: A Classroom Approach, Satish Kumar, Tata McGraw-Hill Education, 2004.

Web Link:

1. Swayam NPTEL: Deep Learning: https://onlinecourses.nptel.ac.in/noc22_cs22/preview

IV Year I Semester	Computer Vision (Professional Elective-IV)	L	T	P	C
		3	0	0	3

Course Objectives:

- To understand the Fundamental Concepts related to sources, shadows and shading
- To understand the Geometry of Multiple Views

UNIT –I:

CAMERAS: Pinhole Cameras Radiometry – Measuring Light: Light in Space, Light Surfaces, Important Special Cases Sources, Shadows, And Shading: Qualitative Radiometry, Sources and Their Effects, Local Shading Models, Application: Photometric Stereo, Interreflections: Global Shading Models Color: The Physics of Color, Human Color Perception, Representing Color, A Model for Image Color, Surface Color from Image Color.

UNIT-II:

Linear Filters: Linear Filters and Convolution, Shift Invariant Linear Systems, Spatial Frequency and Fourier Transforms, Sampling and Aliasing, Filters as Templates, Edge Detection:Noise, Estimating Derivatives, Detecting Edges Texture0:Representing Texture, Analysis (and Synthesis) Using Oriented Pyramids, Application: Synthesis by Sampling Local Models, Shape from Texture.

UNIT-III:

The Geometry of Multiple Views: Two Views Stereopsis: Reconstruction, Human Stereopsis, Binocular Fusion, Using More Cameras Segmentation by Clustering: What Is Segmentation? Human Vision: Grouping and Getstalt, Applications: Shot Boundary Detection and Background Subtraction, Image Segmentation by Clustering Pixels, Segmentation by Graph-Theoretic Clustering,

UNIT-IV:

Segmentation by Fitting a Model: The Hough Transform, Fitting Lines, Fitting Curves, Fitting as a Probabilistic Inference Problem, Robustness Segmentation and Fitting Using Probabilistic Methods: Missing Data Problems, Fitting, and Segmentation, The EM Algorithm in Practice, Tracking With Linear Dynamic Models: Tracking as an Abstract Inference Problem, Linear Dynamic Models, Kalman Filtering, Data Association, Applications and Examples

UNIT- V:

Geometric Camera Models: Elements of Analytical Euclidean Geometry, Camera Parameters and the Perspective Projection, Affine Cameras and Affine Projection Equations Geometric Camera Calibration: Least-Squares Parameter Estimation, A Linear Approach to Camera Calibration, Taking Radial Distortion into Account, Analytical Photogrammetry,
Case study: Mobile Robot Localization Model- Based Vision: Initial Assumptions, Obtaining Hypotheses by Pose Consistency, Obtaining Hypotheses by pose Clustering, Obtaining Hypotheses Using Invariants, Verification, Case study: Registration in Medical Imaging Systems, Curved Surfaces and Alignment.

Course outcomes:

1. Understand the principles of camera models, radiometry, and color perception in computer vision.
2. Apply linear filtering techniques and edge detection methods for image processing.
3. Analyze multi-view geometry concepts and segmentation techniques for image understanding.
4. Implement probabilistic methods and dynamic models for image segmentation and object tracking.
5. Perform camera calibration and geometric modeling for applications like robot localization and medical image registration.

Text Books:

1. David A. Forsyth and Jean Ponce: Computer Vision – A Modern Approach, PHI Learning (Indian Edition), 2009.

Reference Books:

1. E. R. Davies: Computer and Machine Vision – Theory, Algorithms and Practicalities, Elsevier (Academic Press), 4th edition, 2013.
2. R. C. Gonzalez and R. E. Woods "Digital Image Processing" Addison Wesley 2008. 3. Richard Szeliski "Computer Vision: Algorithms and Applications" Springer-Verlag London Limited 2011.

IV Year I Semester	Blockchain Technology (Professional Elective-IV)	L	T	P	C
		3	0	0	3

Course Objectives:

1. To learn the fundamentals of Block Chain and various types of block chain and consensus mechanism.
2. To understand public block chain system, Private block chain system and consortium block chain.
3. Able to know the security issues of blockchain technology.

UNIT – I:

Fundamentals of Blockchain: Introduction, Origin of Blockchain, Blockchain Solution, Components of Blockchain, Block in a Blockchain, The Technology and the Future.

Blockchain Types and Consensus Mechanism: Introduction, Decentralization and Distribution, Types of Blockchain, Consensus Protocol.

Cryptocurrency: Bitcoin, Altcoin and Token: Introduction, Bitcoin and the Cryptocurrency, Cryptocurrency Basics, Types of Cryptocurrencies, Cryptocurrency Usage.

UNIT – II:

Public Blockchain System: Introduction, Public Blockchain, Popular Public Blockchains, The Bitcoin Blockchain, Ethereum Blockchain.

Smart Contracts: Introduction, Smart Contract, Characteristics of a Smart Contract, Types of Smart Contracts, Types of Oracles, Smart Contracts in Ethereum, Smart Contracts in Industry.

UNIT – III:

Private Blockchain System: Introduction, Key Characteristics of Private Blockchain, Private Blockchain, Private Blockchain Examples, Private Blockchain and Open Source, E-commerce Site Example, Various Commands (Instructions) in E-commerce Blockchain, Smart Contract in Private Environment, State Machine, Different Algorithms of Permissioned Blockchain, Byzantine Fault, Multichain.

Consortium Blockchain: Introduction, Key Characteristics of Consortium Blockchain, Need of Consortium Blockchain, Hyperledger Platform, Overview of Ripple, Overview of Corda.

Initial Coin Offering: Introduction, Blockchain Fundraising Methods, Launching an ICO, Investing in an ICO, Pros and Cons of Initial Coin Offering, Successful Initial Coin Offerings, Evolution of ICO, ICO Platforms.

UNIT – IV:

Security in Blockchain: Introduction, Security Aspects in Bitcoin, Security and Privacy Challenges of Blockchain in General, Performance and Scalability, Identity Management and Authentication, Regulatory Compliance and Assurance, Safeguarding Blockchain Smart Contract (DApp), Security Aspects in Hyperledger Fabric.

Applications of Blockchain: Introduction, Blockchain in Banking and Finance, Blockchain in Education, Blockchain in Energy, Blockchain in Healthcare, Blockchain in Real-estate, Blockchain in Supply Chain, The Blockchain and IoT. Limitations and Challenges of Blockchain.

UNIT – V:

Blockchain Case Studies:
Case Study 1 – Retail,
Case Study 2 – Banking and Financial Services,
Case Study 3 – Healthcare,
Case Study 4 – Energy and Utilities.

Blockchain Platform using Python: Introduction, Learn How to Use Python Online Editor, Basic Programming Using Python, Python Packages for Blockchain.

Blockchain Platform using

Course outcomes:

1. Understand blockchain fundamentals, types, and cryptocurrency components.
2. Explain public, private, and consortium blockchains with smart contracts and ICOs.
3. Analyze security challenges, privacy concerns, and compliance in blockchain systems.
4. Evaluate blockchain applications across finance, healthcare, energy, and supply chain.
5. Develop blockchain applications and smart contracts using Python tools.

Text book:

1. "Block chain Technology", Chandramouli Subramanian, Asha A.George, Abhilasj K A and Meena Karthikeyan , Universities Press.

Reference Books:

1. Blockchain Blue print for Economy, Melanie Swan, SPD Oreilly.
2. Blockchain for Business, Jai Singh Arun, Jerry Cuomo, Nitin Gauar, Pearson Addition Wesley

IV Year I Semester	Agile Methodologies (Professional Elective-V)	L	T	P	C
		3	0	0	3

Course Objectives:
The main objectives of this course are to introduce the important concepts of Agile software development Process, emphasize the role of stand-up meetings in software collaboration, impart the knowledge on values and principles in understanding agility

UNIT I:
Learning Agile: Getting Agile into your brain, Understanding Agile values, No Silver Bullet, Agile to the Rescue, adding Agile makes a difference. A fractured perspective, how a fractured perspective causes project problems. The Agile Manifesto, Purpose behind Each Practice. Individuals and Interactions Over Processes and Tools, Working Software over Comprehensive Documentation, Customer Collaboration over Contract Negotiation, Responding to Change over Following a Plan, Principles over Practices. Understanding the Elephant, Methodologies Help You Get It All in Place at Once, Where to Start with a New Methodology.

UNIT II:
The Agile Principles: The 12 Principles of Agile Software, The Customer Is Always Right, "Do as I Say, Not as I Said". Delivering the Project, Better Project Delivery for the Ebook Reader Project. Communicating and Working Together, Better Communication for the Ebook Reader Project. Project Execution—Moving the Project Along, A Better Working Environment for the Ebook Reader Project Team. Constantly Improving the Project and the Team. The Agile Project: Bringing All the Principles Together

UNIT III:
SCRUM and Self-Organizing Teams: The Rules of Scrum, Act I: I Can Haz Scrum?, Everyone on a Scrum Team owns the Project, The Scrum Master Guides the Team's Decisions, The Product Owner Helps the Team Understand the Value of the Software, Everyone Owns the Project, Scrum Has Its Own Set of Values ,Status Updates Are for Social Networks!, The Whole Team Uses the Daily Scrum, Feedback and the Visibility-Inspection-Adaptation Cycle, The Last Responsible Moment, How to Hold an Effective Daily Scrum. Sprinting into a Wall, Sprints, Planning, and Retrospectives, Iterative or Incremental? The Product Owner Makes or Breaks the Sprint, Visibility and Value, How to Plan and Run an Effective Scrum Sprint
Scrum Planning and Collective Commitment: Not Quite Expecting the Unexpected, User Stories, Velocity, and Generally Accepted Scrum Practices, Make Your Software Useful, User Stories Help Build Features Your Users Will Use, Conditions of Satisfaction, Story Points and Velocity, Burndown Charts, Planning and Running a Sprint Using Stories, Points, Tasks, and a Task Board. Victory Lap, Scrum Values Revisited, Practices Do Work Without the Values (Just Don't Call It Scrum), Is Your Company's Culture Compatible with Scrum Values.

UNIT IV:
XP And Embracing Change: Going into Overtime, The Primary Practices of XP, Programming Practices, Integration Practices, Planning Practices, Team Practices, Why Teams Resist Changes, and How the Practices Help. The Game Plan Changed, but We're Still Losing, The XP Values Help the Team Change Their Mindset, XP Helps Developers Learn to Work with Users, Practices Only "Stick" When the Team Truly Believes in Them, An Effective Mindset Starts with the XP Values, The XP Values, Paved with Good Intentions. The Momentum Shifts, Understanding the XP Principles Helps You Embrace Change, The Principles of XP, XP Principles Help You Understand Planning, XP Principles Help You Understand Practices—and Vice Versa, Feedback Loops.
XP, Simplicity, and Incremental Design: Code and Design, Code Smells and Antipatterns (or, How to Tell If You're Being Too Clever), XP Teams Look for Code Smells and Fix Them, Hooks, Edge Cases, and Code That Does Too Much. Make Code and Design Decisions at the Last Responsible Moment, Fix Technical Debt by Refactoring Mercilessly, Use Continuous

Integration to Find Design Problems, Avoid Monolithic Design, Incremental Design and the Holistic XP Practices. Teams Work Best When They Feel Like They Have Time to Think, Team Members Trust Each Other and Make Decisions Together. The XP Design, Planning, Team, and Holistic Practices Form an Ecosystem Incremental Design Versus Designing for Reuse, When Units Interact in a Simple Way, the System Can Grow Incrementally, Great Design Emerges from Simple Interactions, Final Score.

UNIT V:

Lean, Eliminating Waste, and Seeing the whole: Lean Thinking, Commitment, Options Thinking, and Set-Based Development, Creating Heroes and Magical Thinking. Eliminate Waste, use a Value Stream Map to Help See Waste Clearly, gain a Deeper Understanding of the Product, See the Whole, Find the Root Cause of Problems That You Discover. Deliver As Fast as Possible, Use an Area Chart to Visualize Work in Progress, Control Bottlenecks by Limiting Work in Progress.

Kanban, Flow, and Constantly Improving: The Principles of Kanban, find a Starting Point and Evolve Experimentally from There. Stories Go into the System; Code Comes Out, Improving Your Process with Kanban, Visualize the Workflow, Limit Work in Progress. Measure and Manage Flow, Managing Flow with WIP Limits Naturally Creates Slack. Make Process Policies Explicit So Everyone Is on the Same Page. Emergent Behavior with Kanban.

The Agile Coach: Coaches Understand Why People Don't Always Want to Change. The Principles of Coaching.

Course outcomes:

1. Understand Agile principles, values, and their impact on project success.
2. Explain Scrum roles, rules, and practices for effective team collaboration.
3. Apply Extreme Programming (XP) practices like continuous integration and design.
4. Analyze Lean principles and Kanban techniques for efficient workflow management.
5. Develop Agile coaching skills to manage teams and foster collaboration.

Text Books:

1. Andrew Stellman, Jill Alison Hart, Learning Agile, O'Reilly, 2015.

Reference Books:

1. Andrew stellman, Jennifer Green, Head first Agile, O'Reilly, 2017.
2. Rubin K , Essential Scrum : A practical guide to the most popular Agile process, Addison-Wesley, 2013

IV Year I Semester	Big Data Analytics (Professional Elective-V)	L	T	P	C
		3	0	0	3

Course Objectives: This course is aimed at enabling the students to

- To provide an overview of an exciting growing field of big data analytics.
- To introduce the tools required to manage and analyze big data like Hadoop, NoSQL, Map Reduce, HIVE, Cassandra, Spark.
- To teach the fundamental techniques and principles in achieving big data analytics with scalability and streaming capability.
- To optimize business decisions and create competitive advantage with Big Data analytics

UNIT I: big data, convergence of key trends, unstructured data, industry examples of big data, web analytics, big data and marketing, fraud and big data, risk and big data, credit risk management, big data and algorithmic trading, big data and healthcare, big data in medicine, advertising and big data, big data technologies, introduction to Hadoop, open source technologies, cloud and big data, mobile business intelligence, Crowd sourcing analytics, inter and trans firewall analytics.

UNIT II: Introduction to NoSQL, aggregate data models, aggregates, key-value and document data models, relationships, graph databases, schema less databases, materialized views, distribution models, sharding, master-slave replication, peer- peer replication, shardingand replication, consistency, relaxing consistency, version stamps, Working with Cassandra ,Table creation, loading and reading data.

UNIT III: Data formats, analyzing data with Hadoop, scaling out, Architecture of Hadoop distributed file system (HDFS), fault tolerance ,with data replication, High availability, Data locality , Map Reduce Architecture, Process flow, Java interface, data flow, Hadoop I/O, data integrity, compression, serialization. Introduction to Hive, data types and file formats, HiveQL data definition, HiveQL data manipulation, Logical joins, Window functions, Optimization, Table partitioning, Bucketing, Indexing, Join strategies.

UNIT IV: Apache spark- Advantages over Hadoop, lazy evaluation, In memory processing, DAG, Spark context, Spark Session, RDD, Transformations- Narrow and Wide, Actions, Data frames ,RDD to Data frames, Catalyst optimizer, Data Frame Transformations, Working with Dates and Timestamps, Working with Nulls in Data, Working with Complex Types, Working with JSON, Grouping, Window Functions, Joins, Data Sources, Broadcast Variables, Accumulators, Deploying Spark- On-Premises Cluster Deployments, Cluster Managers- Standalone Mode, Spark on YARN , Spark Logs, The Spark UI- Spark UI History Server, Debugging and Spark First Aid

UNIT V: Spark-Performance Tuning, Stream Processing Fundamentals, Event-Time and State full Processing - Event Time, State full Processing, Windows on Event Time- Tumbling Windows, Handling Late Data with Watermarks, Dropping Duplicates in a Stream, Structured Streaming Basics - Core Concepts, Structured Streaming in Action, Transformations on Streams, Input and Output.

Course outcomes:

1. Understand big data concepts, applications, and technologies like Hadoop and cloud platforms.
2. Explain NoSQL databases, data models, replication, sharding, and Cassandra operations.
3. Analyze Hadoop architecture, MapReduce programming, and data management with Hive.
4. Apply Apache Spark concepts including RDDs, DataFrames, and cluster deployment.
5. Demonstrate performance tuning and real-time stream processing using Spark Structured Streaming.

Text Books:

1. Big Data, Big Analytics: Emerging, Michael Minnelli, Michelle Chambers, and

AmbigaDhiraj, 1st edition ,2013
2. SPARK: The Definitive Guide, Bill Chambers &MateiZaharia, O'Reilley, 2018-first Edition.
3. Business Intelligence and Analytic Trends for Today's Businesses", Wiley, First edition-2013.
4. P. J. Sadalage and M. Fowler, "NoSQL Distilled: A Brief Guide to the Emerging World Polyglot Persistence", Addison-Wesley Professional, 2012
5. Tom White, "Hadoop: The Definitive Guide", Third Edition, O'Reilley, 2012

Reference Books:

1. "Hadoop Operations", O'Reilley, Eric Sammer, First Edition -2012.
2. "Programming Hive", O'Reilley, E. Capriolo, D. Wampler, and J. Rutherglen, 2012.
3. "HBase: The Definitive Guide", O'Reilley, Lars George, September 2011: First Edition..
4. "Cassandra: The Definitive Guide", O'Reilley, Eben Hewitt, 2010.
5. "Programming Pig", O'Reilley, Alan Gates, October 2011: First Edition

IV Year I Semester	Mobile Computing (Professional Elective-V)	L	T	P	C
		3	0	0	3

Course Objectives:
the main objectives of the course are

1. To introduce the basic concepts and principles in mobile computing. This includes major itechniques involved,and networks & systems issues for the design and implementation of mobile computing systems and applications.
2. To explore both theoritical and practical issuses of mobile computing.
3. To provide an oppurtunity for the students to understand the key components and technologies involved and to gain hands-on experience in building mobile applications.
4. To understand latest network architecture and its interfaces

UNIT-I:
Mobile Communications: An Overview- Mobile Communication-guided transmission, unguided transmission- signal propagation frequencies, antennae, modulation, modulation methods and standards for voice-oriented data communication standards, modulation methods and standards for data and voice communication, mobile computing- novel applications and limitations, mobile computing architecture, mobile system networks. Mobile devices and systems: Cellular networks and frequency reuse, Mobile smart phones, Smart mobiles and systems, handheld pocket computers, Handheld devices, Smart systems, Limitations of mobile devices

UNIT-II:
GSM and other 2G Architectures: GSM-services and system architecture, Radio interfaces of GSM, Protocols of GSM, Localization, Call handling, GPRS system architecture. Wireless medium access control, CDMA, 3G,and 4G
Communication: Modulation, Multiplexing, Controlling the medium access, Spread spectrum, Coding methods, IMT-20003G wireless communication standards, WCDMA 3G communication standards, CDMA 3G communication standards, Broadband wireless access, 4Gnetworks.

UNIT-III:
Mobile IP Network layer: IP and Mobile IP network layers: OSI layer functions, TCP/IP and Internet protocol, Mobile internet protocol; Packet delivery and Handover Management; Location Management: Agent Discovery; Mobile TCP Introduction to Mobile Adhoc network: fixed infrastructure architecture, MANET infrastructure architecture; MANET: properties, spectrum, applications; Security in Ad-hoc network; Wireless sensor networks; sensor network applications.

UNIT-IV:
Synchronization: Synchronization in mobile computing systems, Usage models for Synchronization in mobile application, Domain-dependant specific rules for data synchronization, Personal information manager, synchronization and conflict resolution strategies, synchronizer; Mobile agent: mobile agent design, aglets; Application Server

UNIT-V:
Mobile Wireless Short Range Networks and Mobile Internet: Wireless networking and wireless LAN, Wireless LAN (WLAN) architecture, IEEE 802.11protocol layers, Wireless application protocol (WAP)-WAP1.1 architecture, wireless datagram protocol (WDP), Wireless Transport Layer Security (WTLS), wireless transaction and session layers, wireless application environment.
Course outcomes:

1. Understand mobile communication architectures, wireless transmission, and mobile devices.
2. Explain GSM, 2G/3G/4G networks, access protocols, and multiple access techniques.
3. Analyze mobile IP, handover management, MANETs, and wireless sensor networks.

4. Apply synchronization techniques, mobile agent concepts, and server-side components.
5. Demonstrate knowledge of WLANs, short-range wireless networks, WAP, and WTLS protocols.

TEXTBOOK:

1. RAJ KAMAL, "Mobile Computing," second edition, Oxford.
2. ASOKE K TALUKDER, HASANAHMED, ROOPA R YAVAGAL, "Mobile Computing, Technology Applications and Service Creation" Second Edition, Mc Graw Hill.
3. UWE Hansmann, Lother Merk, Martin S. Nocklous, Thomas Stober, "Principles of Mobile Computing," Second Edition, Springer

IV Year I Semester	Cyber Physical Systems (Professional Elective-V)	L	T	P	C
		3	0	0	3

Course Objectives:

1. Understand symbolic synthesis techniques for modeling and controlling Cyber-Physical Systems (CPS).
2. Learn security requirements, attack models, and countermeasures for securing CPS.
3. Analyze synchronization challenges and distributed consensus in CPS environments.
4. Apply real-time scheduling methods considering timing, memory, and uncertainty in CPS.
5. Explore model integration, interaction models, and formal semantics for CPS design and development.

UNIT I:

Symbolic Synthesis for Cyber-Physical Systems: Introduction and Motivation, Basic Techniques - Preliminaries, Problem Definition, Solving the Synthesis Problem, Construction of Symbolic Models, Advanced Techniques: Construction of Symbolic Models, Continuous-Time Controllers, Software Tools

UNIT II:

Security of Cyber-Physical Systems: Introduction and Motivation, Basic Techniques - Cyber Security Requirements, Attack Model, Countermeasures, Advanced Techniques: System Theoretic Approaches

UNIT III:

Synchronization in Distributed Cyber-Physical Systems: Challenges in Cyber-Physical Systems, A Complexity-Reducing Technique for Synchronization, Formal Software Engineering, Distributed Consensus Algorithms, Synchronous Lockstep Executions, Time-Triggered Architecture, Related Technology, Advanced Techniques

UNIT IV:

Real-Time Scheduling for Cyber-Physical Systems: Introduction and Motivation, Basic Techniques - Scheduling with Fixed Timing Parameters, Memory Effects, Multiprocessor/Multicore Scheduling, Accommodating Variability and Uncertainty

UNIT V:

Model Integration in Cyber-Physical Systems: Introduction and Motivation, Causality, Semantic Domains for Time, Interaction Models for Computational Processes, Semantics of CPS DSMLs, Advanced Techniques, ForSpec, The Syntax of CyPhyML, Formalization of Semantics, Formalization of Language Integration.

Course outcomes:

1. Understand symbolic synthesis techniques for model construction and control in CPS.
2. Analyze security challenges and apply system-theoretic approaches to protect CPS.
3. Evaluate synchronization challenges and apply consensus algorithms in distributed CPS.
4. Apply real-time scheduling to manage timing variability and resource constraints in CPS.
5. Integrate models in CPS using formal methods, semantics, and domain-specific languages.

Text Books:

1. Raj Rajkumar, Dionisio De Niz, and Mark Klein, Cyber-Physical Systems, Addison-Wesley Professional, 2016
2. Rajeev Alur, Principles of Cyber-Physical Systems, MIT Press

Reference Books:

1. E.A.Lee, Sanjit Seshia, Introduction to Embedded Systems: A Cyber-Physical Systems Approach, MIT Press
2. Andre Platzer, Logical Foundations of Cyber-Physical Systems, (2e), Springer Publishing, 2018

IV Year I Semester	Openelective offered by other department Optional Elective-III	L	T	P	C
		3	0	0	3

IV Year I Semester	Open elective offered by other department Optional Elective-IV	L	T	P	C
		0	0	0	3

IV Year I Semester	Mean Stack Technologies (Skill Enhancement Course)	L	T	P	C
		0	1	2	2

Course Objectives

1. Learn HTML5, JavaScript, and DOM manipulation for responsive client-side development.
2. Develop backend applications using Node.js, Express.js, routing, and security practices.
3. Work with NoSQL databases like MongoDB and integrate them using TypeScript.
4. Build Angular applications with components, services, routing, and reactive programming.
5. Integrate frontend and backend using HTTP protocols to develop secure full-stack applications.

UNIT I: HTML5 & JavaScript Essentials

HTML5: Structure, Forms, Tables, Media, Security Practices. JavaScript: Syntax, Functions, DOM Manipulation, Async Programming.

Tasks:

1. **Create a static web page** using HTML5 elements (headings, images, lists, forms).
2. **Add validation logic** to the HTML form using JavaScript functions.
3. **Build an interactive component** with DOM manipulation and event handling.
4. **Fetch data from an API** using JavaScript Fetch API and display it on the page.

UNIT II: Node.js & Express.js

Node.js: Core modules, NPM, file system, creating a server.Express.js: Routing, Middleware, API development, Security practices.

Tasks:

1. **Create a simple Node.js web server** that returns a response to browser requests.
2. **Use Express.js** to define RESTful routes and return JSON data.
3. **Connect Express app to MongoDB** and perform basic CRUD operations.
4. **Implement session management** using cookies and Helmet middleware.

UNIT III: MongoDB & TypeScript

MongoDB: CRUD, indexing, aggregation, collections. TypeScript: Interfaces, Classes, Generics, Modules.

Tasks:

1. **Install MongoDB and create a cluster** using MongoDB Atlas.
2. **Write MongoDB CRUD queries** to insert, update, delete documents.
3. **Create a TypeScript app** using interfaces, classes, and functions.
4. **Build a module-based project** in TypeScript connecting to a MongoDB collection.

UNIT IV: Angular Basics & Component Development

Angular setup, Components, Templates, Data Binding.Directives (ngIf, ngFor, ngSwitch), Pipes, Forms.

Tasks:

1. **Create a new Angular component** and display content using interpolation.
2. **Implement ngIf and ngFor** to dynamically show/hide elements and list data.
3. **Build a login form** with validation using template-driven approach.
4. **Use built-in pipes** to transform data (uppercase, lowercase, date).

UNIT V: Angular Advanced Features & Backend Integration

Services, Dependency Injection, HttpClient, RxJS. Routing, Route Guards, Lazy Loading, Nested Routes.

Tasks:

1. **Create an Angular service** to fetch and display data using HttpClient.
2. **Implement routing** between components with RouterLink and route parameters.
3. **Add a route guard** to restrict access based on login status.
4. **Enable lazy loading** for a module and observe performance improvements.

Course Outcomes

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

1. **Design and develop responsive web pages** using HTML5, JavaScript, and asynchronous programming techniques.

2. **Build and deploy RESTful APIs** using Node.js and Express.js with secure middleware handling.
3. **Perform CRUD operations on MongoDB** and write modular, type-safe applications using TypeScript.
4. **Develop scalable Angular applications**, applying data binding, component architecture, and client-side routing.
5. **Integrate frontend and backend systems**, implementing services, route guards, and lazy loading for performance optimization.

Textbooks

1. *Programming the World Wide Web*, Robet W Sebesta, Pearson.
2. *Pro MEAN Stack Development*, Elad Elrom, Apress.
3. *Full Stack JavaScript Development with MEAN*, Ihrig & Bretz, SitePoint.
4. *MongoDB – The Definitive Guide*, Kristina Chodorow, O'Reilly.

Online Resources

1. Angular: https://infyspringboard.onwingspan.com/...Angular_JS
2. MongoDB: https://infyspringboard.onwingspan.com/...MongoDB
3. JavaScript: https://infyspringboard.onwingspan.com/...Javascript
4. Node.js & Express: https://infyspringboard.onwingspan.com/...Node

IV Year I Semester	Constitution of India	L	T	P	C
		2	0	0	-

Course Objectives:

- Understand the premises informing the twin themes of liberty and freedom from a civil rights perspective.
- To address the growth of Indian opinion regarding modern Indian intellectuals' constitutional role and entitlement to civil and economic rights as well as the emergence of nationhood in the early years of Indian nationalism.
- To address the role of socialism in India after the commencement of the Bolshevik Revolution in 1917 and its impact on the initial drafting of the Indian Constitution.

UNIT–I: History of Making of the IndianConstitution: History, Drafting Committee, (Composition & Working)

Philosophy of the IndianConstitution- Preamble, Salient, Features

UNIT–II: Contours of Constitutional Rights &Duties: Fundamental Rights, Right toEquality, Right toFreedom, Right againstExploitation, Right to Freedom ofReligion, Cultural and EducationalRights, Right to ConstitutionalRemedies, Directive Principles of StatePolicy, Fundamental Duties.

UNIT–III: Organs ofGovernance: Parliament, Composition, Qualifications andDisqualifications, Powers andFunctions, **Executive-** President, Governor, Council ofMinisters, Judiciary, Appointment and Transfer of Judges,Qualifications, Powers andFunctions

UNIT–IV: LocalAdministration: District's Administration head: Role andImportance, Municipalities: Introduction, Mayor and role of Elected Representative CEO of MunicipalCorporation, Pachayati raj: Introduction, PRI:ZilaPachayat, Elected officials and their roles, CEO ZilaPachayat: Position androle, Block level: Organizational Hierarchy (Differentdepartments), Village level: Role of Elected and Appointedofficials, Importance of grass rootdemocracy

UNIT–V: ElectionCommission: Election Commission: Role andFunctioning, Chief Election Commissioner and ElectionCommissioners, State Election Commission: Role andFunctioning, Institute and Bodies for the welfare of SC/ST/OBC andwomen.

Course Outcomes:

1. Understand the history and basic structure of the Indian Constitution.
2. Explain fundamental rights, duties, and directive principles.
3. Describe the functions of Parliament, Executive, and Judiciary.
4. Identify the structure and role of local government bodies.
5. Recognize the role of the Election Commission and welfare bodies.

Text Books:

1. The Constitution of India, 1st Edition, (Bare Act), GovernmentPublication, 1950
2. Framing of Indian Constitution, 1st Edition,Dr. S. N. Busi, Dr. B. R. Ambedkar 2015

Reference Books:

1. Indian Constitution Law, 7th Edition, M. P. Jain, Lexis Nexis,2014

Open Electives, offered to other department students:

Open Elective I: Principles of Operating Systems/ Computer Organization and Architecture
Open Elective II: Principles of Database Management Systems
Open Elective III: Object Oriented Programming Through Java
Open Elective IV: Principles of Software Engineering /Computer Networks

Open Elective I

Principles of Operating Systems/ Computer Organization and Architecture

Open Elective I	Principles of Operating Systems	L	T	P	C
		3	0	0	3

Course Objectives:

The main objectives of the course is to make student

- Understand the basic concepts and principles of operating systems, including process management, memory management, file systems, and Protection
- Make use of process scheduling algorithms and synchronization techniques to achieve better performance of a computer system.
- Illustrate different conditions for deadlock and their possible solutions.

UNIT - I

Operating Systems Overview: Introduction, Operating system functions, Operating systems operations, Computing environments, Free and Open-Source Operating Systems

System Structures: Operating System Services, User and Operating-System Interface, system calls, Types of System Calls, system programs, Operating system Design and Implementation, Operating system structure, Buildingand Booting an Operating System, Operating system debugging

UNIT - II

Processes: Process Concept, Process scheduling, Operations on processes, Inter-process communication. Threads and Concurrency: Multithreading models, Thread libraries, Threading issues. CPU Scheduling: Basic concepts, Scheduling criteria, Scheduling algorithms, Multiple processor scheduling.

UNIT – III

Synchronization Tools: The Critical Section Problem, Peterson's Solution, Mutex Locks, Semaphores, Monitors, Classic problems of Synchronization. Deadlocks: system Model, Deadlock characterization, Methods for handling Deadlocks, Deadlock prevention, Deadlock avoidance, Deadlock detection, Recovery from Deadlock

UNIT – IV

Memory-Management Strategies: Introduction, Contiguous memory allocation, Paging, Structure of the Page Table, Swapping. Virtual Memory Management: Introduction, Demand paging, Copy-on-write,Page replacement, Allocation of frames, Thrashing Storage Management: Overview of Mass Storage Structure, HDD Scheduling.

UNIT - V File System:

File System Interface: File concept, Access methods, Directory Structure; File system Implementation: File-system structure, File-system Operations, Directory implementation, Allocation method, Free space management; File-System Internals: File-System Mounting, Partitions and Mounting, File Sharing. Protection: Goals of protection, Principles of protection, Protection Rings, Domain of protection, Access matrix.

Text Books:

1. Operating System Concepts, Silberschatz A, Galvin P B, Gagne G, 10th Edition, Wiley, 2018.
2. Modern Operating Systems, Tanenbaum A S, 4th Edition, Pearson , 2016.

Reference Books:

1. Operating Systems -Internals and Design Principles, Stallings W, 9th edition, Pearson, 2018
2. Operating Systems: A Concept Based Approach, D.M Dhamdhere, 3rd Edition, McGraw-Hill, 2013

Online Learning Resources:

1. https://nptel.ac.in/courses/106/106/106106144/
2. http://peterindia.net/OperatingSystems.html

Open Elective I	Computer Organization and Architecture	L	T	P	C
		3	0	0	3

Course Objectives:
The main objectives of the course is to
- provide students with a comprehensive understanding of digital logicdesign principles and computer organization fundamentals
- Describe memory hierarchy concepts
- Explain input/output (I/O) systems and their interaction with theCPU, memory, and peripheral devices

UNIT – I:
Data Representation: Binary Numbers, Fixed PointRepresentation. FloatingPoint Representation. Number base conversions, Octal and Hexadecimal Numbers, components, Signed binary numbers, Binary codes
Digital Logic Circuits-I: Basic Logic Functions, Logic gates, universal logic gates, Minimization of Logic expressions. K-Map Simplification, Combinational Circuits, Decoders, Multiplexers
UNIT – II:
Digital Logic Circuits-II: Sequential Circuits, Flip-Flops, Binary counters, Registers, Shift Registers, Ripple counters
Basic Structure of Computers: Computer Types, Functional units, Basic operational concepts, Bus structures, Software, Performance,multiprocessors and multi computers, Computer Generations, Von-Neumann Architecture
UNIT – III:
Computer Arithmetic: Addition and Subtraction of Signed Numbers, Designof Fast Adders, Multiplication of Positive Numbers, Signed-operand Multiplication, Fast Multiplication, Integer Division, Floating-Point Numbers and Operations
Processor Organization: Fundamental Concepts, Execution of a Complete Instruction, Multiple-Bus Organization, Hardwired Control and Multi programmed Control
UNIT – IV:
The Memory Organization: Basic Concepts, Semiconductor RAM Memories, Read-Only Memories, Speed, Size and Cost, Cache Memories, Performance Considerations, Virtual Memories, Memory Management Requirements, Secondary Storage
UNIT – V:
Input/Output Organization: Accessing I/O Devices, Interrupts, Processor Examples, Direct Memory Access, Buses, Interface Circuits, Standard I/O Interfaces

Textbooks:
1. Computer Organization, Carl Hamacher, Zvonko Vranesic, Safwat Zaky, 6th edition, McGraw Hill
2. Digital Design, 6th Edition, M. Morris Mano, Pearson Education.
3. Computer Organization and Architecture, William Stallings, 11thEdition, Pearson.

Reference Books:
1. Computer Systems Architecture, M.Moris Mano, 3rdEdition, Pearson
2. Computer Organization and Design, David A. Paterson, John L. Hennessy, Elsevier
3. Fundamentals of Logic Design, Roth, 5thEdition, Thomson

Online Learning Resources:
1. https://nptel.ac.in/courses/106/103/106103068/

Open Elective II:	Principles of Database Management Systems	L	T	P	C
		3	0	0	3

Course Objectives:

1. Introduce database management systems and the formal foundation of the relational model.
2. Explain the concepts and usage of basic SQL as a universal database language.
3. Demonstrate systematic database design through conceptual modeling and normalization.
4. Provide an overview of physical database design, including indexing and storage techniques.
5. Understand query optimization, transaction management, and database system performance considerations.

UNIT I:

Introduction: Database system, Characteristics (Database Vs File System), Database Users, Advantages of Database systems, Database applications. Brief introduction of different Data Models; Concepts of Schema, Instance and data independence; Three tier schema architecture for data independence; Database system structure, environment, Centralized and Client Server architecture for the database.

Entity Relationship Model: Introduction, Representation of entities, attributes, entity set, relationship, relationship set, constraints, sub classes, super class, inheritance, specialization, generalization using ER Diagrams.

UNIT II:

Relational Model: Introduction to relational model, concepts of domain, attribute, tuple, relation, importance of null values, constraints (Domain, Keyconstraints, integrity constraints) and their importance, Relational Algebra, Relational Calculus. BASIC SQL: Simple Database schema, data types, table definitions (create, alter), different DML operations (insert, delete, update).

UNIT III:

SQL: Basic SQL querying (select and project) using where clause, arithmetic & logical operations, SQL functions(Date and Time, Numeric, String conversion).Creating tables with relationship, implementation of key and integrity constraints, nested queries, sub queries, grouping, aggregation, ordering, implementation of different types of joins, view (updatable and non- updatable), relational set operations.

UNIT IV:

Schema Refinement (Normalization): Purpose of Normalization or schema refinement, concept of functional dependency, normal forms based on functional dependency Lossless join and dependency preservingdecomposition, (1NF, 2NF and 3 NF), concept of surrogate key, Boyce-Codd normal form(BCNF), MVD, Fourth normal form(4NF), Fifth Normal Form (5NF).

UNIT V:

Transaction Concept: Transaction State, ACID properties, Concurrent Executions, Serializability, Recoverability, Implementation of Isolation, Testing for Serializability, lock based, time stamp based, optimistic, concurrency protocols, Deadlocks, Failure Classification, Storage, Recovery and Atomicity, Recovery algorithm. Introduction to Indexing Techniques: B+ Trees, operations on B+Trees, Hash Based Indexing.

Course Outcomes :

1. Demonstrate understanding of database systems, architectures, and ER modeling.
2. Apply relational model concepts, constraints, and basic SQL operations.
3. Perform advanced SQL queries using joins, subqueries, aggregation, and views.
4. Implement normalization techniques to design efficient and consistent database schemas.

5. Analyze transaction management, concurrency control, recovery, and indexing mechanisms.

Text Books:

1) Database Management Systems, 3rd edition, Raghurama Krishnan, Johannes Gehrke, TMH (For Chapters 2, 3, 4)
2) Database System Concepts,5th edition, Silberschatz, Korth, Sudarsan, TMH (For Chapter 1 and Chapter 5)

Reference Books:

1) Introduction to Database Systems, 8thedition, C J Date, Pearson.
2) Database Management System, 6th edition, Ramez Elmasri, Shamkant B. Navathe, Pearson
3) Database Principles Fundamentals of Design Implementation and Management, Corlos Coronel, Steven Morris, Peter Robb, Cengage Learning.

Web-Resources:

1) https://nptel.ac.in/courses/106/105/106105175/
2) https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_01275806667282022456_shared/overview

Open Elective III:	Object Oriented Programming Through Java	L	T	P	C
		3	0	0	3

Course Objectives:

1. identify Java language components and how they work together in applications
2. Learn the fundamentals of object-oriented programming in Java, including defining classes, invoking methods, using class libraries.
3. learn how to extend Java classes with inheritance and dynamic binding and how to use exception handling in Java applications
4. understand how to design applications with threads in Java • understand how to use Java APIs for program development

UNIT I

Object Oriented Programming: Basic concepts, Principles, Program Structure in Java: Introduction, Writing Simple Java Programs, Elements or Tokens in Java Programs, Java Statements, Command Line Arguments, User Input to Programs, Escape Sequences Comments, Programming Style. Data Types, Variables, and Operators: Introduction, Data Types in Java, Declaration of Variables, Data Types, Type Casting, Scope of Variable Identifier, Literal Constants, Symbolic Constants, Formatted Output with printf() Method, Static Variables and Methods, Attribute Final, Introduction to Operators, Precedence and Associativity of Operators, Assignment Operator (=), Basic Arithmetic Operators, Increment (++) and Decrement (-) Operators, Ternary Operator, Relational Operators, Boolean Logical Operators, Bitwise Logical Operators.

Control Statements: Introduction, if Expression, Nested if Expressions, if– else Expressions, Ternary Operator? Switch Statement, Iteration Statements, while Expression, do–while Loop, for Loop, Nested for Loop, for– Each for Loop, Break Statement, Continue Statement.

UNIT II

Classes and Objects: Introduction, Class Declaration and Modifiers, Class Members, Declaration of Class Objects, Assigning One Object to Another, Access Control for Class Members, Accessing Private Members of Class, Constructor Methods for Class, Overloaded Constructor Methods, Nested Classes, Final Class and Methods, Passing Arguments by Value and by Reference, Keyword this.

Methods: Introduction, Defining Methods, Overloaded Methods, Overloaded Constructor Methods, Class Objects as Parameters in Methods, Access Control, Recursive Methods, Nesting of Methods, Overriding Methods, Attributes Final and Static.

UNIT III

Arrays: Introduction, Declaration and Initialization of Arrays, Storage of Array in Computer Memory, Accessing Elements of Arrays, Operations on Array Elements, Assigning Array to Another Array, Dynamic Change of Array Size, Sorting of Arrays, Search for Values in Arrays, Class Arrays, Two-dimensional Arrays, Arrays of Varying Lengths, Three-dimensional Arrays, Arrays as Vectors**.**

Inheritance: Introduction, Process of Inheritance, Types of Inheritances, Universal Super Class-Object Class, Inhibiting Inheritance of Class Using Final, Access Control and Inheritance, Multilevel Inheritance, Application of Keyword Super, Constructor Method and Inheritance, Method Overriding, Dynamic Method Dispatch, Abstract Classes, Interfaces and Inheritance. **Interfaces:** Introduction, Declaration of Interface, Implementation of Interface, Multiple Interfaces, Nested Interfaces, Inheritance of Interfaces, Default Methods in Interfaces, Static Methods in Interface, Functional Interfaces, Annotations.

UNIT IV

Packages and Java Library: Introduction, Defining Package, Importing Packages and Classes into Programs, Path and Class Path, Access Control, Packages in Java SE, Java.lang Package and its Classes, Class Object, Enumeration, class Math, Wrapper Classes, Auto-boxing and Auto-unboxing, Java util Classes and Interfaces, Formatter Class, Random Class, Time Package, Class Instant (java.time.Instant), Formatting for Date/Time in Java, Temporal Adjusters Class, Temporal Adjusters Class.

Exception Handling: Introduction, Hierarchy of Standard Exception Classes, Keywords throws and throw, try, catch, and finally Blocks, Multiple Catch Clauses, Class Throwable, Unchecked Exceptions, Checked Exceptions.
Java I/O and File: Java I/O API, standard I/O streams, types, Byte streams, Character streams, Scanner class, Files in Java (Text Book 2)

UNIT V

String Handling in Java: Introduction, Interface Char Sequence, Class String, Methods for Extracting Characters from Strings, Comparison, Modifying, Searching; Class String Buffer.
Multithreaded Programming: Introduction, Need for Multiple Threads Multithreaded Programming for Multi-core Processor, Thread Class, Main Thread-Creation of New Threads, Thread States, Thread PrioritySynchronization, Deadlock and Race Situations, Inter-thread Communication - Suspending, Resuming, and Stopping of Threads.
Java FX GUI: Java FX Scene Builder, Java FX App Window Structure, displaying text and image, event handling, laying out nodes in scene graph, mouse events (Text Book 3)

Text Books:

1) JAVA one step ahead, Anitha Seth, B.L.Juneja, Oxford.
2) Joy with JAVA, Fundamentals of Object-Oriented Programming, Debasis Samanta, Monalisa Sarma, Cambridge, 2023.
3) JAVA 9 for Programmers , Paul Deitel , Harvey Deitel , 4 th Edition, Pearson.

References Books:

1) The complete Reference Java, 11th edition, Herbert Schildt, TMH
2) Introduction to Java programming, 7th Edition, Y Daniel Liang, Pearson

Online Resources:

1) https://nptel.ac.in/courses/106/105/106105191/
2) https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_012880464547618816347_shared/overview

Open Elective IV
Principles of Software Engineering /Computer Networks

Open Elective IV:	Principles of Software Engineering	L	T	P	C
		3	0	0	3

Course Objectives:
The objectives of this course are to introduce

1. Software life cycle models, Software requirements and SRS document.
2. Project Planning, quality control and ensuring good quality software.
3. Software Testing strategies, use of CASE tools, Implementation issues, validation & verification procedures.

UNIT I:
Introduction: Evolution, Software development projects, Exploratory style of software developments, Emergence of software engineering, Notable changes in software development practices, Computer system engineering.
Software Life Cycle Models: Basic concepts, Waterfall model and its extensions, Rapid application development, Agile development model, Spiral model.
UNIT II:
Software Project Management: Software project management complexities, Responsibilities of a software project manager, Metrics for project size estimation, Project estimation techniques, Empirical Estimation techniques, COCOMO, Halstead's software science, risk management. **Requirements Analysis and Specification:** Requirements gathering and analysis, Software Requirements Specification (SRS), Formal system specification, Axiomatic specification, Algebraic specification, Executable specification and 4GL.
UNIT III:
Software Design: Overview of the design process, how to characterize a good software design? Layered arrangement of modules, Cohesion and Coupling. Approaches to software design. **Agility:** Agility and the Cost of Change, Agile Process, Extreme Programming (XP), Other Agile Process Models, Tool Set for the Agile Process (Text Book 2)
Function-Oriented Software Design: Overview of SA/SD methodology, structured analysis, developing the DFD model of a system, Structured design, Detailed design, and Design Review. **User Interface Design:** Characteristics of a good user interface, Basic concepts, Types of user interfaces, Fundamentals of component-based GUI development, and user interface design methodology.

UNIT IV:
Coding And Testing: Coding, Code review, Software documentation, Testing, Black-box testing, White-Box testing, Debugging, Program analysis tools, Integration testing, Testing object-oriented programs, Smoke testing, and Some general issues associated with testing.
Software Reliability and Quality Management: Software reliability. Statistical testing, Software quality, Software quality management system, ISO 9000. SEI Capability maturity model. Few other important quality standards and Six Sigma.
UNIT V:
Computer-Aided Software Engineering (Case): CASE and its scope, CASE environment, CASE support in the software life cycle, other characteristics of CASE tools, Towards second generation CASE Tool, and Architecture of a CASE Environment.
Software Maintenance: Characteristics of software maintenance, Software reverse engineering, and Software maintenance process models and Estimation of maintenance cost.
Software Reuse: reuse- definition, introduction, reason behind no reuse so far, Basic issues in any reuse program, A reuse approach, and Reuse at organization level.
Text Books:

1. Fundamentals of Software Engineering, Rajib Mall, 5th Edition, PHI.
2. Software Engineering A practitioner's Approach, Roger S. Pressman, 9th Edition, Mc-Graw Hill International Edition.

Reference Books:

1. Software Engineering, Ian Sommerville,10th Edition, Pearson.
2. Software Engineering, Principles and Practices, Deepak Jain, Oxford University Press.

e-Resources:

1) https://nptel.ac.in/courses/106/105/106105182/
2) https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_01260589506387148827_shared/overview
3) https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_01338269041100390473 5_shared/overview

Open ElectiveIV:	Computer Networks	L	T	P	C
		3	0	0	3

Course Objective:

1. To educate basic knowledge of networking technologies and network management concepts
2. To interpret the layering concepts in computer networks.
3. To analyze the functions of each layer and gain knowledge in different applications that use computer networks.
4. To emphasize the hand-on experience of network topology in a laboratory environment
5. To be familiar with contemporary issues in networking technologies

UNIT – I: Introduction: Network Topologies WAN, LAN, MAN. Reference models-The OSI Reference Model- the TCP/IP Reference Model - A Comparison of the OSI and TCP/IP Reference Models Examples of Networks: Novell Networks, Arpanet, Internet.

UNIT – II: Physical Layer and overview of Physical Layer Switching:
Transmission Modes-Transmission media (Guided and Unguided Media). Multiplexing: frequency division multiplexing, wave length division multiplexing, synchronous time division multiplexing, statistical time division multiplexing, introduction to switching: Circuit Switched Networks, Datagram Networks, Virtual Circuit Networks

UNIT-III: Data link layer: Design issues: Framing, Flow control, Error control, CRC, Checksum: idea, one's complement internet checksum, services provided to Network Layer, Elementary Data Link Layer protocols: simplex protocol, Simplex stop and wait, Simplex protocol for Noisy Channel. Sliding window protocol: One bit, Go back N, Selective repeat-Stop and wait protocol, Data link layer in HDLC: configuration and transfer modes, frames, control field.

UNIT-IV: Random Access: ALOHA, MAC addresses, Carrier sense multiple access (CSMA), CSMA /CD, CSMA/CA, Controlled Access: Reservation, Polling, Token Passing, Channelization: frequency division multiple access(FDMA), time division multiple access(TDMA), code division multiple access(CDMA).
Network Layer: IP Addresses – Ipv4&IPv6 – Internetworking, Routing algorithm shortest path routing, Flooding, Hierarchical routing, Broadcast, Multicast, distance vector routing. IEEE Standards: – Standard Ethernet: MAC sub layer, physical layer, Fast Ethernet: MAC sub layer, physical layer, IEEE-802.11: Architecture, MAC sub layer, addressing mechanism, frame structure.

UNIT-V: Transport Layer & Application Layer: Process to Process Delivery -User Datagram - Protocol (UDP) - Transmission Control Protocol (TCP) - Congestion Control- Quality of services (QOS) - Integrated Services - Domain Name Space (DNS) - FTP – HTTP- WWW & HTTP.

Course Outcomes:

1. Understand various network topologies and compare OSI and TCP/IP reference models.
2. Learn about physical layer transmission media, multiplexing techniques, and switching methods.
3. Master data link layer protocols, including error control, flow control, and HDLC.
4. Understand random and controlled access techniques, including CSMA, FDMA, TDMA, and CDMA.
5. Explore transport and application layer protocols such as TCP/UDP, congestion

control, and HTTP.

TEXT BOOKS:

1. Tanenbaum and David J Wetherall, Computer Networks, 5th Edition, Pearson Edu, 2010
2. Computer Networks: A Top Down Approach, Behrouz A. Forouzan, Firouz Mosharraf, McGraw Hill Education

REFERENCE BOOKS:

1. Larry L. Peterson and Bruce S. Davie, “Computer Networks - A Systems Approach” (5th edition)
2.Computer Networks, Mayank Dave, CENGAGE Understanding communications and Networks, 3rd Edition, W.A. Shay, Thomson

HONORS COURSES

TRACK1: WEB FRAMEWORK

2-2 Angular JS Framework
3-1 FLASK Framework
3-2 Dot Net Core
(3-2 Any Course from SWAYAM NPTEL Suggested by BoS in this track.)
4-1 Java Enterprise Framework

TRACK2: NETWORKS

2-2. Introduction to Networks (ITN)
3-1. Switching, Routing, and Wireless Essentials
3-2. Enterprise Networking, Security, and Automation
(3-2 Any Course from SWAYAM NPTEL Suggested by BoS in this track.)
4-1. Wireless Sensor Networks.

TRACK3: SECURITY

1) 2-2. **Cyber Laws and Security Policies**
2) 3-1. Secure Coding
3) 3-2. Vulnerability Assessment & Penetration Testing
4) Privacy and Security in Online Social Media(THROUGH MOOCS)
5) 4-1. Malware Analysis

TRACK4: ARTIFICIAL INTELLIGENCE

1) 2-2. AI for Problem Solving.
2) 3-1. Social Network Analysis
3) 3-2. Deep Learning for Computer Vision
4) Bio Informatics/ GPU Architecture and Programming (Through MOOCs)
5) 4-1. AI in Health Care

Subject Code	Subject Name	L	T	P	C
HONORS	Angular JS Framework	3	0	0	3

Course Objectives:

1. To understand model view framework for building applications.
2. To create modules for binding the application.
3. To understand dependency injection for implementing services.
4. To create and establish routes redirects and navigation.
5. To validate forms for the submission of data.

Unit I

Angular JS – Introduction to Angular JS, Java Script vs Angular, MVC Framework, Component Based Model, Setting Up the Environment, Installation of Node and NPM, Angular CLI, Creating and Running Project, Add Dependencies, The Anatomy of an AngularJS app, First Application.What is a Component, Create and Start Component.

Unit II

Data Binding: Introduction to Data Binding, Types of Binding, Binding Data from Component, Async, Template Interpolation, Looping with ngFor, Condition with ngif, Passing inputs and variables to Components, ngModel for 2-way binding, ngOnInit, Styling with components, Creating multiple modules. Combine Forms with Data binding.

Unit III

Dependency Injection: Understanding Dependency Injection(DI), Services, Creating a Service, Service Injection Context, Rest Calls with HttpClient, Building Angular Project.

Unit IV

Routing & Wrap Up: Introduction to Routing, Angular Project with routing, Creating routes, Route redirects and wild cards, Route Configuration, Static Data in Route, Nested Routing, Navigation Controlling.

Unit V

Form Handling: Introduction to Form Handling, Template Driven, Form Validation, ng-minlength, ngmaxlength, ng-pattern, ng-required, Submitting Forms, Event Handling with Forms.

APPLICATIONS:

- Online Web Applications
- Financial, Banking Applications and Gateways etc
- Online and Social Media Applications

Course Outcomes:

1. Understand the fundamentals of Angular JS and its architecture.
2. Apply data binding objects for implementing modules.
3. Implement service and retrieve rest call data.
4. Understand routes and their configuration in angular.
5. Implement form handling with event driven apps.

TEXT BOOKS:

1. Angular 6 by Example: Get up and running with Angular by building modern realworld web apps, 3rd Edition, by Chandermani Arora.
2. Pro Angular 6, Apress, by Adam Freeman

REFERENCE BOOKS:

1. Angular JS by Green, Orielly
2. Professional AngularJS (WROX), by Valeri Karpov

Subject Code	Subject Name	L	T	P	C
HONORS	FLASK Framework	3	0	0	3

Prerequisite: Knowledge of Python and Web Technologies

Course Objectives:

1. Understand the fundamentals of web development using Flask.
2. Develop dynamic web applications with routing, templates, and forms.
3. Integrate databases with Flask using ORM (SQLAlchemy).
4. Implement RESTful APIs and apply authentication techniques.
5. Deploy Flask applications on cloud or local servers.

UNIT I: Introduction to Flask & Web Basics

Introduction to Web Frameworks and Flask,FlaskvsDjango,Installing Flask and setting up the environment,Understanding WSGI and Flask Application Structure,First Flask App: routing, URL building,Debugging and reloading

UNIT II: Templates and Static Files

Introduction to Jinja2 Templates,Passing data to templates,Template inheritance and macros,Working with static files (CSS, JS, Images),Creating reusable layouts and components

UNIT III: Forms and User Input

Handling GET and POST requests,Flask-WTF and form validation,Working with Flask sessions and cookies,Redirects, flash messages,File upload and handling

UNIT IV: Database Integration

Introduction to SQLAlchemyORM,Connecting to SQLite/MySQL/PostgreSQL,Creating models and relationships,Performing CRUD operations,Flask-Migrate for schema migrations

UNIT V: RESTful APIs and Deployment

Creating RESTful APIs with Flask,Flask-RESTfulextension,Implementing authentication (basic and token-based),Unit testing Flask applications,Deployment on Heroku/Render/Docker

Course Outcomes:

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

1. Build web applications using the Flask framework.
2. Design dynamic user interfaces using Jinja2.
3. Integrate Flask apps with databases using ORM.
4. Create and consume REST APIs.
5. Deploy Flask applications on real-world platforms.

Text Books:

1. Miguel Grinberg, “Flask Web Development”, O’Reilly Media, 2nd Edition, 2018.
2. Ronan Schwarz, "Learning Flask Framework", Packt Publishing.

Reference Books:

1. Gareth Dwyer, “Mastering Flask”, Packt Publishing.
2. Charles Severance, “Python for Everybody**”**, Free online version.

Subject Code	Subject Name	L	T	P	C
HONORS	DOT NET Core	3	0	0	3

Course Objectives:

1. Understand Visual Studio Code and .NET Core development tools for an efficient coding environment.
2. Apply OOPS principles effectively within .NET Core applications.
3. Gain practical expertise in manipulating data using collections, LINQ, and generics.
4. Design and deploy secure and responsive web applications using ASP.NET Core and front-end frameworks.
5. Implement secure and efficient web applications with effective state management and persistence strategies in .NET Core.

Unit I: The .NET Core Technology & Introduction to C#:
Fundamentals of the .NET Core framework and its architecture,Overview of .NET Core vs .NET Framework,Common Type System (CTS), Common Language Specification (CLS),Base Class Library (BCL) and Common Language Runtime (CLR),C# features and basics: Program structure, data types, operators, decision-making statements, loops, arrays, and strings

Unit II: OOP Concepts and Asynchronous Programming: Class, object, inheritance, abstract classes, interfaces, polymorphism, exception handling,Collections in C# and LINQ,

Unit III: ASP.NET Core Web Development: Overview of ASP.NET Core framework,ASP.NET Core page lifecycle, Razor Pages, MVC pattern.Introduction to Tag Helpers, HTML Helpers, Validation, Integrating modern front-end frameworks (React or Angular or Vue) with ASP.NET Core

Unit IV: ASP.NET Core Session and State Management: Overview of session state, its importance, and the challenges of maintaining user state in web applications, Client-side and server-side session management

Unit V: Working with Databases in .NET Core (Entity Framework Core): Overview of Entity Framework Core, Connected and Disconnected Architecture, Database connectivity, Integration with cloud databases and services (e.g., Azure SQL, AWS RDS).

Course Outcomes:

1. Demonstrate proficiency in C# syntax, structure, and .NET Core development standards.
2. Apply object-oriented and asynchronous programming to develop scalable .NET Core applications.
3. Implement data manipulation using collections, LINQ, and generics effectively.
4. Design and develop secure, dynamic web applications using ASP.NET Core and front-end tools.
5. Build secure, database-integrated web applications with proper state management.

Text Books

1. Pro ASP.NET Core 3, by Adam Freeman, Apress, 2019, 1st Edition
2. C# 8.0 and .NET Core 3.0 – Modern Cross-Platform Development, by Mark J. Price, Packt Publishing, 2019, 4th Edition
3. Entity Framework Core in Action, by Jon P Smith, Manning Publications, 2018, 1st Edition

Reference Books

1. ASP.NET Core in Action by Andrew Lock, Manning Publications, 2018, 1st Edition
2. Programming ASP.NET Core, by Dino Esposito, Microsoft Press, 2018, 1st Edition
3. Pro C# 8 with .NET Core: Foundational Principles and Practices in Programming, by Andrew Troelsen, Philip Japikse, Apress, 2019, 9th Edition

Subject Code	Subject Name	L	T	P	C
HONORS	Java Enterprise Framework	3	0	0	3

Course Objectives:

1. Implement CRUD operations using Java APIs
2. Understand server side environment using Servlets.
3. Create server side web pages using Java Server Pages.
4. Understand Object relational mapping using Hibernate.
5. Write business logics using Spring MVC and AOP programming.

UnitI:

Java Database Connectivity - JDBC Overview & Architecture, JDBC Driver Types, TypesofResultSet, Statement, PreparedStatement, CallableStatement, Executing DDL and DMLCommands.

Unit-II:

Servlets: Web Server, Container, Servlet Overview, Life cycle of Servlet, Handling WebForm Data in Servlets., Dynamically including Content in Servlets, Handling Exceptions inWeb Applications, Reading and Setting Cookies, Session Tracking, Servlet Filters, AccessingDatabases.

Unit-III:

Java Server Pages:

Overview of JSP, JSP Architecture& lifecycle, Components of Java Server Pages, Implicit Objects & Standard JSP Tags, Scope of JSP objects, Dynamically including content in JSPs, Handling Form data in JSPs, Accessing Databases, Tag Libraries.

Unit-IV:

HIBERNATE: Introduction to Hibernate, Hibernate Architecture, Understanding Object Persistence, Hibernate Basics, Types of Relations, Querying Persistent Objects, Hibernate Query Language (HQL)

Unit -V:

Spring Framework: Spring Architecture and Container, Spring Setup in Eclispe, Spring Bean Factory, Bean factory VS Application Context, Dependency Injection (DI), Types of DI, Bean Auto wiring,CollectionswithSpring,BeanScopes,EventHandlinginSpring,Introductionto Aspect Oriented Programming(AOP).

Course Outcomes:

1. Implement data base operations from frontend java APIs.
2. Write server side programs for controlling HTTP requests.
3. Create dynamic server side webpages using JSP tags.
4. Create Hibernate objects form aping objects and database relations.

Create model based java objects for controlling server based applications

Text Books:

1. JDBC,ServletsandJSPBlackBook,byKogentSolutionsInc.Santosh KumarK,Dreamtech Publications. 2nd Edition, 2016.
2. Java Servlet &JSP Cook book by Bruce W. Perry Publisher(s), O'Reilly Media, 2004
3. SpringandHibernate-2ed,byK.SantoshKumar, McGraw-Hill Education

Reference Books:

1. SpringinAction,4ed(Manning),by Craig Walls(Author),Dream tech.
2. JavaPersistencewithHibernate:RevisedofHibernateinAction,Dreamtech,by ChristianBauer.

Course Code	Subject Name	L	T	P	C
HONORS	Introduction to Networks (ITN)	3	0	0	3

Course Objectives

1. Understand core network components including devices, protocols, and transmission media.
2. Learn IPv4 addressing, subnetting, and perform subnet calculations for network segmentation.
3. Identify different cable types and connectors used in LAN environments.
4. Gain practical skills in setting up and troubleshooting basic network connections across multiple operating systems.
5. Understand the role of network switches in connecting devices and forwarding data within a network.

Unit-I – Standards and Concepts

Identify the fundamental conceptual building blocks of networks - Differentiate between bandwidth and throughput - Differentiate between LAN, WAN, MAN, CAN, PAN, and WLAN - Compare and contrast cloud and on-premises applications and services - Describe common network applications and protocols.

Unit – II: Addressing and Subnet Formats
Compare and contrast private addresses and public addresses - Identify IPv4 addresses and subnet formats - Identify IPv6 addresses and prefix formats.

Unit – III: Endpoints and Media Types
Identify cables and connectors commonly used in local area network - Differentiate between Wi-Fi, cellular, and wired network technologies - Describe endpoint devices - Demonstrate how to set up and check network connectivity on Windows, Linux, Mac OS, Android, and Apple iOS.

Unit – IV: Infrastructure
Identify the status lights on a Cisco device when given instruction by an engineer - Use a network diagram provided by an engineer to attach the appropriate cables - . Identify the various ports on network devices - Explain basic routing concepts - Explain basic switching concepts.

Unit – V: Diagnosing Problems
Demonstrate effective troubleshooting methodologies and help desk best practices, including ticketing, documentation, and information gathering - Perform a packet capture with Wire shark and save it to a file - Run basic diagnostic commands and interpret the results - Differentiate between different ways to access and collect data about network devices - Run basic show commands on a Cisco network device.

Security: Describe how firewalls operate to filter traffic - Describe foundational security concepts - Configure basic wireless security on a home router (WPAx).

Text Books:
1. Introduction to Networks Companion Guide (CCNAv7) Cisco Networking Academy,
2. ISBN-13: 978-0-13-663366-2 ISBN-10: 0-13-663366-8

References Books:
1. CCNA : Cisco Certified Network Associate study guide / Todd Lammle. — 7th ed. Wiley Publishing, Inc.ISBN 978-0-470-90107-6

Weblinks:

1. https://www.ccri.edu/faculty_staff/comp/jmowry/CSCO-1850-PP.html
2. https://examscisco.com/ccna-v7-0/ccna-1-v7-introduction-to-networks-v7-02-itn-exam-answers/

Course Code	Subject Name	L	T	P	C
HONORS	Switching, Routing, and WirelessEssentials	3	0	0	3

Course Objectives

1. Understand Layer 2 switching, OSI model operations, and configure basic VLANs.
2. Configure redundancy mechanisms using Spanning Tree Protocol (STP) and EtherChannel.
3. Set up basic DHCPv4 servers and explain SLAAC, DHCPv6, and FHRPs for network reliability.
4. Configure basic switch security features and implement secure Wireless LANs.
5. Understand routing principles and configure basic static routes on network routers.

Unit-I

Switching Concepts and VLANS: Basic Device Configuration - Switching Concepts – VLANs - Inter-VLAN Routing

Unit – II:

Redundant Networks: STP - Etherchannel

Unit – III:

Available and Reliable Networks :DHCPv4 - SLAAC and DHCPv6 Concepts - FHRP Concepts

Unit – IV:

L2 Security and WLANs :LAN Security Concepts - Switch Security Configuration - WLAN Concepts - WLAN Configuration

Unit – V:

Routing Concepts and Configuration: Routing Concepts - IP Static Routing - Troubleshoot Static and Default Routes

Course Outcomes:

1. Configuring basic VLANs on a network switch using appropriate methods (port-based, MAC-based, etc.)
2. Understanding the concepts and functionalities of Spanning Tree Protocol (STP) in preventing loops on switched networks
3. Configuring a basic DHCPv4 server to automatically assign IP addresses to network devices
4. Applying security measures to mitigate network vulnerabilities and protect sensitive data
5. Applying static routes to connect different networks and ensure proper traffic flow

Text Books:

1. Switching, Routing, and Wireless Essentials Companion Guide, Cisco Networking Academy Published by: Cisco Press Hoboken, New Jersey.ISBN-13: 978-0-13-672935-8 ISBN-10: 0-13-672935-5

References Books:

1. CCNA: Cisco Certified Network Associate study guide / Todd Lammle. — 7th ed. Wiley Publishing, Inc.ISBN 978-0-470-90107-6

Weblinks

- https://itexamanswers.net/ccna-2-v7-exam-answers-switching-routing-and-wireless-essentials-v7-0-srwe.html
- https://examscisco.com/ccna-v7-0/ccna-2-v7-switching-routing-and-wireless-essentials-v7-02-srwe-exam-answers/#google_vignette

Course Code	Subject Name	L	T	P	C
HONORS	Enterprise Networking Security and Automation	3	0	0	3

Course Objectives

1. Configure a single-area OSPFv2 network using a network simulator or real equipment.
2. Understand the principles of Network Address Translation (NAT) for IPv4 and its role in network security and addressing.
3. Analyze different VPN technologies like PPTP, L2TP/IPsec, and OpenVPN
4. Apply network troubleshooting methodologies to identify and resolve common network issues.
5. Understand the principles of network automation and its role in automating network configuration and management tasks.

Unit-I

OSPF Concepts and Configuration: Single-Area OSPFv2 Concepts - Single-Area OSPFv2 Configuration

Unit – II:

Network Security: Network Security Concepts - ACLs Concepts - ACLS for IPv4 Configuration - NAT for IPv4

Unit – III:

WAN: WAN Concepts - VPN and IPsec Concepts

Unit – IV:

Optimize, Monitor, and Troubleshoot Networks: QoS Concepts - Network Management - Network Design - Network Troubleshooting**.**

Unit – V:

Network Virtualization and Automation: Network Virtualization - Network Automation

Course Outcomes:

1. Apply the different OSPFv2 components like areas, routers, neighbors, and LSAs
2. Understand the fundamental concepts of network security, including threats, vulnerabilities, and security controls.
3. Apply the various technologies used to connect geographically dispersed locations in a WAN
4. Apply network design principles to optimize WAN performance, scalability, and security
5. Apply the knowledge to design, implement, and manage automated network solutions.

Text Books:

1. Enterprise Networking, Security, and Automation Companion Guide (CCNAv7) Cisco Networking Academy ISBN-13: 978-0-13-663432-4ISBN-10: 0-13-663432-X

References Books:

1. A Practical Introduction To Enterprise Network And Security Management. [2 ed.] CRC Press. ISBN: 978-0-367-64251-8 (hbk), ISBN: 978-1-032-04802-4 (pbk), ISBN: 978-1-003-12369-9 (ebk)

Weblinks

1. https://ptgmedia.pearsoncmg.com/images/9780136634324/samplepages/9780136634324_Sample.pdf
2. https://www.oreilly.com/library/view/enterprise-networking-security/9780136634171/

Course Code	Subject Name	L	T	P	C
HONORS	Wireless Sensor Networks	3	0	0	3

Course Objectives

1. Define WSN and Dynamic modulation scaling.
2. Explore working of the MAC protocols
3. Demonstrate Routing and Data gathering protocols
4. Illustrate working of Embedded OS.
5. Explore a wide range of WSN applications in different sectors

Unit-I – CHARACTERISTICS OF WSN

Characteristic requirements for WSN – Challenges for WSNs – WSN vsAdhoc Networks – Sensor node architecture – Commercially available sensor nodes –Imote, IRIS, Mica Mote, EYES nodes, BTnodes, TelosB, Sunspot -Physical layer and transceiver design considerations in WSNs, Energy usage profile, Choice of modulation scheme, Dynamic modulation scaling, Antenna considerations.

Unit – II:MEDIUM ACCESS CONTROL PROTOCOLS

Fundamentals of MAC protocols – Low duty cycle protocols and wakeup concepts – Contention based protocols – Schedule-based protocols – SMAC – BMAC – Traffic adaptive medium access protocol (TRAMA) – The IEEE 802.15.4 MAC protocol.

Unit – III:ROUTING AND DATA GATHERING PROTOCOLS

Routing Challenges and Design Issues in Wireless Sensor Networks, Flooding and gossiping – Data centric Routing – SPIN – Directed Diffusion – Energy aware routing – Gradient-based routing – Rumor Routing – COUGAR – ACQUIRE – Hierarchical Routing – LEACH, PEGASIS – Location Based Routing – GF, GAF, GEAR, GPSR – Real Time routing Protocols – TEEN, APTEEN, SPEED, RAP – Data aggregation - data aggregation operations – Aggregate Queries in Sensor Networks – Aggregation Techniques – TAG, Tiny DB.

Unit – IV: EMBEDDED OPERATING SYSTEMS

Operating Systems for Wireless Sensor Networks – Introduction – Operating System Design Issues – Examples of Operating Systems – TinyOS – Mate – MagnetOS – MANTIS – OSPM – EYES OS – SenOS – EMERALDS – PicOS – Introduction to Tiny OS – NesC – Interfaces and Modules – Configurations and Wiring – Generic Components – Programming in Tiny OS using NesC, Emulator TOSSIM.

Unit – V: APPLICATIONS OF WSN

WSN Applications – Home Control – Building Automation – Industrial Automation – Medical Applications – Reconfigurable Sensor Networks – Highway Monitoring – Military Applications – Civil and Environmental Engineering Applications – Wildfire Instrumentation – Habitat Monitoring – Nanoscopic Sensor Applications – Case Study: IEEE 802.15.4 LR-WPANs Standard – Target detection and tracking – Contour/edge detection – Field sampling.

Course Outcomes:

1. Understand the basics, characteristics and challenges of Wireless Sensor Network
2. Apply the knowledge to identify appropriate physical and MAC layer protocol
3. Apply the knowledge to identify the suitable routing algorithm based on the network and user requirement
4. Analysis of OS used in Wireless Sensor Networks and build basic modules
5. Analyze specific WSN application using a case study approach

Text Books:

1. Wireless Sensor Networks Technology, Protocols, and Applications, KazemSohraby, Daniel Minoli and TaiebZnati, John Wiley & Sons, 2007
2. Protocols and Architectures for Wireless Sensor Network, Holger Karl and Andreas Willig John Wiley & Sons, Ltd ,2005

References Books:

1. A survey of routing protocols in wireless sensor networks , K. Akkaya and M. Younis, Elsevier
2. Ad Hoc Network Journal, Vol. 3, no. 3, pp. 325--349
3. TinyOSProgramming , Philip Levis
4. Wireless Sensor Network Designs , Anna Ha´c , John Wiley& Sons Ltd

Course Code	Subject Name	L	T	P	C
HONORS	Cyber Laws and Security Policies	3	0	0	3

COURSE OBJECTIVES:

1. To impart a comprehensive understanding of cyber laws and legal frameworks related to cybersecurity.
2. To explore national and international cybercrime laws and digital evidence handling.
3. To familiarize students with IT Act, intellectual property rights, and privacy issues in cyberspace.
4. To develop knowledge of security policies, standards, and compliance requirements.
5. To enable students to draft, implement, and audit effective organizational security policies.

UNIT – I:

Introduction to Cyber Law and Cybercrime:

Overview of cyberspace and cyber law. Definition and classification of cybercrimes – hacking, cyberstalking, phishing, identity theft, cyber terrorism. Causes and prevention of cybercrimes. Legal challenges in cyberspace. Need for cyber regulations and frameworks. Overview of computer ethics and cyber jurisprudence.

UNIT – II:

Indian Cyber Law – IT Act 2000 and Amendments:

Objectives and features of the Information Technology Act, 2000. Definitions under the Act – electronic records, digital signatures, cyber offences. Offences and penalties under IT Act. Adjudication process and appellate tribunal. Amendments in IT Act 2008 – cybercrime regulation, data protection provisions, and legal recognition of digital evidence.

UNIT – III:

Legal Aspects of E-Commerce and Intellectual Property Rights:

Legal recognition of electronic contracts and digital signatures. Cyber contracts and their enforceability. Consumer protection in e-commerce. Intellectual property issues – copyright, trademark, and patent in the digital environment. Software piracy, digital rights management, and licensing. Jurisdictional issues in online transactions.

UNIT – IV:

Cyber Security Policies and Governance:

Introduction to security policies – definition, purpose, and scope. Key elements of security policy – access control, data classification, incident response. Policy creation and enforcement. Security standards and frameworks – ISO/IEC 27001, NIST, COBIT. Compliance requirements – HIPAA, PCI-DSS, GDPR. Risk-based policy formulation and best practices.

UNIT – V:

Cyber Forensics and Legal Investigation:

Introduction to cyber forensics. Role of digital evidence in legal proceedings. Legal procedures for evidence collection, preservation, and analysis. Chain of custody and admissibility in court. Case studies on cybercrime investigation and legal proceedings. Roles of CERT-IN and law enforcement agencies in cybercrime management.

COURSE OUTCOMES:

1. Understand the scope and significance of cyber laws in the context of cyber security.
2. Analyze cybercrimes and the legal provisions applicable under the IT Act and other laws.
3. Apply intellectual property and legal concepts to digital content and e-commerce.
4. Formulate and evaluate cyber security policies based on international standards and regulations.
5. Demonstrate knowledge of legal procedures and forensic practices in cybercrime investigation.

TEXTBOOKS:

1. **K. Kumar** – *Cyber Laws: Intellectual Property & E-Commerce, Internet & E-Business*, LexisNexis, 5th Edition.

2. **Pavan Duggal** – *Cyber Law: The Indian Perspective*, Saakshar Law Publications, Latest Edition.
3. **Nandan Kamath** – *Law Relating to Computers, Internet and E-Commerce*, Universal Law Publishing Co., 2nd Edition.

REFERENCE BOOKS:

1. **Jonathan Rosenoer** – *CyberLaw: The Law of the Internet*, Springer, 1st Edition.
2. **Mark F. Grady & Francesco Parisi** – *The Law and Economics of Cybersecurity*, Cambridge University Press.
3. **Chris Reed & Rebecca Wong** – *Cyber Security and the Law: Regulating the Connected World*, Routledge, 2019.

Subject Code	Subject Name	L	T	P	C
HONORS	Secure Coding	3	0	0	3

Course Objectives:

1. To understand the security development process.
2. Knowledge of outline of the techniques for developing a secure application.
3. To handling dynamic memory management effectively.
4. Knowledge on stored procedures and XSS attacks.
5. Acquire knowledge on software architecture and design.

UNIT–I:

Introduction-Need for secure systems, Proactive security development process, Security principles to live by and threat modeling.

UNIT–II:

Secure Coding in C-Character strings- String manipulation errors, String Vulnerabilities and exploits Mitigation strategies for strings, Pointers, Mitigation strategies in pointer based vulnerabilities Buffer Overflow based vulnerabilities.

UNIT–III:

Secure Coding in C++ and Java-Dynamic memory management, Common errors in dynamic memory management, Memory managers, Double –free vulnerabilities, Integer security, Mitigation strategies.

UNIT–IV:

Database and Web Specific Input Issues-Quoting the Input, Use of stored procedures, Building SQL statements securely, XSS related attacks and remedies.

UNIT–V:

Software Security Engineering-Requirements engineering for secure software: Misuse and abuse cases, SQUARE process model Software security practices and knowledge for architecture and design.

Course Outcomes: At the end of the course, student will be able to

1. Analyze secure systems and various security principles.
2. Understand the development of process of software leads to secure coding practices
3. Apply Secure programs and various risk in the dynamic memory management.
4. Understand XSS related attacks and remedies
5. Understand various software architecture models.

Text Book:

1. Michael Howard, David LeBlanc, “Writing Secure Code”, Microsoft Press, 2nd Edition, 2003.

Reference Books:

1. Robert C. Seacord, “Secure Coding in C and C++”, Pearson Education, 2nd edition, 2013.
2. Julia H. Allen, Sean J. Barnum, Robert J. Ellison, Gary McGraw, Nancy R. Mead, “Software Security Engineering: A guide for Project Managers”, Addison-Wesley Professional, 2008.

Subject Code	Subject Name	L	T	P	C
HONORS	Vulnerability Assessment & Penetration Testing	3	0	0	3

Course Objectives:

1. To identify penetration testing process.
2. To identify the various information gathering and scanning procedures of security systems.
3. To identify various system hacking procedures.
4. To understand the impact of hacking in real time machines.
5. To understand the impact of hacking in wireless networks.

UNIT-I: Introduction:

Penetration Testing phases/Testing Process, types and Techniques, Blue/Red Teaming, Strategies of Testing, Non-Disclosure Agreement Checklist, Phases of hacking, Open-source/proprietary Pentest Methodologies.

UNIT –II: Information Gathering and Scanning:

Information gathering methodologies- Foot printing, Competitive Intelligence- DNS Enumerations- Social Engineering attacks, Port Scanning- Network Scanning- Vulnerability Scanning- NMAP scanning tool- OS Fingerprinting- Enumeration.

UNIT-III: System Hacking:

Password cracking techniques- Key loggers- Escalating privileges Hiding Files, Double Encoding, Steganography technologies and its Countermeasures. Active and passive sniffing- ARP Poisoning, MAC Flooding- SQL Injection - Error- based, Union-based, Time-based, Blind SQL, Out-of-band, Injection Prevention Techniques.

UNIT IV: Advanced System Hacking:

Broken Authentication, Sensitive Data Exposure, XML External Entities, Broken Access Code, XSS - Stored, Reflected, DOM Based.

UNIT V: **Wireless Pentest:**

Wi-Fi Authentication Modes, Bypassing WLAN Authentication, Types of Wireless Encryption, WLAN Encryption Flaws, AP Attack, Attacks on the WLAN Infrastructure, DoS-Layer1, Layer2, Layer 3, DDoS Attack, Client Disassociation, Wireless Hacking Methodology, Wireless Traffic Analysis**.**

Course Outcomes:

1. Understand Penetration testing process.
2. Understand information gathering methodologies.
3. Analyze various Vulnerabilities assessments.
4. Apply System Hacking Techniques in real time applications.
5. Understand Bypassing WLAN Authentication

Textbooks:

1. Kali Linux 2: Windows Penetration Testing, By Wolf Halton, Bo Weaver , June 2016 PacktPublishing

Reference Books:

1. Mastering Modern Web Penetration Testing By Prakhar Prasad,October 2016 PacktPublishing.
2. SQL Injection Attacks and Defense 1st Edition, by Justin Clarke-Salt, Syngress Publication

Subject Code	Subject Name	L	T	P	C
HONORS	Malware Analysis	3	0	0	3

Course Objectives:

1. To understand the types and behavior of malware and its impact on systems.
2. To explore static and dynamic techniques used to analyze malware.
3. To use tools and methods to dissect and understand malicious code.
4. To study malware evasion techniques and methods to detect them.
5. To understand the process of reverse engineering and reporting malware behavior.

UNIT I: Introduction to Malware

Definition of malware, types of malware – viruses, worms, Trojans, ransomware, spyware, adware, rootkits. Modes of infection, persistence techniques, and payloads.Malware trends, attack vectors and their impact on individuals and organizations.

UNIT II: Static Malware Analysis

Basic static analysis techniques, examining file properties, strings, hashes, and metadata. Identifying suspicious patterns and file structures. Use of tools like PEview, Dependency Walker, VirusTotal, and hashing utilities.

UNIT III: Dynamic Malware Analysis

Setting up isolated lab environments (VMs and sandboxes), monitoring file, registry, and process activity. Using tools like Procmon, Process Explorer, Wireshark, and Regshot. Observing network activity and behavioral characteristics of malware in runtime.

UNIT IV: Reverse Engineering and Evasion Techniques

Introduction to reverse engineering, disassemblers (IDA Pro, Ghidra), decompilers, debugging with OllyDbg or x64dbg.Analyzing obfuscated code. Understanding packing, encryption, anti-debugging, anti-VM, and code injection techniques used by modern malware.

UNIT V: Case Studies and Malware Reporting

Hands-on case studies on real malware samples (e.g., WannaCry, Zeus, Stuxnet). Creating malware reports, behavior flowcharts, and Indicators of Compromise (IOCs). Introduction to threat intelligence sharing platforms (e.g., MISP, VirusTotal Community).

.

Course Outcomes:

After completing this course, students will be able to:

1. Identify different categories of malware and their propagation mechanisms.
2. Perform basic static and dynamic analysis using malware analysis tools.
3. Analyze malicious executables, scripts, and macros.
4. Understand anti-analysis and evasion techniques used by malware.
5. Document and report findings from malware investigations.

Textbooks:

1. Michael Sikorski and Andrew Honig, *Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software*, No Starch Press.
2. Monnappa K A, *Learning Malware Analysis*, Packt Publishing.

Reference Books:

1. Alexey Kleymenov, *Mastering Malware Analysis*, Packt Publishing.
2. Michael Hale Ligh et al., *Malware Analyst's Cookbook and DVD*, Wiley.
3. Chris Sanders and Jason Smith, *Practical Packet Analysis*, No Starch Press.

HONORS	Artificial Intelligence for Problem Solving	L	T	P	C
		3	0	0	3

Course Objectives:

This course deals with

- Autonomous agent behavior in solving problems intelligently.
- Imagining the consequence of AI decision making system to identify the wide variety of search methods that agents can employ for problem solving.

Unit-I

AI Philosophy: Introduction: History, Can Machines think? Winograd Schema Challenge, Language and Thought, Wheels & Gears**, Philosophy**: Mind, Reasoning, Computation, Dartmouth Conference, The Chess Saga, Epiphenomena**State Space Search:** Depth First Iterative Deepening.

Unit-II

Heuristic Search: Best First Search, Hill Climbing, Solution Space, TSP, Escaping Local Optima, Stochastic Local Search. **Population Based Methods**: Genetic Algorithms, SAT, TSP, emergent Systems, Ant Colony Optimization

Unit-III

Finding Optimal Paths: Branch & Bound, A*, Admissibility of A*, Informed Heuristic Functions. **Space Saving Versions of A*:** Weighted A*, IDA*, RBFS, Monotone Condition, Sequence Alignment, DCFS, SMGS, Beam Stack Search

Unit-IV

Game Playing: Game Theory, Board Games and Game Trees, Algorithm Minimax, AlphaBeta and SSS*. **Automated Planning:** Domain Independent Planning, Blocks World, Forward &Backward Search, Goal Stack Planning, Plan Space Planning. **Problem Decomposition:** Means Ends Analysis, Algorithm Graphplan, Algorithm AO*

Unit-V

Rule Based Expert Systems: Production Systems, Inference Engine, Match-Resolve-Execute, Rete Net. **Deduction as Search:** Logic, Soundness, Completeness, First Order Logic, Forward Chaining, Backward Chaining**. Constraint Processing:** CSPs, Consistency Based Diagnosis, Algorithm Backtracking, Arc Consistency, Algorithm Forward Checking.

Course Outcome

Students will be able to

1. Understand the philosophy of AI and State Space Search for problem solving.
2. Explore the operations of Heuristic Search and Population Based Methods.
3. Apply various optimal search techniques and Space Saving Algorithms
4. Develop Game Playing Algorithms using planning and decomposition problems.
5. Develop systems using Rule Based Systems and Constraint Satisfaction.

Books and references

Text Book:

1.Deepak Khemani. A First Course in Artificial Intelligence, McGraw Hill Education (India), 2013.

Reference Books:

1.Stefan Edelkamp and Stefan Schroedl. Heuristic Search: Theory and Applications, Morgan Kaufmann, 2011.
2.John Haugeland, Artificial Intelligence: The Very Idea, A Bradford Book, The MIT Press, 1985.
3.Pamela McCorduck, Machines Who Think: A Personal Inquiry into the History and Prospects of Artificial Intelligence, A K Peters/CRC Press; 2 edition, 2004.
4.Zbigniew Michalewicz and David B. Fogel. How to Solve It: Modern Heuristics. Springer; 2nd edition, 2004.
5.Judea Pearl. Heuristics: Intelligent Search Strategies for Computer Problem Solving, Addison-Wesley, 1984.
6.Elaine Rich and Kevin Knight. Artificial Intelligence, Tata McGraw Hill, 1991.
7.Stuart Russell and Peter Norvig. Artificial Intelligence: A Modern Approach, 3rd Edition, Prentice Hall, 2009.

8.Eugene Charniak, Drew McDermott. Introduction to Artificial Intelligence, Addison-Wesley, 1985.
9.Patrick Henry Winston. Artificial Intelligence, Addison-Wesley, 1992.

Link:
https://onlinecourses.nptel.ac.in/noc21_cs79/preview

HONORS	Social Network Analysis	L	T	P	C
		3	0	0	3

Course Objective:

- Social network analysis is to understand a community by mapping the relationships that connect them as a network, and then trying to draw out key individuals, groups within the network and/or associations between the individuals.

Unit-I:

Introduction to Social Network Analysis, Introduction to Colab, Introduction to NetworkX, **Network Measures**- Basics, Node Centrality, Transitive and Reciprocity, Similarity and Degeneracy.

Unit-II:

Network Growth Models-Properties of real world network, Random Network Model, Ring Lattice, Network Model, Watts Strogatz Model, Link **Analysis**- Applications, Signal Networks, Strong and Weak Ties.

Unit III:

Graph Visualization Tools. Community Detection - Applications, Types of communities, Detection Methods, Overlapping, Community Detection Vs Community v's Search. **Link Prediction-**Applications, Temporal changes in network, Prediction Networks and Heuristic Models.

Unit IV:

Cascade Behaviour- Models, Probabilistic Cascades, Epedemic Models, Independent Models, CascadePrediction. **Network Effects&Anomaly Detection**- Outliers and Network Based Anomalies, Challenges Anomaly detection in static and dynamic models.,

Unit-V:

Graph Representation Learning, Coding on Graph Representation Learning- Criteria on GRL, Representation Learning Methods**, Applications and Case Studies**- Malicious activities on OSNs and Recommendation Systems.

Course Outcomes:

1. Understand the importance of social network analysis and various network measures.
2. Summarize network growth models and applications of link analysis
3. Apply different graph visualization tools on community detection and link predictions.
4. Development of models using cascading behavior with promising network effects.
5. Development of Recommendation Systems

Books and references

1. Social Network Analysis, Tanmoy Chakraborty, Wiley, 2021
2. Network Science, Albert-LazzloBarabasi
3. Social Network Analysis: Methods and Applications, Stanley Wasserman, Katherine Faus

Link:

https://onlinecourses.nptel.ac.in/noc22_cs117/preview

HONORS	Deep Learning for Computer Vision	L	T	P	C
		3	0	0	3

Course Objective:

- Recognize and describe how mathematical and scientific concepts are applied in computer vision.
- Identify and interpret appropriate sources of information relating to computer vision.
- Apply knowledge of computer vision to real life scenarios.

Unit-1:

Introduction and Overview: Introduction to Image Formation:Capture and Representation; Linear Filtering, Correlation and Convolution. Visual Features and Representations: Edge, Blobs, Corner Detection; Scale Space and Scale Selection; SIFT, SURF; HoG, LBP, etc.

Unit-II:

Visual Matching: Bag-of-words, VLAD; RANSAC, Hough transform; Pyramid Matching; Optical Flow. **Deep Learning Review-** Review of Deep Learning, Multi-layer Perceptrons, Backpropagation. **Convolutional Neural Networks (CNNs)-** Introduction to CNNs; Evolution of CNN Architectures: AlexNet, ZFNet, VGG, InceptionNets, ResNets, DenseNets

Unit-III:

Visualization and Understanding CNNs: Visualization of Kernels; Backprop-to-image/Deconvolution Methods; Deep Dream, Hallucination, Neural Style Transfer; CAM,Grad-CAM, Grad-CAM++; Recent Methods (IG, Segment-IG, SmoothGrad)

Unit-IV

CNNs for Recognition and Verification (Siamese Networks, Triplet Loss, Contrastive Loss, Ranking Loss); CNNs for Detection: Background of Object Detection, R-CNN, Fast R-CNN, Faster R-CNN, YOLO, SSD, RetinaNet; CNNs for Segmentation: FCN, SegNet, U-Net, Mask-RCNN. **Recurrent Neural Networks (RNNs):** Review of RNNs; CNN + RNN Models for Video Understanding: Spatio-temporal Models, Action/Activity Recognition

Unit-V:

Attention Models: Introduction to Attention Models in Vision; Vision and Language: Image Captioning, Visual QA, Visual Dialog; Spatial Transformers; Transformer Networks. **Deep Generative Models:**Deep Generative Models: GANs, VAEs; Other Generative Models: PixelRNNs, NADE, Normalizing Flows. Applications.

Course Outcomes:

1. Describe how mathematical and scientific concepts are applied in computer vision through images.
2. Identify appropriate sources of information and architecture relating to computer vision.
3. Understand the concept of visualization using CNN.
4. Apply the concepts of advanced CNN and RNN techniques for computer vision problems.
5. Apply knowledge of computer vision to real life scenarios.

Text Books :

1. Ian Goodfellow, YoshuaBengio, Aaron Courville, Deep Learning, 2016
2. Michael Nielsen, Neural Networks and Deep Learning, 2016

References Books:

3. YoshuaBengio, Learning Deep Architectures for AI, 2009
4. Richard Szeliski, Computer Vision: Algorithms and Applications, 2010.
5. Simon Prince, Computer Vision: Models, Learning, and Inference, 2012.
6. David Forsyth, Jean Ponce, Computer Vision: A Modern Approach, 2002.

Link:

https://onlinecourses.nptel.ac.in/noc20_cs88/preview

HONORS	BIO-INFORMATICS: APPLICATIONS AND ALGORITHMS	L	T	P	C
		3	0	0	3

Course Objective:

- Bioinformatics is an interdisciplinary field of science for analysing and interpreting vast biological data using computational techniques.
- Major aspects of bioinformatics such as the development of databases, computationally derived hypothesis, algorithms, and computer-aided drug design are illustrated in detail.

Unit-I: Introduction, DNA sequence analysis, DNA Databases. Protein structure and function, protein sequence databases, sequence alignment. PAM matrix, Global and local alignment, BLAST: features and scores.

Unit-II:

Multiple sequence alignment, Conservation score, phylogenetic trees. Protein sequence analysis, hydrophobicity profiles, non-redundant datasets. Protein secondary structures, Ramachandran plot, propensity, secondary structure prediction

Unit-III:

Protein tertiary structure, Protein Data Bank, visualization tools, structural classification. Protein structural analysis, protein structure prediction.

Unit-IV:

Protein stability, energetic contributions, database, stabilizing residues.Protein folding rates, proteins interactions, binding site residues

Unit-V:

Computer aided drug design, docking, screening, QSAR. Development of algorithms, awk programming, machine learning techniques, applications using WEKA.

Course Outcomes:

1. Understand DNA and protein sequence databases and analysis.
2. Illustrate the secondary structures and 3D structural analysis.
3. Summarize the potential of protein tertiary structure and work on classification.
4. Interpret the protein stability,protein structure, folding rates, stability upon mutation, and intermolecular interactions.
5. Development of computer-aided drug design using docking and QSAR studies.

Text Books :

1. M. Michael Gromiha, Protein Bioinformatics: From Sequence to Function, Academic Press, 2010
2. D.E. Krane and M.L. Raymer, Fundamental concepts of bioinformatics, Pearson Education Inc. 2006

Link:

https://archive.nptel.ac.in/content/syllabus_pdf/102106065.pdf
https://onlinecourses.nptel.ac.in/noc21_bt06/preview

HONORS	GPU Architecture and Programming	L	T	P	C
		3	0	0	3

Course Objective:

- Design of multi- and many-core processors such as graphics processing units (GPUs).
- Philosophy of GPU concepts, including the evolution of GPU computing at high-level.
- Overview of GPU architecture, and key differences from CPU architecture.

Unit-I:

Review of Traditional Computer Architecture – Basic five stage RISC Pipeline, Cache Memory, Register File, SIMD instructions. GPU architectures - Streaming Multi Processors, Cache Hierarchy,The Graphics Pipeline

Unit-II:

Introduction to **CUDA programming** and Multi-dimensional mapping of dataspace, Synchronization

Unit-III:

Warp Scheduling- Host System- External Hardware and GPU roles – policies and features,**Divergence and Memory Access Coalescing**- Differences between Threads and Arrays in Coalescing.

Unit-IV:

Optimization Case Studies-Optimizing Reduction Kernels, Kernel Fusion, Thread and Block.

Unit-V:

OpenCL -Basics- OpenCL for Heterogeneous Computing, Application Design: Efficient Neural Network Training/Inferencing

Course Outcomes:

1. Understand the basic architecture of GPU and Pipeline Process.
2. Apply CUDA Programming and Multi-dimensional programming of data spacing.
3. Illustrate the necessity of Warp Scheduling and Memory Access.
4. Summarize the role of GPU in optimization case studies.
5. Apply Open CL in developing heterogeneous components.

Books and references

1. “Computer Architecture -- A Quantitative Approach” - John L.Hennessy and David A. Patterson
2. "Programming Massively Parallel Processors" - David Kirk and Wen-meiHwu
3. Heterogeneous Computing with OpenCL” -- Benedict Gaster,Lee Howes, David R. Kaeli

Link:

https://onlinecourses.nptel.ac.in/noc20_cs41/preview

HONORS	AI for Health Care	L	T	P	C
		3	0	0	3

Course Objective:

- The students should be able to understand how Al is transforming the practice of medicine.
- The students should learn the practical experience in applying machine learning to concrete problems in medicine

Unit-I:

Disease detection with computer vision- Medical Image Diagnosis, Eye Disease and Cancer Diagnosis, Building and Training a Model for Medical Diagnosis, Training, prediction, and loss, Image Classification and Class Imbalance, Generating More Samples, Model Testing

Unit-II:

Evaluating models- Sensitivity, Specificity, and Evaluation Metrics, Accuracy in terms of conditional probability, Confusion matrix, ROC curve and Threshold**. Image segmentation on MRI images-** Medical Image Segmentation, MRI Data and Image Registration, Segmentation, 2D U-Net and 3D U-Net Data augmentation and loss function for segmentation, Different Populations and Diagnostic Technology, External validation

Unit-III:

Linear prognostic models- Medical Prognosis, Atrial fibrillation, Liver Disease Mortality, Risk of heart disease, Evaluating Prognostic Models, Concordant Pairs, Risk Ties, Permissible Pairs.**Prognosis with Tree-based models-** Decision trees for prognosis, fix overfitting, Different distributions, Missing Data example, Imputation

Unit-IV:

Survival Models and Time- Survival Model, Survival function, collecting time data, Estimating the survival function.Build a risk model using linear and tree-based models**.** Hazard Functions, Relative risk, Individual vs. baseline hazard, Survival Trees, Nelson Aalen estimator.

Unit-V:

Medical Treatment Effect Estimation- Analyze data from a randomized control trial, Average treatment effect, Conditional average treatment effect, T-Learner, S-Learner, C-for-benefit.

Course Outcomes:

After completion of course, students would be able to:

1. Apply tree-based machine learning to estimate patient survival rates.
2. Analyse convolutional neural network image classification and segmentation models to make diagnoses of lung and brain disorders.
3. Apply natural language processing to extract information from unstructured medical data.
4. Apply survival models to build realistic applications.
5. Understand different types of prognosis models related to different diseases.

Books and References

- Deep Medicine: How Artificial Intelligence Can Make Healthcare Human Again, Eric Topol, Basic Books, 1st edition 2019.
- Machine Learning and Al for Healthcare: Big Data for Improved Health Outcomes, Arjun Panesar, Apress, 1st ed. Edition, 2019.
- Artificial Intelligence in Healthcare, 2020, ISBN 978-0-12-818438-7, Elsevier Inc.

Links

1. https://www.coursera.org/learn/ai-for-medical-diagnosis
2. https://www.coursera.org/learn/ai-for-medical-prognosis#syllabus
3. https://www.coursera.org/learn/ai-for-medical-treatment/#syllabus

MINORS COURSES

Minor in IT

01. Principles of Database Management Systems
02. Principles of Software Engineering
03. Advanced Data Structures & Algorithm Analysis
04. Principles of Operating Systems

Any of the following 12 Week 3 credit NPTEL MOOC Courses

05. Artificial Intelligence: Knowledge Representation and Reasoning
06. Computer Networks and Internet Protocol
07. Machine Learning and Deep Learning - Fundamentals and Applications
08. The Joy of Computing using Python
09. Discrete Mathematics for CS
10. Human Computer Interaction (In English)
11. Data Analytics with Python
12. Foundations of Cyber Physical Systems

Subject Code	Subject Name	L	T	P	C
MINORS	Principles of Database Management Systems	3	0	0	3

Course Objectives:

The main objectives of the course is to

- Introduce database management systems and to give a good formal foundation on the relational model of data and usage of Relational Algebra
- Introduce the concepts of basic SQL as a universal Database language
- Demonstrate the principles behind systematic database design approaches by covering conceptual design, logical design through normalization
- Provide an overview of physical design of a database system, by discussing Database indexing techniques and storage techniques

UNIT I:

Introduction: Database system, Characteristics (Database Vs File System), Database Users, Advantages of Database systems, Database applications. Brief introduction of different Data Models; Concepts of Schema, Instance and data independence; Three tier schema architecture for data independence; Database system structure, environment, Centralized and Client Server architecture for the database.

Entity Relationship Model: Introduction, Representation of entities, attributes, entity set, relationship, relationship set, constraints, sub classes, super class, inheritance, specialization, generalization using ER Diagrams.

UNIT II:

Relational Model: Introduction to relational model, concepts of domain, attribute, tuple, relation, importance of null values, constraints (Domain, Keyconstraints, integrity constraints) and their importance, Relational Algebra, Relational Calculus. BASIC SQL: Simple Database schema, data types, table definitions (create, alter), different DML operations (insert, delete, update).

UNIT III:

SQL: Basic SQL querying (select and project) using where clause, arithmetic & logical operations, SQL functions(Date and Time, Numeric, String conversion).Creating tables with relationship, implementation of key and integrity constraints, nested queries, sub queries, grouping, aggregation, ordering, implementation of different types of joins, view (updatable and non- updatable), relational set operations.

UNIT IV:

Schema Refinement (Normalization): Purpose of Normalization or schema refinement, concept of functional dependency, normal forms based on functional dependency Lossless join and dependency preservingdecomposition, (1NF, 2NF and 3 NF), concept of surrogate key, Boyce-Codd normal form(BCNF), MVD, Fourth normal form(4NF), Fifth Normal Form (5NF).

UNIT V:

Transaction Concept: Transaction State, ACID properties, Concurrent Executions, Serializability, Recoverability, Implementation of Isolation, Testing for Serializability, lock based, time stamp based, optimistic, concurrency protocols, Deadlocks, Failure Classification, Storage, Recovery and Atomicity, Recovery algorithm. Introduction to Indexing Techniques: B+ Trees, operations on B+Trees, Hash Based Indexing:

Text Books:

3) Database Management Systems, 3rd edition, Raghurama Krishnan, Johannes Gehrke, TMH (For Chapters 2, 3, 4)
4) Database System Concepts,5th edition, Silberschatz, Korth, Sudarsan, TMH (For Chapter 1 and Chapter 5)

Reference Books:

4) Introduction to Database Systems, 8thedition, C J Date, Pearson.
5) Database Management System, 6th edition, Ramez Elmasri, Shamkant B. Navathe, Pearson
6) Database Principles Fundamentals of Design Implementation and Management, Corlos Coronel, Steven Morris, Peter Robb, Cengage Learning.

Web-Resources:

3) https://nptel.ac.in/courses/106/105/106105175/
4) https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_01275806667282022456_shared/overview

Subject Code	Subject Name	L	T	P	C
MINORS	Principles of Software Engineering	3	0	0	3

Course Objectives:

The objectives of this course are to introduce

- Software life cycle models, Software requirements and SRS document.
- Project Planning, quality control and ensuring good quality software.
- Software Testing strategies, use of CASE tools, Implementation issues, validation & verification procedures.

UNIT I:

Introduction: Evolution, Software development projects, Exploratory style of software developments, Emergence of software engineering, Notable changesin software development practices, Computer system engineering. Software Life Cycle Models: Basic concepts, Waterfall model and its extensions, Rapid application development, Agile development model, Spiralmodel.

UNIT II:

Software Project Management: Software project management complexities, Responsibilities of a software project manager, Metrics for project size estimation, Project estimation techniques, Empirical Estimation techniques, COCOMO, Halstead's software science, risk management.

Requirements Analysis and Specification: Requirements gathering and analysis, Software Requirements Specification (SRS), Formal system specification, Axiomatic specification, Algebraic specification, Executable specification and 4GL.

UNIT III:

Software Design: Overview of the design process, how to characterize a good software design? Layered arrangement of modules, Cohesion and Coupling. Approaches to software design.

Agility: Agility and the Cost of Change, Agile Process, Extreme Programming (XP), Other Agile Process Models, Tool Set for the Agile Process (Text Book 2)

Function-Oriented Software Design: Overview of SA/SD methodology, structured analysis, Developing the DFD model of a system, Structured design, Detailed design, and Design Review.

User Interface Design: Characteristics of a good user interface, Basic concepts, Types of user interfaces, Fundamentals of component-based GUI development, and user interface design methodology.

UNIT IV:

Coding And Testing: Coding, Code review, Software documentation, Testing, Black-box testing, White-Box testing, Debugging, Program analysis tools, Integration testing, Testing object-oriented programs, Smoke testing, and Some general issues associated with testing.

Software Reliability and Quality Management: Software reliability. Statistical testing, Software quality, Software quality management system, ISO 9000. SEI Capability maturity model. Few other important quality standards, and Six Sigma.

UNIT V:

Computer-Aided Software Engineering (Case): CASE and its scope, CASE environment, CASE support in the software life cycle, other characteristics of CASE tools, Towards second generation CASE Tool, and Architecture of a CASE Environment.

Software Maintenance: Characteristics of software maintenance, Software reverse engineering, Software maintenance process models and Estimation of maintenance cost.
Software Reuse: reuse- definition, introduction, reason behind no reuse so far, Basic issues in any reuse program, A reuse approach, and Reuse at organization level.

Text Books:

1. Fundamentals of Software Engineering, Rajib Mall, 5th Edition, PHI.
2. Software Engineering A practitioner's Approach, Roger S. Pressman, 9th Edition, Mc-Graw Hill International Edition.

Reference Books:

1. Software Engineering, Ian Sommerville,10th Edition, Pearson.
2. Software Engineering, Principles and Practices, Deepak Jain, Oxford University Press.

e-Resources:

1)https://nptel.ac.in/courses/106/105/106105182/
2)https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_
01260589506387148827_shared/overview
3)https://infyspringboard.onwingspan.com/web/en/app/toc/lex_auth_
01338269041100390473 5_shared/overview

Subject Code	Subject Name	L	T	P	C
MINORS	Advanced Data Structures & Algorithm Analysis	3	0	0	3

Course Objectives:

The main objectives of the course is to

• provide knowledge on advance data structures frequently used in Computer Science domain

• Develop skills in algorithm design techniques popularly used

• Understand the use of various data structures in the algorithm design

Course Outcomes:

After completion of the course, students will be able to

1. Illustrate the working of the advanced tree data structures and their applications (L2)

2. Understand the Graph data structure, traversals and apply them in various contexts. (L2)

3. Use various data structures in the design of algorithms (L3)

4. Recommend appropriate data structures based on the problem being solved (L5)

5. Analyze algorithms with respect to space and time complexities (L4)

UNIT – I: Introduction to Algorithm Analysis, Space and Time Complexity analysis, Asymptotic Notations. AVL Trees – Creation, Insertion, Deletion operations and Applications Heap Trees (Priority Queues) – Min and Max Heaps, Operations and Applications

UNIT – II: Graphs – Terminology, Representations, Basic Search and Traversals, Connected Components and Biconnected Components, applications Divide and Conquer: The General Method, Quick Sort, Merge Sort, Strassen's matrix multiplication, Convex Hull

UNIT – III: Greedy Method: General Method, Job Sequencing with deadlines, Knapsack Problem, Minimum cost spanning trees, Single Source Shortest Paths Dynamic Programming: General Method, All pairs shortest paths, Single Source Shortest Paths– General Weights (Bellman Ford Algorithm), Optimal Binary Search Trees, 0/1 Knapsack, String Editing, Travelling Salesperson problem

UNIT – IV: Backtracking: General Method, 8-Queens Problem, Sum of Subsets problem, Graph Coloring, 0/1 Knapsack Problem Branch and Bound: The General Method, 0/1 Knapsack Problem, Travelling Salesperson problem

UNIT – V: NP Hard and NP Complete Problems: Basic Concepts, Cook's theorem NP Hard Graph Problems: Clique Decision Problem (CDP), Chromatic Number Decision Problem (CNDP), Traveling Salesperson Decision Problem (TSP) NP Hard Scheduling Problems: Scheduling Identical Processors, Job Shop Scheduling

Textbooks:

1. Fundamentals of Data Structures in C++, Horowitz, Ellis; Sahni, Sartaj; Mehta, Dinesh, 2nd Edition Universities Press

2. Computer Algorithms in C++, Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran, 2nd Edition University Press

Reference Books:

1. Data Structures and program design in C, Robert Kruse, Pearson Education Asia

2. An introduction to Data Structures with applications, Trembley & Sorenson, McGraw Hill

3. The Art of Computer Programming, Vol.1: Fundamental Algorithms, Donald E Knuth, Addison-Wesley, 1997.

4. Data Structures using C & C++: Langsam, Augenstein & Tanenbaum, Pearson, 1995

5. Algorithms + Data Structures & Programs:, N.Wirth, PHI

6. Fundamentals of Data Structures in C++: Horowitz Sahni & Mehta, Galgottia Pub.

7. Data structures in Java:, Thomas Standish, Pearson Education Asia

Online Learning Resources:

1. https://www.tutorialspoint.com/advanced_data_structures/index.asp

2. http://peterindia.net/Algorithms.html

3. https://www.youtube.com/playlist?list=PLDN4rrl48XKpZkf03iYFl-O29szjTrs_O

Subject Code	Subject Name	L	T	P	C
MINORS	Principles of Operating Systems	3	0	0	3

Course Objectives:

- Provide knowledge about the services rendered by operating systems.
- Present detail discussion on processes, threads and scheduling algorithms.
- Expose the student with different techniques of handling deadlocks.
- Discuss various file-system implementation issues and memory management techniques.
- Learn the basics of Linux operating system.

UNIT-I:
Operating Systems Overview: Introduction: what is an operating system, Types of operating systems, operating systems concepts, operating systems services, Introduction to System call, System call types, Operating System Generation.

UNIT–II:
Process Management: Process concept: Process Concept, Process Scheduling, Operations on Processes, Inter process Communication. Multithreaded Programming: Overview, Multithreading models, Threading Issues. Process scheduling: Basic Concepts, Scheduling Criteria, Scheduling Algorithms.

UNIT-III:
Synchronization: Process Synchronization, The Critical-Section Problem- Dekker's Solution, Petersons Solution, Synchronization Hardware, Semaphores, Classic Problems of Synchronization, Monitors, Synchronization examples. Principles of deadlock: System Model, Deadlock Characterization, Deadlock Prevention, Detection and Avoidance, Recovery form Deadlock.

UNIT-IV:
Memory Management: Memory Management strategies: Swapping, Contiguous Memory Allocation, Segmentation, Paging, Structure of the Page Table. Virtual Memory Management: Virtual Memory, Demand Paging, Page-Replacement Algorithms, Thrashing.

UNIT-V:
File system Interface- the concept of a file, Access Methods, Directory and Disk structure, File system mounting. File System implementation: File system structure, allocation methods, Free-space management. Mass-storage structure: overview of Mass-storage structure, Disk scheduling, Device drivers
Android Software Platform: Android Architecture, Operating System Services, Android Runtime Application Development, Application Structure, Application Process management.

Course Outcomes:

1. Understand the importance of operating systems and different types of system calls.
2. Analyse process scheduling algorithms and various IPC mechanisms.
3. Analyse the process synchronization, different ways for deadlocks handling.
4. Analyse different page replacement methods, various File management techniques.
5. Understand Android environment and behaviour.

Text Books:
1. Silberschatz A, Galvin P B, and Gagne G, Operating System Concepts, 9thedition, Wiley, 2013.

2. Tanenbaum A S, Modern Operating Systems, 3rd edition, Pearson Education, 2008.

Reference Books:

1. Tanenbaum A S, Woodhull A S, Operating Systems Design and Implementation, 3rd edition, PHI, 2006.
2. Dhamdhere D M, Operating Systems A Concept Based Approach, 3rd edition, Tata McGraw-Hill, 2012.
3. Stallings W, Operating Systems -Internals and Design Principles, 6th edition, Pearson Education, 2009.
4. Nutt G, Operating Systems, 3rd edition, Pearson Education, 2004

Subject Code	Subject Name	L	T	P	C
MINORS	Artificial Intelligence: Knowledge Representation and Reasoning	3	0	0	3

Course Objectives:

1. To introduce the need for knowledge in intelligent systems.
2. To understand various knowledge representation (KR) techniques.
3. To learn logical inference using propositional and predicate logic.
4. To explore structured KR using semantic networks, frames, and scripts.
5. To study reasoning under uncertainty using fuzzy logic and probabilistic models.

UNIT I: Introduction to Knowledge Representation

Knowledge and its role in intelligent behavior, types of knowledge, characteristics of good knowledge representation systems. Introduction to propositional logic, syntax, semantics, inference rules, applications, and limitations.

UNIT II: Predicate Logic and Inference

First-order predicate logic, syntax and semantics, quantifiers, substitution. Knowledge engineering using predicate logic. Logical inference techniques: unification, resolution, forward chaining, backward chaining, and their applications in AI systems.

UNIT III: Structured Representations

Semantic networks, frames, scripts, and conceptual dependencies. Comparison of structured representation methods.Representation of common-sense knowledge and typical relationships.Use of ontologies for domain modeling.

UNIT IV: Non-Monotonic Reasoning and Uncertainty

Default reasoning, truth maintenance systems, non-monotonic logics. Introduction to probabilistic reasoning: Bayesian networks, Dempster-Shafer theory. Basics of fuzzy logic and reasoning with vague and incomplete data.

UNIT V: Applications and Advanced Topics

Applications of KR in natural language processing, expert systems, and robotics. Ontologies, description logic, and semantic web. Case studies in medical diagnosis, intelligent agents, and reasoning in real-time environments.

Course Outcomes:

After completing this course, students will be able to:

1. Explain the importance of knowledge in artificial intelligence.
2. Represent information using logic-based and structured models.
3. Perform inference using resolution and unification techniques.
4. Apply non-monotonic and uncertain reasoning to AI problems.
5. Build simple knowledge-based systems for real-world applications.

Textbooks:

1. Elaine Rich, Kevin Knight, Shivashankar B. Nair, *Artificial Intelligence*, 3rd Edition, McGraw-Hill.
2. Stuart Russell, Peter Norvig, *Artificial Intelligence: A Modern Approach*, 3rd Edition, Pearson.

Reference Books:

1. Brachman& Levesque, *Knowledge Representation and Reasoning*, Elsevier.
2. Nils J. Nilsson, *Principles of Artificial Intelligence*, Morgan Kaufmann.
3. John Luger, *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*, Pearson.

Subject Code	Subject Name	L	T	P	C
MINORS	Computer Networks and Internet Protocol	3	0	0	3

Course Objectives:

1. To introduce the fundamentals of computer networks, including network types, architecture, and protocols.
2. To understand the functionalities of each layer in the OSI and TCP/IP reference models.
3. To study IP addressing, subnetting, and routing protocols in detail.
4. To analyze the working of transport layer protocols and congestion control techniques.
5. To familiarize students with common application layer protocols and basic network security concepts.

UNIT I: Introduction to Computer Networks
Basics of computer networks, types of networks, transmission media, network topologies, circuit switching, packet switching, OSI model, TCP/IP model.

UNIT II: Data Link Layer
Framing, error detection, error correction, flow control, stop-and-wait protocol, sliding window protocol, medium access control protocols, ALOHA, CSMA/CD, Ethernet, switches, VLANs.

UNIT III: Network Layer and IP Addressing
Logical addressing, IPv4, IPv6, subnetting, supernetting, routing algorithms, distance vector routing, link state routing, IP protocol, ICMP protocol, RIP, OSPF.

UNIT IV: Transport Layer
TCP, UDP, connection establishment, three-way handshake, flow control, error control, congestion control, leaky bucket algorithm, token bucket algorithm, quality of service, multiplexing, demultiplexing.

UNIT V: Application Layer and Security
HTTP, FTP, DNS, SMTP, DHCP, symmetric cryptography, asymmetric cryptography, authentication, firewalls, virtual private networks (VPNs).

Course Outcomes:

Upon successful completion of the course, the student will be able to:

1. Explain the structure and functionality of computer networks and communication models.
2. Demonstrate knowledge of data link layer operations including framing, error detection, and MAC protocols.
3. Apply IP addressing, subnetting, and routing techniques to design and troubleshoot networks.
4. Analyze transport layer protocols and evaluate congestion control methods.
5. Describe the role of application layer protocols and implement basic network security techniques.

Textbooks:

1. Andrew S. Tanenbaum and David J. Wetherall, *Computer Networks*, 5th Edition, Pearson.
2. Behrouz A. Forouzan, *Data Communications and Networking*, 5th Edition, McGraw-Hill Education.

Reference Books:

1. William Stallings, *Data and Computer Communications*, 10th Edition, Pearson.
2. Larry L. Peterson and Bruce S. Davie, *Computer Networks: A Systems Approach*, 5th Edition, Morgan Kaufmann.
3. James F. Kurose and Keith W. Ross, *Computer Networking: A Top-Down Approach*, 7th Edition, Pearson.

Subject Code	Subject Name	L	T	P	C
MINORS	Machine Learning and Deep Learning - Fundamentals and Applications	3	0	0	3

Course Objectives:

1. To introduce the fundamental concepts of machine learning (ML) and deep learning (DL).
2. To explore supervised, unsupervised, and reinforcement learning techniques.
3. To study the architecture and functioning of deep neural networks.
4. To implement real-time applications using ML and DL algorithms.
5. To expose students to tools and platforms used in ML/DL projects.

UNIT I: Introduction to Machine Learning

Introduction to machine learning, differences between AI, ML, and DL, types of machine learning – supervised, unsupervised, semi-supervised, and reinforcement learning. Steps in developing a machine learning model. Applications in areas like healthcare, finance, agriculture, and robotics.

UNIT II: Supervised Learning

Linear regression, logistic regression, decision trees, support vector machines (SVM), k-nearest neighbors (KNN), naïve Bayes classifiers. Evaluation metrics – accuracy, precision, recall, F1 score, confusion matrix. Training, testing, and cross-validation techniques.

UNIT III: Unsupervised and Reinforcement Learning

Clustering techniques – K-means, hierarchical clustering, DBSCAN. Dimensionality reduction – PCA, t-SNE. Basics of reinforcement learning – agent, environment, reward, policy, value function. Introduction to Q-learning.

UNIT IV: Fundamentals of Deep Learning

Neural networks – perceptron, activation functions, feedforward and backpropagation algorithms. Deep neural networks, convolutional neural networks (CNNs), recurrent neural networks (RNNs), LSTMs. Introduction to frameworks like TensorFlow and PyTorch.

UNIT V: Applications and Case Studies

Application of ML/DL in image classification, object detection, speech recognition, natural language processing (NLP), time series prediction. Case studies from healthcare (e.g., disease prediction), finance (fraud detection), and self-driving cars. Deployment of ML models and model evaluation in real-world scenarios.

Course Outcomes:

After completing this course, students will be able to:

1. Understand basic concepts and types of learning in ML and DL.
2. Apply supervised and unsupervised learning methods to real-world problems.
3. Design and train deep neural networks for various applications.
4. Use appropriate ML/DL tools for data analysis and model development.
5. Solve domain-specific problems using ML and DL techniques.

Textbooks:

1. AurélienGéron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 2nd Edition, O'Reilly.

2. Ian Goodfellow, YoshuaBengio, and Aaron Courville, *Deep Learning*, MIT Press.

Reference Books:

1. Tom M. Mitchell, *Machine Learning*, McGraw-Hill.
2. Trevor Hastie, Robert Tibshirani, Jerome Friedman, *The Elements of Statistical Learning*, Springer.
3. Kelleher, Mac Namee, and D'Arcy, *Fundamentals of Machine Learning for Predictive Data Analytics*, MIT Press.

Subject Code	Subject Name	L	T	P	C
MINORS	The Joy of Computing using Python	3	0	0	3

Course Objectives:
The main objectives of the course are to

- Introduce core programming concepts of Python programming language.
- Demonstrate about Python data structures like Lists, Tuples, Sets and dictionaries
- Implement Functions, Modules and Regular Expressions in Python Programming and to create practical and contemporary applications using these

UNTI-I:
History of Python Programming Language, Thrust Areas of Python, Installing Anaconda Python Distribution, Installing and Using Jupyter Notebook. Parts of Python Programming Language: Identifiers, Keywords, Statements and Expressions, Variables, Operators, Precedence and Associativity, Data Types, Indentation, Comments, Reading Input, Print Output, Type Conversions, the type () Function and Is Operator, Dynamic and Strongly Typed Language. Control Flow Statements: if statement, if-else statement, if...elif...else, Nested if statement, while Loop, for Loop, continue and break Statements, Catching Exceptions Using try and except Statement.

Sample Experiments:
1. Write a program to find the largest element among three Numbers.
2. Write a Program to display all prime numbers within an interval
3. Write a program to swap two numbers without using a temporary variable.
4. Demonstrate the following Operators in Python with suitable examples.
i) Arithmetic Operators ii) Relational Operators iii) Assignment Operators iv) Logical Operators v) Bit wise Operators vi) Ternary Operator vii) Membership Operators viii) Identity Operators
5. Write a program to add and multiply complex numbers
6. Write a program to print multiplication table of a given number.

UNIT-II:
Functions: Built-In Functions, Commonly Used Modules, Function Definition and Calling the function, return Statement and void Function, Scope and Lifetime of Variables, Default Parameters, Keyword Arguments, *args and **kwargs, Command Line Arguments. Strings: Creating and Storing Strings, Basic String Operations, Accessing Characters in String by Index Number, String Slicing and Joining, String Methods, Formatting Strings. Lists: Creating Lists, Basic List Operations, Indexing and Slicing in Lists,BuiltIn Functions Used on Lists, List Methods, del Statement.

Sample Experiments:
7. Write a program to define a function with multiple return values.
8. Write a program to define a function using default arguments.
9. Write a program to find the length of the string without using any library functions.
10. Write a program to check if the substring is present in a given string ornot.
11. Write a program to perform the given operations on a list:
i. addition ii. Insertion iii. slicing
12. Write a program to perform any 5 built-in functions by taking any list.

UNIT-III:
Dictionaries: Creating Dictionary, Accessing and Modifying key:value Pairs in Dictionaries, Built-In Functions Used on Dictionaries, Dictionary Methods, del Statement. Tuples and Sets: Creating Tuples, Basic Tuple Operations, tuple() Function, Indexing and Slicing in Tuples, Built-In Functions Used on Tuples, Relation between Tuples and Lists, Relation between Tuples and Dictionaries, Using zip() Function, Sets, Set Methods, Frozenset.

Sample Experiments:
13. Write a program to create tuples (name, age, address, college) for at least two members and concatenate the tuples and print the concatenated tuples.
14. Write a program to count the number of vowels in a string (No controlflow allowed).
15. Write a program to check if a given key exists in a dictionary or not.
16. Write a program to add a new key-value pair to an existing dictionary.
17. Write a program to sum all the items in a given dictionary.

UNIT-IV:
Files: Types of Files, Creating and Reading Text Data, File Methods to Read and Write Data, Reading and Writing Binary Files, Pickle Module, Reading and Writing CSV Files, Python os and os.path Modules. Object-Oriented Programming: Classes and Objects, Creating Classes in Python, Creating Objects in Python, Constructor Method, Classes with Multiple Objects, Class Attributes Vs Data Attributes, Encapsulation, Inheritance, Polymorphism.

Sample Experiments:
18. Write a program to sort words in a file and put them in another file.The output file should have only lower-case words, so any upper-case words from source must be lowered.
19. Python program to print each line of a file in reverse order.
20. Python program to compute the number of characters, words and lines in a file.
21. Write a program to create, display, append, insert and reverse the order of the items in the array.
22. Write a program to add, transpose and multiply two matrices.
23. Write a Python program to create a class that represents a shape. Include methods to calculate its area and perimeter. Implement subclasses for different shapes like circle, triangle, and square.

UNIT-V:
Introduction to Data Science: Functional Programming, JSON and XML in Python, NumPy with Python, Pandas.

Sample Experiments:
24. Python program to check whether a JSON string contains complex object or not.
25. Python Program to demonstrate NumPy arrays creation using array () function.
26. Python program to demonstrate use of ndim, shape, size, dtype.
27. Python program to demonstrate basic slicing, integer and Boolean indexing.
28. Python program to find min, max, sum, cumulative sum of array
29. Create a dictionary with at least five keys and each key represent valueas a list where this list contains at least ten values and convert this dictionary as a pandas data frame and explore the data through the data frame as follows:
    a) Apply head () function to the pandas data frame
    b) Perform various data selection operations on Data Frame
30. Select any two columns from the above data frame, and observe the change in one attribute with respect to other attribute with scatter and plot operations in matplotlib

Reference Books:
1. Gowri shankar S, Veena A., Introduction to Python Programming, CRC Press.
2. Python Programming, S Sridhar, J Indumathi, V M Hariharan, 2nd Edition, Pearson, 2024
3. Introduction to Programming Using Python, Y. Daniel Liang, Pearson.

Online Learning Resources/Virtual Labs:
1. https://www.coursera.org/learn/python-for-applied-data-science-ai
2. https://www.coursera.org/learn/python?specialization=python#sylla bus

Subject Code	Subject Name	L	T	P	C
MINORS	Discrete Mathematics for CS	3	0	0	3

Course Objectives:

1. To introduce the fundamental concepts of discrete mathematics used in computer science.
2. To understand logic, sets, functions, and relations and their applications.
3. To study combinatory, counting techniques, and mathematical induction.
4. To explore graphs, trees, and their relevance in algorithms and data structures.
5. To develop the ability to model and solve problems using discrete structures.

UNIT I: Mathematical Logic and Proof Techniques

Introduction to propositional logic, logical connectives, truth tables, tautologies, contradictions, logical equivalence. Predicates and quantifiers.Rules of inference. Methods of proof – direct, contrapositive, contradiction, and mathematical induction.

UNIT II: Set Theory and Relations

Basic set operations, Venn diagrams, Cartesian products, power sets, principle of inclusion and exclusion. Properties of relations – reflexive, symmetric, transitive, equivalence relations. Closures and partitions.

UNIT III: Functions and Recursion

Types of functions – one-to-one, onto, inverse functions. Composition of functions.Growth of functions – big-O, big-Ω, and big-Θ notation.Recurrence relations, solutions using substitution and iteration.Recursive algorithms.

UNIT IV: Combinatorics and Counting

Basic counting principles – addition, multiplication rules. Permutations and combinations. Pigeonhole principle. Binomial theorem.Inclusion-exclusion principle.Applications in computing.

UNIT V: Graph Theory and Trees

Graphs – definitions, types, representation, degree, subgraphs, paths, cycles, connectivity. Graph traversal – BFS and DFS. Trees – properties, binary trees, spanning trees, minimum spanning tree (Prim's and Kruskal's algorithms). Applications in data structures and networks.

Course Outcomes:

After completing this course, students will be able to:

1. Apply logic and set theory to solve computational problems.
2. Understand functions, relations, and mathematical reasoning.
3. Use combinatorics and counting principles for problem-solving.
4. Analyze and apply graph theory and tree structures in computing.
5. Model and solve real-world problems using discrete mathematical concepts.

Textbooks:

1. Kenneth H. Rosen, *Discrete Mathematics and Its Applications*, 7th Edition, McGraw-Hill.
2. Tremblay and Manohar, *Discrete Mathematical Structures with Applications to Computer Science*, McGraw-Hill.

Reference Books:

1. Seymour Lipschutz and Marc Lipson, *Discrete Mathematics*, Schaum's Outline Series, McGraw-Hill.
2. Ralph P. Grimaldi, *Discrete and Combinatorial Mathematics: An Applied Introduction*, Pearson.
3. Thomas Koshy, *Discrete Mathematics with Applications*, Academic Press.

Subject Code	Subject Name	L	T	P	C
MINORS	Human Computer Interaction	3	0	0	3

Course Objectives:

1. To understand the fundamentals of human-computer interaction and user-centered design.
2. To study the psychological and social aspects of user interface design.
3. To learn how to design, evaluate, and improve user interfaces.
4. To introduce various interaction styles, devices, and design principles.
5. To explore usability testing, prototyping, and HCI applications in modern systems.

UNIT I: Introduction to Human-Computer Interaction

Definition and importance of HCI, human-centered design, usability goals and principles, history of HCI, interaction models, ergonomics and human factors, and the role of HCI in software engineering.

UNIT II: Human Capabilities and Interaction Styles

Understanding human perception, cognition, memory, attention, and learning in the context of interaction. Various interaction styles – command-line, menus, graphical user interfaces, direct manipulation, voice, and natural language interaction.

UNIT III: Interface Design and Development

Principles of user interface design, design process and guidelines, interface metaphors, design standards, consistency, feedback and error handling. Use of wireframes, mockups, and design patterns for creating interfaces.

UNIT IV: Evaluation Techniques and Usability Testing

Types of evaluation – formative and summative, heuristic evaluation, cognitive walkthroughs, usability testing methods, task analysis, user studies, surveys and focus groups, and interpreting evaluation results.

UNIT V: Advanced Topics and Applications

HCI in web, mobile, and ubiquitous computing. Accessibility and inclusive design. Emotional design, persuasive interfaces, HCI for AR/VR and wearable devices. Case studies and trends in intelligent interfaces and HCI research.

Course Outcomes:

Upon successful completion of the course, the student will be able to:

1. Explain the principles and models of human-computer interaction.
2. Design effective, user-friendly, and accessible user interfaces.
3. Apply usability and cognitive models to improve interaction systems.
4. Use techniques for interface prototyping and user evaluation.
5. Develop HCI solutions for web, mobile, and emerging technologies.

Textbooks:

1. Alan Dix, Janet Finlay, Gregory Abowd, Russell Beale, *Human-Computer Interaction*, 3rd Edition, Pearson Education.
2. Ben Shneiderman, Catherine Plaisant, *Designing the User Interface*, 6th Edition, Pearson.

Reference Books:

1. Don Norman, *The Design of Everyday Things*, MIT Press.
2. Jenny Preece, Yvonne Rogers, Helen Sharp, *Interaction Design: Beyond Human-Computer Interaction*, Wiley.

Wilbert Galitz, *The Essential Guide to User Interface Design*, Wiley

Subject Code	Subject Name	L	T	P	C
MINORS	Data Analytics with Python	3	0	0	3

Course Objectives:

1. To introduce basic concepts of data analytics and its role in real-world problem solving.
2. To teach data manipulation, analysis, and visualization using Python.
3. To explore Python libraries such as NumPy, Pandas, Matplotlib, and Seaborn.
4. To perform exploratory data analysis, statistical analysis, and data cleaning.
5. To apply analytics techniques in projects involving real-world datasets.

UNIT I: Introduction to Data Analytics

Overview of data analytics, data types and sources, structured and unstructured data, steps in data analytics process, real-world applications, introduction to Python programming and Jupyter notebooks.

UNIT II: Data Handling with Python

Working with Python data structures – lists, tuples, dictionaries. File handling and reading data from CSV/Excel. Introduction to NumPy for numerical operations.Pandas for dataframes, data selection, filtering, sorting, and aggregation.

UNIT III: Data Cleaning and Transformation

Handling missing values, duplicate data, and outliers. Data transformation – normalization, encoding categorical data, data binning, mapping and replacing values, date/time manipulation.

UNIT IV: Data Visualization and Analysis

Creating plots using Matplotlib and Seaborn – line plots, bar charts, histograms, box plots, scatter plots, heatmaps. Descriptive statistics – mean, median, mode, variance, standard deviation. Correlation and covariance analysis.

UNIT V: Exploratory Data Analysis and Case Studies

Performing EDA using Python. Case studies involving real datasets – analyzing trends, drawing inferences, and summarizing findings. Introduction to mini-projects using datasets from finance, healthcare, education, and e-commerce.

Course Outcomes:

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

1. Understand and explain the key steps in the data analytics process.
2. Use Python libraries for data handling, cleaning, and transformation.
3. Create visualizations and summarize data using graphs and charts.
4. Apply statistical techniques to derive insights from data.
5. Develop basic analytics applications using real-time datasets.

Textbooks:

1. Wes McKinney, *Python for Data Analysis*, 2nd Edition, O'Reilly.
2. YashvantKanetkar, *Let Us Python*, BPB Publications.

Reference Books:

1. Joel Grus, *Data Science from Scratch: First Principles with Python*, O'Reilly.
2. Jake VanderPlas, *Python Data Science Handbook*, O'Reilly.

Ramesh Sharda, DursunDelen, Efraim Turban, *Analytics, Data Science, and Artificial Intelligence*, Pearson

Subject Code	Subject Name	L	T	P	C
MINORS	Foundations of Cyber Physical Systems	3	0	0	3

Course Objectives:

1. To introduce the fundamentals of Cyber Physical Systems (CPS) and their architecture.
2. To understand the integration of computation, networking, and physical processes.
3. To explore modeling, control, and real-time aspects of CPS.
4. To study sensors, actuators, embedded systems, and communication protocols used in CPS.
5. To analyze the role of CPS in modern applications like smart cities, healthcare, and autonomous systems.

UNIT I: Introduction to Cyber Physical Systems

Definition and scope of CPS, architecture and components, characteristics of CPS, examples and applications in real-time systems, comparison with traditional embedded systems, and introduction to the Internet of Things (IoT) in CPS.

UNIT II: Modeling and Control of CPS

Modeling of physical systems, continuous and discrete models, hybrid systems, finite state machines, timed automata, and statecharts. Basic concepts of feedback control systems and their role in CPS.

UNIT III: Sensors, Actuators and Embedded Systems

Overview of sensors and actuators used in CPS, analog and digital signal processing, embedded system hardware and software components, microcontrollers, real-time operating systems (RTOS), and basic I/O interfacing.

UNIT IV: Communication and Networking in CPS

Communication needs in CPS, networked control systems, protocols for CPS such as CAN, ZigBee, MQTT, and 6LoWPAN. Wireless sensor networks, cloud integration, latency, and bandwidth considerations in CPS communication.

UNIT V: Applications and Case Studies

Applications in smart grids, smart homes, healthcare monitoring, autonomous vehicles, and industrial automation. Case studies demonstrating CPS design, deployment, and performance evaluation. Introduction to security, safety, and privacy challenges in CPS.

Course Outcomes:

After completing this course, students will be able to:

1. Understand the core concepts and components of Cyber Physical Systems.
2. Model and analyze CPS using hybrid systems and state machines.
3. Apply embedded hardware and software principles in CPS design.
4. Evaluate CPS performance with respect to timing, safety, and reliability.
5. Identify and describe real-world CPS applications in various domains.

Textbooks:

1. Rajeev Alur, *Principles of Cyber-Physical Systems*, MIT Press.
2. Edward Ashford Lee, Sanjit A. Seshia, *Introduction to Embedded Systems – A Cyber-Physical Systems Approach*, 2nd Edition, MIT Press.

Reference Books:

1. Fei Hu, *Cyber-Physical Systems: Integrated Computing and Engineering Design*, CRC Press.
2. Dieter Uckelmann et al., *Architecting the Internet of Things*, Springer.
3. Joseph Yiu, *The Definitive Guide to ARM® Cortex®-M3 and Cortex®-M4 Processors*, Newnes.