Shri Vishnu Engineering College for Women (A), Bhimavaram

Department of Computer Science & Engineering

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

B.Tech. – III Year I Semester

S.No	Category	Title	L	T	P	C
1	Professional Core	Data Warehousing and Data Mining	3	0	0	3
2	Professional Core	Computer Networks	3	0	0	3
3	Professional Core	Formal Languages and Automata Theory	3	0	0	3
4	Professional Elective-I	1. Object Oriented Analysis and Design 2. Artificial Intelligence 3. Microprocessors & Microcontrollers 4. Quantum Computing 5. 12 week MOOC Swayam/ NPTEL course recommended by the BoS	3	0	0	3
5	Open Elective-I	OR Entrepreneurship Development & Venture Creation	3	0	0	3
6	Professional Core	Data Mining Lab	0	0	3	1.5
7	Professional Core	Computer Networks Lab	0	0	3	1.5
8	Skill Enhancement course	Full Stack development-2	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	3	4.5
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
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	C
1	Professional Core	Compiler Design	3	0	0	3
2	Professional Core	Cloud Computing	3	0	0	3
3	Professional Core	Cryptography & Network Security	3	0	0	3
4	Professional Elective-II	1. Software Testing Methodologies 2. Cyber Security 3. DevOps 4. Machine Learning 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. Big Data Analytics 5. Distributed Operating System 6. 12 week MOOC Swayam/NPTEL course recommended by the BoS	3	0	0	3
6	Open Elective – II		3	0	0	3
7	Professional Core	Cloud Computing Lab	0	0	3	1.5
8	Professional Core	Cryptography & Network Security 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
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
HC	Honors Course (Student may select from the 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

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 CSE

1.	Principles of Database Management Systems	3-0-3-4.5 (II-II)
2.	Principles of Software Engineering	3-0-0-3 (III-I)
3.	Advanced Data Structures & Algorithm Analysis	3-0-3-4.5 (III-II)
4.	Principles of Operating Systems	3-0-0-3 (IV-I)
	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.	Fundamentals of Object Oriented Programming	
9.	Discrete Mathematics for CS	
10.	Software Engineering	

COURSES OFFERED FOR HONORS DEGREE IN CSE

Note: ***To obtain Honor's degree, student needs to obtain 18 credits by successfully completing any of the following courses in the concern stream.***

1.	Social Network Analysis	12 Week 3 Credit Course, MOOCS
2.	Applied Linear Algebra in AI & ML	12 Week 3 Credit Course, MOOCS
3.	Design & Implementation of Human-Computer Interfaces – NPTEL MOOCS	
4.	Cryptography and Network Security	12 Week 3 Credit Course, MOOCS
5.	Privacy and Security in Online Social Media	12 Week 3 Credit Course, MOOCS
6.	Deep Learning for Natural Language Processing	- 12 Week 3 Credit Course, MOOCS
7.	Computer Vision	- 12 Week 3 Credit Course, MOOCS
8.	Applied Time-Series Analysis	12 Week 3 Credit Course, MOOCS
9.	Parallel Computer Architecture	12 Week 3 Credit Course, MOOCS
10.	Reinforcement Learning	12 Week 3 Credit Course, MOOCS
11.	GPU Architecture and Programming	12 Week 3 Credit Course, MOOCS
12.	Computational Complexity	12 Week 3 Credit Course, MOOCS
13.	Quantum Algorithms and Cryptography	12 Week 3 Credit Course, MOOCS
14.	Unmanned Arial Systems & Robotics	12 Week 3 Credit Course, MOOCS
15.	Prompt Engineering for Generative AI	(III - II)

1.	Computer Networks	3-0-0-3
2.	Artificial Intelligence	3-0-0-3
3.	Cyber Security	3-0-0-3
4.	Introduction to Data Science	3-0-3-4.5
5.	Data Warehousing and Data Mining	3-0-0-3
6.	Object Oriented Programming Through Java	3-0-3-4.5
7.	Cloud computing	3-0-0-3
8.	Graph Theory	3-0-0-3
9.	Data Analytics with Python	
10.	Foundations of Cryptography	

III Year I Semester	DATA WAREHOUSING & DATA MINING	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

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

Course Outcomes (COs):

CO1: Understand and apply basic data warehousing concepts like OLAP and data cubes for designing data warehouses.

CO2: Analyze data mining techniques, data types, and measures of similarity and dissimilarity.

CO3: Apply data preprocessing methods such as cleaning, integration, reduction, and transformation to prepare data for mining.

CO4: Implement and evaluate classification algorithms like decision trees and Bayesian classifiers.

CO5: Use clustering and association rule mining algorithms (like Apriori, FP-Growth, K-Means, DBSCAN) to find patterns and groups in data.

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 Description**CourseOutcomes:**s 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)

Mapping of COs to POs:

POs/ COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11	PS O1	PS O2
CO1	3	2	2	2	2	–	–	–	–	1	–	2	2
CO2	3	3	2	3	2	–	–	–	–	–	–	2	3
CO3	3	3	2	3	2	–	–	–	–	–	–	3	3
CO4	3	3	3	3	3	–	–	–	–	–	–	3	2
CO5	3	3	3	3	3	–	–	–	–	1	–	3	3

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.
3. Data Mining and Data Warehousing: Principles and Practical Techniques, Parteek Bhatia, Cambridge University Press (27 June 2019)

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
5. Modern Data Warehousing Mining and Visualization Core Concepts, MARAKAS, Pearson Publication (1 January 2012)

III Year I Semester	COMPUTER NETWORKS	L	T	P	C
		3	0	0	3

Course Objectives:

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

Course Outcomes (COs):

CO1: Explain basic concepts of computer networks, types, topologies, and compare OSI and TCP/IP models.
CO2: Use error detection, correction, flow control, and sliding window protocols in data link layer.
CO3: Analyze different media access methods like ALOHA and CSMA, and compare Ethernet standards.
CO4: Design routing and congestion control methods, and understand IPv4 and IPv6 protocols and addresses.
CO5: Describe transport and application layer protocols like TCP, UDP, DNS, HTTP, and email.

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.

Mapping of COs to POs:

POs/ COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11	PS O1	PS O2
CO1	3	2	2	–	1	–	–	–	–	1	–	2	1
CO2	3	3	3	2	2	–	–	–	–	–	–	3	2
CO3	3	3	2	2	2	–	–	–	–	–	–	3	2
CO4	3	3	3	3	2	–	–	–	–	–	–	3	3
CO5	3	2	2	2	2	–	–	–	–	1	–	2	3

Text Books:

1. Computer Networksm, Andrew S Tanenbaum, Fifth Edition. Pearson Education/PHI
2. Data Communications and Networks, Behrouz A. Forouzan, Fifth Edition TMH.
3. Computer Networks, Kurose, Pearson Education; 8th edition (31 May 2022)

References Books:

1. Data Communications and Networks- Achut S Godbole, AtulKahate
2. Computer Networks, Mayank Dave, CENGAGE.
3. Networking All-in-One for Dummies, Doug Lowe, Seventh, Wiley (1 January 2018)

III Year I Semester	FORMAL LANGUAGES AND AUTOMATA THEORY	L	T	P	C
		3	0	0	3

Course Objectives:

- To learn fundamentals of Regular and Context Free Grammars and Languages
- To understand the relation between Regular Language and Finite Automata and machines
- To learn how to design Automata's and machines as Acceptors, Verifiers and Translators
- To understand the relation between Contexts free Languages, PDA and TM
- To learn how to design PDA as acceptor and TM as Calculators

Course Outcomes (COs):

CO1: Build and analyze different types of finite automata (DFA, NFA, ε-NFA), and explain how they are equivalent and minimized, including Mealy and Moore machines.

CO2: Create regular expressions, show how they relate to finite automata, and use Pumping Lemma and closure properties to solve problems on regular languages.

CO3: Construct context-free grammars, analyze parse trees, and simplify grammars using standard forms like CNF and GNF.

CO4: Design pushdown automata for context-free languages and explain their equivalence with context-free grammars, including deterministic and non-deterministic types.

CO5: Design Turing machines for basic languages, explain different TM types, and understand decidability, undecidability, and complexity classes like NP and NP-Complete.

UNIT I

Finite Automata: Need of Automata theory, Central Concepts of Automata Theory, Automation, Finite Automation, Transition Systems, Acceptance of a String, DFA, Design of DFAs, NFA, Design of NFA, Equivalence of DFA and NFA, Conversion of NFA into DFA, Finite Automata with Є-Transitions, Minimization of Finite Automata, Finite Automata with output-Mealy and Moore Machines, Applications and Limitation of Finite Automata.

UNIT II

Regular Expressions, Regular Sets, Identity Rules, Equivalence of two RE, Manipulations of REs, Finite Automata and Regular Expressions, Inter Conversion, Equivalence between FA and RE, Pumping Lemma of Regular Sets, Closure Properties of Regular Sets, Grammars, Classification of Grammars, Chomsky Hierarchy Theorem, Right and Left Linear Regular Grammars, Equivalence between RG and FA, Inter Conversion.

UNIT III

Formal Languages, Context Free Grammar, Leftmost and Rightmost Derivations, Parse Trees, Ambiguous Grammars, Simplification of Context Free Grammars-Elimination of Useless Symbols, Є-Productions and Unit Productions, Normal Forms-Chomsky Normal Form and Greibach Normal Form, Pumping Lemma, Closure Properties, Applications of Context Free Grammars.

UNIT IV

Pushdown Automata, Definition, Model, Graphical Notation, Instantaneous Description, Language Acceptance of Pushdown Automata, Design of Pushdown Automata, Deterministic and Non – Deterministic Pushdown Automata, Equivalence of Pushdown Automata and Context Free Grammars, Conversion, Two Stack Pushdown Automata, Application of Pushdown Automata.

UNIT V

Turning Machine: Definition, Model, Representation of TMs-Instantaneous Descriptions, Transition Tables and Transition Diagrams, Language of a TM, Design of TMs, Types of TMs, Church's Thesis, Universal and Restricted TM, Decidable and Un-decidable Problems, Halting Problem of TMs, Post's Correspondence Problem, Modified PCP, Classes of P and NP, NP-Hard and NP-Complete Problems.

Mapping of COs to POs:

POs/ COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11	PS O1	PS O2
CO1	3	3	2	2	2	–	–	–	–	1	–	3	2
CO2	3	3	2	3	2	–	–	–	–	–	–	3	2
CO3	3	3	2	3	1	–	–	–	–	–	–	3	1
CO4	3	3	2	3	2	–	–	–	–	–	–	3	2
CO5	3	3	3	3	2	–	–	–	–	1	–	3	3

Text Books:

1. Introduction to Automata Theory, Languages and Computation, J. E. Hopcroft, R. Motwani and J. D. Ullman, 3rd Edition, Pearson, 2008
2. Theory of Computer Science-Automata, Languages and Computation, K. L. P. Mishra and N. Chandrasekharan, 3rd Edition, PHI, 2007.
3. Modern Compiler Design, Dick Grune, Kees van Reeuwijk, Henri E. Bal, Ceriel J.H. Jacobs, Springer, 2016.

Reference Books:

1. Elements of Theory of Computation, Lewis H.P. & Papadimition C.H., Pearson /PHI
2. Theory of Computation, V. Kulkarni, Oxford University Press, 2013
3. Theory of Automata, Languages and Computation, Rajendra kumar, McGraw Hill, 2014
4. THEORY OF COMPUTATION Paperback – 1 January 2024 by Shripriti Publications

e-Resources:

1) https://nptel.ac.in/courses/106/104/106104028/

III Year I Semester	OBJECT ORIENTED ANALYSIS AND DESIGN	L	T	P	C
		3	0	0	3

Course Objectives: The main objective is the students to

- Become familiar with all phases of OOAD.
- Master the main features of the UML.
- 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.
- Learn the Object design Principles and understand how to apply them towards Implementation.

Course Outcomes (COs):

CO1: Explain what makes software complex and show how to design solutions using object-oriented ideas.
CO2: Describe why modeling is important and explain UML and its role in software development.
CO3: Create class and object diagrams in UML and build advanced models with interfaces and packages.
CO4: Model system behavior using use case, interaction, and activity diagrams.
CO5:D esign advanced behavior and architecture models like state machines and deployment diagrams for real systems.

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

Mapping of COs to POs:

POs/ COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11	PS O1	PS O2
CO1	3	2	3	2	2	–	–	–	–	1	–	3	2
CO2	3	2	2	2	2	–	–	–	–	1	–	2	1
CO3	3	3	3	2	3	–	–	–	–	1	–	3	2
CO4	3	3	3	2	3	–	–	–	–	1	–	3	2
CO5	3	2	3	3	3	–	–	–	–	1	–	3	2

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.
3. Object Oriented Analysis &Design With Uml, Dr. Shivani Joshi, S.Chand (G/L) & Company Ltd; Ist edition (1 July 2014)

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.
5. Object Oriented Analysis & Design, F. Margret Sharmila, Charulatha Publications Private Limited (1 January 2019)

III Year I Semester	ARTIFICIAL INTELLIGENCE	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

Course Outcomes (COs):

CO1: Explain basic AI concepts like intelligent agents and how to frame problems for AI.

CO2: Use search methods like BFS, DFS, and A* to solve problems and analyze game-playing strategies like Alpha-Beta pruning.

CO3: Show how to represent knowledge using logic and rules, and reason under uncertainty with probabilities.

CO4: Use logic-based reasoning and compare different machine learning methods like decision trees and reinforcement learning.

CO5: Describe how expert systems work and evaluate examples like MYCIN and XCON.

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.

Mapping of COs to POs:

POs/ COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11	PS O1	PS O2
CO1	3	2	2	2	1	–	–	–	–	1	–	2	2
CO2	3	3	3	3	2	–	–	–	–	–	–	3	3
CO3	3	3	2	3	2	–	–	–	–	1	–	3	3
CO4	3	3	3	3	2	–	–	–	–	–	–	3	3
CO5	3	2	2	2	1	–	–	–	–	1	–	2	2

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
3. AI for Everyone: A Beginner's Handbook for Artificial Intelligence, Saptarsi Goswami, Amit Kumar Das, Amlan Chakrabarti, Pearson, 2024

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.
5. Artificial Intelligence For Dummies, John Paul Mueller, Luca Massaron,, Stephanie Diamond, 3rd Edition, Wiley Publications, 2025.

Online Learning Resources:

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

III Year I Semester	MICROPROCESSORS & MICROCONTROLLERS	L	T	P	C
		3	0	0	3

Course Objectives:

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

Course Outcomes (COs):

CO1: Describe the structure and working of the 8086 microprocessor, including its parts and how it handles interrupts.

CO2: Write simple assembly language programs for the 8086 using different instructions and addressing methods.

CO3: Explain how to connect the 8086 to memory and devices like RAM, ROM, and communication interfaces, and describe how they work.

CO4:D escribe the architecture, special registers, I/O ports, and instructions of the 8051 microcontroller.

CO5: Create programs for 8051 microcontroller to control timers, communication, interrupts, and devices like LCD, keyboard, and sensors; and compare it with other processors like PIC and ARM.

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

Mapping of COs to POs:

POs/ COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11	PS O1	PS O2
CO1	3	2	2	2	2	–	–	–	–	1	–	2	1
CO2	3	3	3	2	3	–	–	–	–	–	–	3	2
CO3	3	2	3	3	3	–	–	–	–	1	–	3	2
CO4	3	2	2	2	2	–	–	–	–	–	–	2	1
CO5	3	3	3	3	3	–	–	–	–	1	–	3	2

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, 3rd edition, McGraw Hill Education, 2017.
3. Raj Kamal, Microcontrollers: Architecture, Programming, Interfacing and System Design, 2nd edition, Pearson, 2012.
4. Microprocessors & Microcontrollers, Santanu Chattopadhyay , McGraw Hill; First Edition, 2024

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.
3. Fundamentals of Microprocessors & Microcontrollers, B. Ram , Sanjay Kumar, Dhanpat Rai Publications (P) Ltd.; Latest Edition, 2021

III Year I Semester	QUANTUM COMPUTING	L	T	P	C
		3	0	0	3

Course Objectives:

To introduce the fundamentals of quantum computing, the problem-solving approach using finite dimensional mathematics

Course Outcomes (COs):

CO1:Understand the basics of quantum computing, including how qubits differ from classical bits and why math, physics, and biology matter in this field.

CO2: Use important math and physics ideas like superposition and entanglement to describe how quantum systems work.

CO3: Explain how qubits are built and show how to create simple quantum circuits using quantum gates.

CO4: Describe important quantum algorithms like Shor’s and Grover’s and know how they compare to classical algorithms.

CO5: Understand how errors happen in quantum computers, learn how to fix them, and explain applications like quantum cryptography and teleportation.

UNIT - I

History of Quantum Computing: Importance of Mathematics, Physics and Biology. Introduction to Quantum Computing: Bits Vs Qubits, Classical Vs Quantum logical operations

UNIT - II

Background Mathematics: Basics of Linear Algebra, Hilbert space, Probabilities and measurements.

Background Physics: Paul's exclusion Principle, Superposition, Entanglement and super-symmetry, density operators and correlation, basics of quantum mechanics, Measurements in bases other than computational basis. Background Biology: Basic concepts of Genomics and Proteomics (Central Dogma)

UNIT - III

Qubit: Physical implementations of Qubit. Qubit as a quantum unit of information. The Bloch sphere Quantum Circuits: single qubit gates, multiple qubit gates, designing the quantum circuits. Bell states.

UNIT - IV

Quantum Algorithms: Classical computation on quantum computers. Relationship between quantum and classical complexity classes. Deutsch’s algorithm, Deutsch’s-Jozsa algorithm, Shor’s factorization algorithm, Grover’s search algorithm.

UNIT - V

Noise and error correction: Graph states and codes, Quantum error correction, fault-tolerant computation. Quantum Information and Cryptography: Comparison between classical and quantum information theory. Quantum Cryptography, Quantum teleportation

Mapping of COs to POs:

POs/ COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11	PS O1	PS O2
CO1	3	2	2	2	2	–	2	–	–	1	–	2	2
CO2	3	3	2	3	2	–	2	–	–	–	–	2	2
CO3	3	2	3	2	3	–	1	–	–	1	–	3	2
CO4	3	3	3	3	3	–	–	–	–	1	–	3	3
CO5	3	3	2	3	3	–	2	–	–	1	–	3	3

Text Books:

1. Quantum Computation and Quantum Information, Nielsen M. A.,Cambridge
2. Programming Quantum Computers, Essential Algorithms and Code Samples, Eric R Johnson, Nic Harrigan, Mercedes Ginemo, Segovia, Oreilly
3. Quantum Computing: Applications and Challenges, Habiba Drias (Editor), Farouk Yalaoui, Springer Nature; 2024th edition (9 July 2024)

Reference Books:

1. Quantum Computing for Computer Scientists, Noson S. Yanofsk, Mirco A. Mannucci
2. Principles of Quantum Computation and Information,Benenti G., Casati G. and Strini G., Vol.I: Basic Concepts, Vol II
3. Basic Tools and Special Topics, World Scientific. Pittenger A. O., An Introduction to Quantum Computing Algorithms
4. Quantum Computing for Everyone, Chris Bernhardt, The MIT Press; Reprint edition (8 September 2020)

III Year I Semester	DATA MINING LAB	L	T	P	C
		0	0	3	1.5

Pre-requisites: Data Base Management Systems, Python Programming

Course Objectives: The main objective of the course is to

- Inculcate Conceptual, Logical, and Physical design of Data Warehouses OLAP applications and OLAP deployment
- Design a data warehouse or data mart to present information needed by management in a form that is usable
- Emphasize hands-on experience working with all real data sets.
- Test real data sets using popular data mining tools such as WEKA, Python Libraries
- Develop ability to design various algorithms based on data mining tools.

Course Outcomes (Cos):
CO1: Design and build data warehouses and data marts using ETL tools and apply OLAP operations like slice, dice, roll-up, and drill-down on real datasets.

CO2: Use the WEKA tool to explore datasets, perform data preprocessing, and understand data attributes, distributions, and visualizations.

CO3: Apply data mining techniques including association rule mining (Apriori), classification algorithms (ID3, J48, Naïve Bayes, k-NN), and evaluate their results using metrics and visualization.

CO4: Implement clustering algorithms such as k-means using WEKA or programming languages (Python/R/Java) and analyze cluster quality and visualizations.

CO5: Develop programs using Python, R, and Java to perform advanced data mining tasks like frequent itemset generation, chi-square calculation, dissimilarity matrix computation, and data visualization.

Software Requirements: WEKA Tool/Python/R-Tool/Rapid Tool/Oracle Data mining

List of Experiments:
1. Creation of a Data Warehouse.

- Build Data Warehouse/Data Mart (using open source tools like Pentaho Data Integration Tool, Pentaho Business Analytics; or other data warehouse tools like Microsoft-SSIS, Informatica, Business Objects, etc.,)
- Design multi-dimensional data models namely Star, Snowflake and Fact Constellation schemas for any one enterprise (ex. Banking, Insurance, Finance, Healthcare, manufacturing, Automobiles, sales etc).
- Write ETL scripts and implement using data warehouse tools.
- Perform Various OLAP operations such slice, dice, roll up, drill up and pivot

2. Explore machine learning tool "WEKA"

- Explore WEKA Data Mining/Machine Learning Toolkit.
- Downloading and/or installation of WEKA data mining toolkit.
- Understand the features of WEKA toolkit such as Explorer, Knowledge Flow interface, Experimenter, command-line interface.
- Navigate the options available in the WEKA (ex. Select attributes panel, Preprocess panel, Classify panel, Cluster panel, Associate panel and Visualize panel)
- Study the arff file format Explore the available data sets in WEKA. Load a data set (ex. Weather dataset, Iris dataset, etc.)
- Load each dataset and observe the following:
  1. List the attribute names and they types
  2. Number of records in each dataset
  3. Identify the class attribute (if any)
  4. Plot Histogram
  5. Determine the number of records for each class.
  6. Visualize the data in various dimensions

3. Perform data preprocessing tasks and Demonstrate performing association rule mining on data sets

- Explore various options available in Weka for preprocessing data and apply Unsupervised filters like Discretization, Resample filter, etc. on each dataset
- Load weather. nominal, Iris, Glass datasets into Weka and run Apriori Algorithm with different support and confidence values.
- Study the rules generated. Apply different discretization filters on numerical attributes and run the Apriori association rule algorithm. Study the rules generated.
- Derive interesting insights and observe the effect of discretization in the rule generation process.

4. Demonstrate performing classification on data sets Weka/R

- Load each dataset and run 1d3, J48 classification algorithm. Study the classifier output. Compute entropy values, Kappa statistic.
- Extract if-then rules from the decision tree generated by the classifier, Observe the confusion matrix.
- Load each dataset into Weka/R and perform Naïve-bayes classification and k-Nearest Neighbour classification. Interpret the results obtained.
- Plot RoC Curves
- Compare classification results of ID3, J48, Naïve-Bayes and k-NN classifiers for each dataset, and deduce which classifier is performing best and poor for each dataset and justify.

5. Demonstrate performing clustering of data sets

- Load each dataset into Weka/R and run simple k-means clustering algorithm with different values of k (number of desired clusters).

- Study the clusters formed. Observe the sum of squared errors and centroids, and derive insights.
- Explore other clustering techniques available in Weka/R.
- Explore visualization features of Weka/R to visualize the clusters. Derive interesting insights and explain.

6. Demonstrate knowledge flow application on data sets into Weka/R
    - Develop a knowledge flow layout for finding strong association rules by using Apriori, FP Growth algorithms
    - Set up the knowledge flow to load an ARFF (batch mode) and perform a cross validation using J48 algorithm
    - Demonstrate plotting multiple ROC curves in the same plot window by using j48 and Random forest tree
7. Demonstrate ZeroR technique on Iris dataset (by using necessary preprocessing technique(s)) and share your observations
8. Write a java program to prepare a simulated data set with unique instances.
9. Write a Python program to generate frequent item sets / association rules using Apriori algorithm
10. Write a program to calculate chi-square value using Python/R. Report your observation.
11. Write a program of Naive Bayesian classification using Python/R programming language.
12. Implement a Java/R program to perform Apriori algorithm
13. Write a R program to cluster your choice of data using simple k-means algorithm using JDK
14. Write a program of cluster analysis using simple k-means algorithm Python/R programming language.
15. Write a program to compute/display dissimilarity matrix (for your own dataset containing at least four instances with two attributes) using Python
16. Visualize the datasets using matplotlib in python/R.(Histogram, Box plot, Bar chart, Pie chart etc.,)

Mapping of COs to POs:

POs/ COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11	PS O1	PS O2
CO1	3	2	3	2	3	–	–	–	–	1	–	3	3
CO2	3	2	2	2	3	–	–	–	–	1	–	3	3
CO3	3	3	3	3	3	–	–	–	–	1	–	3	3
CO4	3	3	3	3	3	–	–	–	–	1	–	3	3
CO5	3	3	3	3	3	–	–	–	–	1	–	3	3

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

Course Outcomes (Cos):

CO1: Identify and connect various network devices to set up a Local Area Network (LAN) and understand its basic operation.

CO2: Implement and analyze data link layer protocols including framing methods (character stuffing, bit stuffing), error detection and correction techniques (checksum, Hamming code, CRC).

CO3: Develop and simulate data link layer flow control protocols such as Sliding Window (Go-Back-N, Selective Repeat) and Stop-and-Wait to ensure reliable data transmission.

CO4: Apply network layer algorithms for routing and congestion control, including Dijkstra's shortest path, distance vector routing, broadcast trees, and leaky bucket congestion control.

CO5: Use network analysis and simulation tools like Wireshark, Nmap, and NS2 to capture and analyze network traffic, detect operating systems, and simulate network behaviors including packet drops and throughput.

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. Wireshark
    - 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.

Mapping of COs to POs:

POs/ COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11	PS O1	PS O2
CO1	3	2	3	2	3	–	–	–	–	1	–	3	2
CO2	3	3	3	3	3	–	–	–	–	1	–	3	2
CO3	3	3	3	3	3	–	–	–	–	1	–	3	2
CO4	3	3	3	3	3	–	–	–	–	1	–	3	3
CO5	3	3	3	3	3	–	–	–	–	1	–	3	3

III Year I Semester	FULL STACK DEVELOPMENT - 2	L	T	P	C
		0	1	2	2

Course Objectives:

The main objectives of the course are to

- Make use of router, template engine and authentication using sessions to develop application in Express JS.
- Build a single page application using RESTful APIs in Express JS
- Apply router and hooks in designing React JS application
- Make use of MongoDB queries to perform CRUD operations on document database

Course Outcomes (Cos):

CO1: Develop web applications using Express JS by implementing routing, middleware, session management, authentication, and RESTful APIs.
CO2: Create interactive user interfaces with React JS by using JSX, components, props, state management, event handling, conditional rendering, lists, and forms.
CO3: Implement client-side routing and dynamic screen updates in React JS applications using React Router and state/hooks.
CO4: Install, configure, and perform CRUD operations on MongoDB databases and collections using MongoDB queries and Mongoose.
CO5: Design and build complete web applications by integrating Express JS, React JS, and MongoDB, such as to-do list and quiz apps.

Experiments covering the Topics:

- Express JS – Routing, HTTP Methods, Middleware, Templating, Form Data
- Express JS – Cookies, Sessions, Authentication, Database, RESTful APIs
- React JS – Render HTML, JSX, Components – function & Class, Props and States, Styles, Respond to Events
- React JS – Conditional Rendering, Rendering Lists, React Forms, React Router, Updating the Screen
- React JS – Hooks, Sharing data between Components, Applications – To-do list and Quiz
- MongoDB – Installation, Configuration, CRUD operations, Databases, Collections and Records

Sample Experiments:

1. Express JS – Routing, HTTP Methods, Middleware.

a. Write a program to define a route, Handling Routes, Route Parameters, Query Parameters and URL building.
b. Write a program to accept data, retrieve data and delete a specified resource using http methods.
c. Write a program to show the working of middleware.

2. Express JS – Templating, Form Data

a. Write a program using templating engine.
b. Write a program to work with form data.

3. Express JS – Cookies, Sessions, Authentication

a. Write a program for session management using cookies and sessions.
b. Write a program for user authentication.

4. Express JS – Database, RESTful APIs

a. Write a program to connect MongoDB database using Mangoose and perform CRUD operations.
b. Write a program to develop a single page application using RESTful APIs.

5. ReactJS – Render HTML, JSX, Components – function & Class

a. Write a program to render HTML to a web page.
b. Write a program for writing markup with JSX.
c. Write a program for creating and nesting components (function and class).
d.

6. ReactJS – Props and States, Styles, Respond to Events

a. Write a program to work with props and states.
b. Write a program to add styles (CSS & Sass Styling) and display data.
c. Write a program for responding to events.

7. ReactJS – Conditional Rendering, Rendering Lists, React Forms

a. Write a program for conditional rendering.
b. Write a program for rendering lists.
c. Write a program for working with different form fields using react forms.

8. ReactJS – React Router, Updating the Screen

a. Write a program for routing to different pages using react router.
b. Write a program for updating the screen.

9. ReactJS – Hooks, Sharing data between Components

a. Write a program to understand the importance of using hooks.
b. Write a program for sharing data between components.

10. MongoDB – Installation, Configuration, CRUD operations

a. Install MongoDB and configure ATLAS
b. Write MongoDB queries to perform CRUD operations on document using insert(), find(), update(), remove()

11. MongoDB – Databases, Collections and Records

a. Write MongoDB queries to Create and drop databases and collections.

b. Write MongoDB queries to work with records using find(), limit(), sort(), createIndex(), aggregate().

12. Augmented Programs: (Any 2 must be completed)

a. Design a to-do list application using NodeJS and ExpressJS.

b. Design a Quiz app using ReactJS.

c. Complete the MongoDB certification from MongoDB University website.

Mapping of COs to POs:

POs/ COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11	PS O1	PS O2
CO1	3	2	3	2	3	–	–	–	–	1	–	3	2
CO2	3	2	3	2	3	–	–	–	–	1	–	3	2
CO3	3	2	3	2	3	–	–	–	–	1	–	3	2
CO4	3	2	3	3	3	–	–	–	–	1	–	3	2
CO5	3	3	3	3	3	–	–	–	–	1	–	3	3

Text Books:

1. Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node, Vasan Subramanian, 2nd edition, APress, O'Reilly.
2. Node.Js in Action, Mike Cantelon, Mark Harter, T.J. Holowaychuk, Nathan Rajlich,Manning Publications. (Chapters 1-11)
3. React Quickly, AzatMardan,Manning Publications(Chapters 1-8,12-14)
4. Full Stack Development with Angular and Spring Boot: Build scalable, responsive, and dynamic enterprise-level web applications, Sangeeta Joshi, BPB Publications , 2024

Web Links:

1. ExpressJS - https://www.tutorialspoint.com/expressjs
2. ReactJS - https://www.w3schools.com/REACT (and) https://react.dev/learn#
3. MongoDB - https://learn.mongodb.com/learning-paths/introduction-to-mongodb
4. Full-Stack Web Development with FastAPI and React: Build Modern Python and JavaScript Applications. Deploy Modern Web Apps, Drake Duncan , 2025.

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

Course Outcomes (Cos):

CO1:Set up the Flutter environment and write basic Dart programs to understand language fundamentals.

CO2:Design responsive user interfaces using core Flutter widgets, layouts, and media queries.

CO3:Implement navigation, state management, and custom widgets to build interactive and dynamic mobile apps.

CO4:Develop forms with validation, apply animations, and fetch data from REST APIs to enhance user experience.

CO5:Test and debug Flutter applications using unit tests and built-in debugging tools to ensure code quality.

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.

Mapping of COs to POs:

POs/ COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11	PS O1	PS O2
CO1	3	2	3	2	3	–	–	–	–	1	–	3	2
CO2	3	2	3	2	3	–	–	–	–	1	–	3	2
CO3	3	3	3	3	3	–	–	–	–	1	–	3	3
CO4	3	3	3	3	3	–	–	–	–	1	–	3	3
CO5	3	3	3	3	3	–	–	–	–	1	–	3	3

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.
4. Flutter Solutions for Web Development: Modern web development with Flutter, Dart, and AI integration, Zaid Kamil, Bpb Publications, 2025.

III Year II Semester	COMPILER DESIGN	L	T	P	C
		3	0	0	3

Course Objectives:

Understand the basic concept of compiler design, and its different phases which will be helpful to construct new tools like LEX, YACC, etc.

Course Outcomes: Students will be able

- **CO1 :** To analyze the fundamental principles of lexical analysis and parsing, and apply this knowledge to design and implement front-end compiler components (scanners and parsers).
- **CO2 :** To practically apply syntax-directed translation techniques to generate intermediate code for various programming constructs.
- **CO3 :** To use different code optimization techniques to improve the performance (speed and memory) of compiled code.
- **CO4 :** To understand and be able to explain the core principles of how programs execute within run-time environments, including storage management and procedure calls.
- **CO5 :** To to design and build basic code generators, incorporating concepts like register allocation to translate intermediate code into executable machine code.

UNIT I:

Lexical Analysis: Language Processors, Structure of a Compiler, Lexical Analysis, The Role of the Lexical Analyzer, Bootstrapping, Input Buffering, Specification of Tokens, Recognition of Tokens, Lexical Analyzer Generator-LEX, Finite Automata, Regular Expressions and Finite Automata, Design of a Lexical Analyzer Generator.

Syntax Analysis: The Role of the Parser, Context-Free Grammars, Derivations, Parse Trees, Ambiguity, Left Recursion, Left Factoring,

UNIT II:

Top Down Parsing: Pre Processing Steps of Top Down Parsing, Backtracking, Recursive Descent Parsing, LL (1) Grammars, Non-recursive Predictive Parsing, Error Recovery in Predictive Parsing.

Bottom Up Parsing: Introduction, Difference between LR and LL Parsers, Types of LR Parsers, Shift Reduce Parsing, SLR Parsers, Construction of SLR Parsing Tables, More Powerful LR Parses, Construction of CLR (1) and LALR Parsing Tables, Dangling Else Ambiguity, Error Recovery in LR Parsing, Handling Ambiguity Grammar with LR Parsers.

UNIT III:

Syntax Directed Translation: Syntax-Directed Definitions, Evaluation Orders for SDD’s, Applications of Syntax Directed Translation, Syntax-Directed Translation Schemes, Implementing L-Attributed SDD’s. **Intermediate Code Generation:** Variants of Syntax Trees, Three Address Code, Types and Declarations, Translation of Expressions, Type Checking, Control Flow, Backpatching, Intermediate Code for Procedures.

UNIT IV:

Code Optimization: The Principle Sources of Optimization, Basic Blocks, Optimization of Basic Blocks, Structure Preserving Transformations, Flow Graphs, Loop Optimization, Data-Flow Analysis, Peephole Optimization

UNIT V:

Run Time Environments: Storage Organization, Run Time Storage Allocation, Activation Records, Procedure Calls, Displays

Code Generation: Issues in the Design of a Code Generator, Object Code Forms, Code Generation Algorithm, Register Allocation and Assignment.

Mapping of COs to POs:

POs/ COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11	PS O1	PS O2
CO1	3	3	3	3	3	–	–	–	–	1	–	3	2
CO2	3	3	3	3	3	–	–	–	–	1	–	3	2
CO3	3	3	3	3	3	–	–	–	–	–	–	3	2
CO4	3	2	2	2	2	–	–	–	–	1	–	2	1
CO5	3	3	3	3	3	–	–	–	–	1	–	3	2

Text Books:

1. Compilers: Principles, Techniques and Tools, Second Edition, Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffry D. Ullman, Pearson, 2007.
2. Modern Compiler Design, Dick Grune, Kees van Reeuwijk, Henri E. Bal, Ceriel J.H. Jacobs, Springer, 2016

Reference Books:

1. Compiler Construction, Principles and Practice, Kenneth C Louden, Cengage Learning, 2006
2. Modern compiler implementation in C, Andrew W Appel, Revised edition, Cambridge University Press.
3. Optimizing Compilers for Modern Architectures, Randy Allen, Ken Kennedy, Morgan Kauffmann, 2001.
4. Levine, J.R., T. Mason and D. Brown, Lex and Yacc, edition, O'Reilly & Associates, 1990.
5. System Programming and Compiler Construction, Dr. Shweta A. Loonkar, TechKnowledge Publications; 2024th edition (30 November 2023);

III Year II Semester	CLOUD COMPUTING	L	T	P	C
		3	0	0	3

Course Objectives:

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

Course Outcomes:

- **CO1:** Students will be able to explain core cloud computing concepts, including its models and key characteristics, along with the foundational technologies like distributed computing and SOA that make it possible.
- **CO2:** Students will be able to evaluate different virtualization and container technologies (e.g., Docker, Kubernetes), understanding their pros, cons, and how they're used in the cloud.
- **CO3:** Students will be able to identify and discuss the key challenges in cloud computing, such as economics, security, and scalability.
- **CO4:** Students will be able to describe and differentiate advanced cloud topics like serverless computing, IoT integration, edge computing, and DevOps principles.
- **CO5:** Students will be able to design fundamental cloud-based solutions by applying their knowledge of cloud services, deployment models, and related technologies.

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.

Mapping of COs to POs:

POs/ COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11	PS O1	PS O2
CO1	3	2	2	2	3	–	–	–	–	1	–	2	3
CO2	3	3	3	3	3	–	–	–	–	1	–	3	3
CO3	3	3	2	2	2	2	2	–	–	1	–	2	3
CO4	3	3	2	3	3	–	2	–	–	1	–	2	3
CO5	3	3	3	3	3	–	–	–	–	1	–	3	3

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.
3. CLOUD COMPUTING: A HANDS-ON APPROACH, Arshdeep Bahga and Vijay Madisetti, The Orient Blackswan, 2014

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)
4. Mastering Cloud Computing” by Rajkumar Buyya, Christian Vecchiola, and Thamarai Selvi, Morgan Kaufmann Publishers, 2013.

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.

Course Outcomes:

- **CO1:** Analyze security basics, common attacks, and dive deep into how symmetric encryption (like AES) works, including its mathematical foundations.
- **CO2:** Apply asymmetric cryptosystems (like RSA and ECC) to build secure communication channels, understanding the math behind them.
- **CO3:** Apply hash functions and digital signatures, and understand how to manage cryptographic keys.
- **CO4:** Describe secure communications at various network layers using protocols like PGP, SSL/TLS, and IPSec.
- **CO5:** Analyze common system vulnerabilities (like viruses and buffer overflows) and apply solutions like firewalls and intrusion detection systems.

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.

Mapping of COs to POs:

POs/ COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11	PS O1	PS O2
CO1	3	3	2	2	3	–	–	–	–	1	–	3	2
CO2	3	3	3	3	3	–	–	–	–	1	–	3	3
CO3	3	3	3	3	3	–	–	–	–	1	–	3	3
CO4	3	3	3	3	3	–	2	–	–	1	–	3	3
CO5	3	3	3	3	3	3	2	–	–	1	–	3	3

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
4. PRACTICAL MATHEMATICAL CRYPTOGRAPHY, Kristian Gjosteen, AYLOR & FRANCIS NP EXCLUSIVE(CBS); 1st edition, 2022.

Reference Books:

1. Network Security and Cryptography, 1st Edition, Bernard Meneges, Cengage Learning,2018.
2. Cryptography: Theory and Practice, Douglas Robert Stinson, Chapman and Hall/CRC; 4th edition (14 August 2018)

III Year II Semester	SOFTWARE TESTING METHODOLOGIES	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.

Course Outcomes:

- **CO1:** Understand Software Testing Basics & Path Testing.
- **CO2:** Apply Flow-Based & Domain Testing**.**
- **CO3:** Analyze Logic & Regular Expressions for Testing.
- **CO4:** Evaluate Software Based on its States.
- **CO5:** Analyze Software with Graph Matrices & Use Testing 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).

Mapping of COs to POs:

POs/ COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11	PS O1	PS O2
CO1	3	3	2	2	3	–	–	–	–	1	–	3	2
CO2	3	3	3	3	3	–	–	–	–	1	–	3	2
CO3	3	3	3	3	3	–	–	–	–	1	–	3	2
CO4	3	3	3	3	3	–	–	–	–	1	–	3	2
CO5	3	3	3	3	3	–	–	–	–	1	–	3	2

Text Books:

1. Software Testing techniques - Baris Beizer, Dreamtech, second edition.
2. Software Testing Tools – Dr. K. V. K. K. Prasad, Dreamtech.
3.SOFTWARE TESTING METHODOLOGIES : EASY MADE, Dr. V. Umadevi, Notion Press; 1st edition, 2020

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.
6. Software Testing Techniques, Boris Beizer , Wiley India; Second edition, 2002.

III Year II Semester	CYBER SECURITY	L	T	P	C
		3	0	0	3

Course Objectives:

The aim of the course is to

- identify security risks and take preventive steps
- understand the forensics fundamentals
- understand the evidence capturing process
- understand the preservation of digital evidence

Course Outcomes:

- CO1: To identify and explain different types of cybercrime, including how they started and who commits them. You'll also grasp the security challenges posed by modern devices.
- CO2: To describe common tools and techniques cybercriminals use, like phishing, malware, and denial-of-service attacks.
- CO3: Analyze and perform basic cybercrime investigations, including collecting and preserving digital evidence and tracking online activity.
- CO4: Use Apply computer forensics tools to analyze systems and devices for evidence.
- CO5: Analyze the legal aspects of cybercrime, especially concerning Indian laws like the IT Act, and understand the consequences of not following them.

UNIT I: Introduction to Cybercrime: Introduction, Cybercrime: Definition and Origins of the Word, Cybercrime and Information Security, Cyber criminals, Classifications of Cybercrime, Cyberstalking, Cybercafe and Cybercrimes, Botnets. Attack Vector, Proliferation ofMobile and Wireless Devices, Security Challenges Posed by Mobile Devices, Attacks on Mobile/Cell Phones, Network and Computer Attacks.

UNIT II: Tools and Methods : Proxy Servers and Anonymizers, Phishing, Password Cracking, Keyloggers and Spywares, Virus and Worms, Trojan Horses and Backdoors, 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 Cybercrime Scenario in India, Consequences of Not Addressing the Weakness in Information Technology Act, Digital Signatures and the Indian IT Act, Amendments to the Indian IT Act, Cybercrime andPunishment, Cyberlaw, Technology and Students: Indian Scenario.

Mapping of COs to POs:

POs/ COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11	PS O1	PS O2
CO1	3	2	2	2	2	3	2	–	–	1	–	2	2
CO2	3	3	2	2	3	3	2	–	–	1	–	2	2
CO3	3	3	3	3	3	3	2	–	–	2	–	3	2
CO4	3	3	3	3	3	2	2	–	–	2	–	3	2
CO5	3	2	2	2	2	3	3	3	–	2	–	2	1

Text Books:

1. Sunit 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.
3. Introduction to Information Security and Cyber Laws, Surya Prakash Tripati, Ritendra Goel, Praveen Kumar Shukla, Dreamtech Press

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.
4. Cyber Forensics, Dejey, Murugan, First Edition, Oxford, 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 Commons BY-NC-SA.

III Year II Semester	DEVOPS	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.

Course Outcomes:

- CO1: Design and optimize DevOps workflows for efficient software development and delivery.
- CO2: Effectively manage code versions and ensure code quality using industry-standard tools.
- CO3: Implement and maintain automated continuous integration (CI) pipelines for builds and testing.
- CO4: Deploy applications using containers (Docker) and establish continuous delivery (CD) practices.
- CO5: Automate infrastructure and application deployments with configuration management and orchestration tools.

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.

Mapping of COs to POs:

POs/ COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11	PS O1	PS O2
CO1	3	3	3	3	3	–	–	–	2	1	–	3	3
CO2	3	3	3	2	3	–	–	–	2	1	–	3	3
CO3	3	3	3	3	3	–	–	–	2	1	–	3	3
CO4	3	3	3	3	3	–	–	–	2	1	–	3	3
CO5	3	3	3	3	3	–	–	–	2	1	–	3	3

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.
3. The DevOps Handbook: How to Create World-Class Agility, Reliability, & Security in Technology Organizations, Gene Kim, Jez Humble, Patrick Debois , John Willis, Nicole Forsgren, Shroff/IT Revolution, Second Edition, 2024.

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.
6. Real-World DevOps Practices, B. Thangaraju, Wiley (17 October 2024).

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.

Course Outcomes:

- **CO1:** Students will be able to articulate the evolution, core paradigms, and complete lifecycle of a machine learning project, from data acquisition to model prediction.
- **CO2:** Students will be able to select, implement, and evaluate appropriate nearest neighbor and decision tree algorithms to solve various classification and regression problems, analyzing their performance.
- **CO3:** Students will be able to analyze the underlying principles of linear discriminants (Perceptron, SVM, Logistic Regression, Linear Regression) and probabilistic classifiers (Naive Bayes), and implement them for diverse machine learning tasks, including handling non-linear separability using the kernel trick.
- **CO4:** Students will be able to design, configure, and train Multi-Layer Perceptrons (MLPs) using the backpropagation algorithm to solve complex non-linear problems, demonstrating an understanding of neural network architectures and learning processes.
- **CO5:** Students will be able to critically evaluate various unsupervised clustering algorithms (e.g., K-Means, Fuzzy C-Means, Agglomerative, Spectral) and select the most suitable technique for different data characteristics and application scenarios.

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.

Mapping of COs to POs:

POs/ COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11	PS O1	PS O2
CO1	3	2	3	2	3	–	–	–	1	1	–	2	3
CO2	3	3	3	3	3	–	–	–	1	1	–	3	3
CO3	3	3	3	3	3	–	–	–	1	1	–	3	3
CO4	3	3	3	3	3	–	–	–	1	1	–	3	3
CO5	3	3	3	3	3	–	–	–	1	1	–	3	3

Text Books:

1. "Machine Learning Theory and Practice", M N Murthy, V S Ananthanarayana, Universities Press (India), 2024
2. Machine Learning for Real World Applications, Dinesh K. Sharma, Springer Nature; 2024th edition (9 October 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.
4. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, O'Reilly, Third Edition, 2022.

III Year II Semester	SOFTWARE PROJECT MANAGEMENT	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

Course Outcomes:

- CO1: Students will be able to understand and compare traditional and modern software development methods, considering their economic impacts and project phases.
- CO2: Students will be able to design software using architectural models and manage projects effectively with various workflows and planning techniques.
- CO3: Students will be able to evaluate different project structures, use automation, and apply metrics to control and improve project performance.
- CO4: Students will be able to use Agile and Scrum methods to manage software projects and improve team agility.
- CO5: Students will be able to implement core DevOps principles, understand its pipeline and ecosystem, and strategize its adoption in projects.

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

Mapping of COs to POs:

POs/ COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11	PS O1	PS O2
CO1	3	3	2	2	2	2	–	–	2	1	2	2	2
CO2	3	3	3	3	3	–	–	–	2	1	2	2	2
CO3	3	3	3	3	3	–	–	–	2	1	2	2	2
CO4	2	3	3	3	3	–	–	–	3	2	2	3	2
C05	3	3	3	3	3	–	–	–	3	2	2	3	3

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.
4. Software Project Effort Estimation: Foundations and Best Practice Guidelines for Success, Adam Trendowicz, Springer-Nature New York Inc; Reprint edition (23 September 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.
6. New Age Software project Management: Navigating the Technological revolution, Harshad Acharya , Adhyyan Books (12 December 2023)

III Year II Semester	MOBILE ADHOC NETWORKS	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.

Course Outcomes:

- CO1: Analyze the key features, benefits, drawbacks, and real-world uses of Ad Hoc Wireless Networks (MANETs) compared to traditional cellular networks, and explain the main challenges and different MAC protocols.
- CO2: Evaluate and create effective routing and transport layer protocols for Ad Hoc Wireless Networks, including solutions for challenges like TCP over MANETs.
- CO3: Develop and apply strong security solutions for Ad Hoc Wireless Networks, covering common attacks, secure routing, and intrusion detection.
- CO4: Design and enhance WSNs by considering factors like energy use and sensor grouping, and choose appropriate communication protocols for data collection in various applications.
- CO5: Evaluate security aspects and key management in WSNs, and effectively use sensor hardware, operating systems, programming languages, and simulation tools for WSN development

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.

Mapping of COs to POs:

POs/ COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11	PS O1	PS O2
CO1	3	3	2	2	2	–	–	–	1	1	–	2	2
CO2	3	3	3	3	3	–	–	–	1	1	–	3	2
CO3	3	3	3	3	3	3	2	3	1	1	–	3	2
CO4	3	3	3	3	3	–	2	–	2	1	–	3	3
CO5	3	3	3	3	3	–	2	–	2	1	–	3	3

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.

3.Multimodal Biometric Security for Mobile Adhoc Network, Dr P Prabhusundhar (Author), Dr B Srinivasan, Bonfring Technology Solutions (1 January 2017).

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.
6. Trust Based Secure Routing in Mobile Adhoc Network, Sachi Joshi (Author), Upesh Patel, LAP Lambert Academic Publishing (26 December 2023)

III Year II Semester	NATURAL LANGUAGE PROCESSING	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.

Course Outcomes:

- CO1: Explain the fundamentals of Natural Language Processing (NLP) and apply basic techniques like regular expressions, finite-state automata, and methods for tokenization and spelling correction.
- CO2: Evaluate and use different N-gram models and Part-of-Speech (PoS) tagging methods, understanding models like Hidden Markov and Maximum Entropy.
- CO3: Create and analyze sentence structures using Context-Free Grammars (CFGs) and Dependency Grammars, and apply parsing algorithms to understand how sentences are built.
- CO4: Design systems to represent meaning using First-Order Logic and apply techniques for Word Sense Disambiguation (WSD) and measuring word similarity.
- CO5: Apply methods for discourse analysis and coreference resolution, and effectively use common NLP lexical resources and tools like WordNet and various corpora.

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).

Mapping of COs to POs:

POs/ COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11	PS O1	PS O2
CO1	3	3	2	2	3	–	–	–	1	1	–	3	3
CO2	3	3	3	3	3	–	–	–	1	1	–	3	3
CO3	3	3	3	3	3	–	–	–	1	1	–	3	3
CO4	3	3	3	3	3	–	–	–	1	1	–	3	3
CO5	3	3	3	3	3	–	–	–	1	1	–	3	3

Text Books:

1. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech, 2ndEdition, Daniel Jurafsky, James H. Martin - Pearson Publication,2014.
2. Natural Language Processing with Python, First Edition, Steven Bird, Ewan Klein and Edward Loper, OReilly Media,2009.
3. Mastering Natural Language Processing with Transformers, Rupesh Kumar Tipu, LAP Lambert Academic Publishing (24 June 2024)

Reference Books:

1. Language Processing with Java and Ling Pipe Cookbook, 1stEdition, Breck Baldwin, Atlantic Publisher, 2015.
2. Natural Language Processing with Java, 2ndEdition, 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, 3rdEdition, TanveerSiddiqui, U.S. Tiwary, Oxford University Press,2008.
5. Understanding Natural Language Processing, T V Geetha, Pearson Education (17 June 2024)

III Year II Semester	BIG DATA ANALYTICS	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

Course Outcomes:

- CO1: Understand what big data is, why it's important, and how it's used in various industries like marketing, healthcare, and finance, recognizing its key technologies.
- CO2: Design and use NoSQL databases like Cassandra, understanding how they store and manage data, including concepts like replication and consistency.
- CO3: Develop solutions using Hadoop's file system (HDFS) and MapReduce for large datasets, and use Hive for data queries and optimization.
- CO4: Utilize Apache Spark for faster data processing, understanding its core features like RDDs and Data Frames, and how to deploy Spark applications.
- CO5: Improve the performance of Spark applications and build systems that can process data in real-time using Structured Streaming and handle continuous data flows.

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.

Mapping of COs to POs:

POs/ COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11	PS O1	PS O2
CO1	3	2	2	2	3	1	–	–	1	1	–	2	3
CO2	3	3	3	3	3	–	–	–	1	1	–	3	3
CO3	3	3	3	3	3	–	–	–	1	1	–	3	3
CO4	3	3	3	3	3	–	–	–	1	1	–	3	3
C05	3	3	3	3	3	–	–	–	1	1	–	3	3

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
6. Big Data Management and Processing, Kuan-Ching Li, Hai Jiang, Albert Y. Zomaya, Chapman & Hall; 1st edition (2 April 2021)

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.

"Programming Pig", O'Reilley, Alan Gates, October 2011: First Edition.

5. Big Data Analytics with Applications in Insider Threat Detection, Bhavani Thuraisingham, Pallabi Parveen, Mohammad Mehedy Masud, Auerbach Publications; 1st edition (30 September 2020)

III Year II Semester	DISTRIBUTED OPERATING SYSTEM	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

Course Outcomes:

- CO1: Explain the core ideas of distributed computing, including its evolution and challenges, and apply message passing techniques for communication between systems.
- CO2: Create and assess how Remote Procedure Calls (RPCs) work, understanding how programs talk to each other across different computers.
- CO3: Apply concepts of Distributed Shared Memory (DSM) for sharing data across systems, and use synchronization methods to coordinate activities in a distributed environment.
- CO4: Design ways to manage computer resources and processes efficiently in a distributed setup, including load balancing and process migration.
- CO5: Evaluate and build distributed file systems that can handle failures and ensure data is always available and consistent.

Unit I:

Fundamentals:

What is 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.

Mapping of COs to POs:

POs/ COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11	PS O1	PS O2
CO1	3	3	2	2	3	–	–	–	1	1	–	3	3
CO2	3	3	3	3	3	–	–	–	1	1	–	3	3
CO3	3	3	3	3	3	–	–	–	1	1	–	3	3
CO4	3	3	3	3	3	1	–	–	1	1	–	3	3
CO5	3	3	3	3	3	1	–	–	1	1	–	3	3

Text books

1. Pradeep. K. Sinha: Distributed Operating Systems: Concepts and Design, PHI, 2007.
2.ADVANCED CONCEPTS IN OPERATING SYSTEMS, Mukesh Singhal (Author), Niranjan Shivaratri , McGraw Hill Education (1 July 2017)

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
4.Distributed Operating Systems: Concepts and Design, Sinha, Prentice Hall India Learning Private Limited; First Edition (1 January 1998)

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.

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

- CO1: Install and configure virtual machines with different operating systems, run basic programs, and transfer files between them.
- CO2: Create and set up virtual servers on cloud platforms like Amazon EC2 or OpenStack, including configuring basic security and web services.
- CO3: Build and deploy simple web applications using cloud services (like Google App Engine) and Docker containers.
- CO4: Set up a Hadoop cluster to process large datasets and use serverless functions for event-driven tasks.
- CO5: Use simulation tools (like CloudSim) to model and test different scheduling methods in a cloud computing scenario.

List of Experiments:

1. Lab on web services
2. Lab on IPC, messaging, publish/subscribe
3. Install Virtual Box/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 Virtual Box 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.

Mapping of COs to POs:

POs/ COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11	PS O1	PS O2
CO1	3	2	2	2	3	–	–	–	1	1	–	3	2
CO2	3	3	3	2	3	1	–	–	1	1	–	3	3
CO3	3	3	3	3	3	–	–	–	1	1	–	3	3
CO4	3	3	3	3	3	–	–	–	1	1	–	3	3
CO5	3	3	3	3	3	–	–	–	1	1	–	3	3

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.
3. CLOUD COMPUTING: A HANDS-ON APPROACH, Arshdeep Bahga and Vijay Madisetti, The Orient Blackswan, 2014

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/
6. Mastering Cloud Computing" by Rajkumar Buyya, Christian Vecchiola, and Thamarai Selvi, Morgan Kaufmann Publishers, 2013.

III Year II Semester	CRYPTOGRAPHY & NETWORK SECURITY LAB	L	T	P	C
		0	0	3	1.5

Course Objectives:

- To learn basic understanding of cryptography, how it has evolved, and some key encryption techniques used today.
- To understand and implement encryption and decryption using Ceaser Cipher, Substitution Cipher, Hill Cipher.

Course Outcomes:

- CO1: Design programs to perform simple data scrambling and unscrambling using basic operations and classic ciphers like Caesar and Substitution.
- CO2: Develop and test standard secret-key encryption methods such as DES, Blowfish, and AES.
- CO3: Implement programs for public-key encryption (like RSA) and demonstrate how two parties can securely share a secret key (Diffie-Hellman).
- CO4: Calculate Data Fingerprints: Compute a unique "fingerprint" (message digest) of text data using hashing algorithms like SHA-1.
- CO5: Develop programming tools and libraries to perform encryption and manage cryptographic keys.

List of Experiments:

1. Write a C program that contains a string (char pointer) with a value \Hello World'. The program should XOR each character in this string with 0 and displays the result.
2. Write a C program that contains a string (char pointer) with a value \Hello World'. The program should AND or and XOR each character in this string with 127 and display the result
3. Write a Java program to perform encryption and decryption using the following algorithms:
   a) Ceaser Cipher
   b) Substitution Cipher
   c) Hill Cipher
4. Write a Java program to implement the DES algorithm logic
5. Write a C/JAVA program to implement the BlowFish algorithm logic
6. Write a C/JAVA program to implement the Rijndael algorithm logic.
7. Using Java Cryptography, encrypt the text “Hello world” using BlowFish. Create your own key using Java key tool.
8. Write a Java program to implement RSA Algorithm
9. Implement the Diffie-Hellman Key Exchange mechanism using HTML and JavaScript. Consider the end user as one of the parties (Alice) and the JavaScript application as other party (bob).
10. Calculate the message digest of a text using the SHA-1 algorithm in JAVA.

Mapping of COs to POs:

POs/ COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11	PS O1	PS O2
CO1	3	2	2	2	3	–	–	–	1	1	–	3	2
CO2	3	3	3	3	3	–	–	–	1	1	–	3	2
CO3	3	3	3	3	3	–	–	–	1	1	–	3	3
CO4	3	3	2	3	3	–	–	–	1	1	–	3	2
CO5	3	3	3	3	3	1	–	–	2	1	–	3	2

III Year II Semester	SOFT SKILLS	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

Course Outcomes:

- CO1: Improve their analytical thinking and listening skills, and use both verbal and non-verbal communication effectively.
- CO2: Apply skills to manage anger, stress, and time, and demonstrate proper social and business etiquette.
- CO3: Use correct grammar and pronunciation, and write professional letters, emails, notes, and meeting minutes.
- CO4: Participate successfully in group discussions, prepare effective resumes, and perform well in interviews.
- CO5: Understand the importance of relationships, adapt to different communication styles, and foster positive interactions.

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

Mapping of COs to POs:

POs/ COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11	PS O1	PS O2
CO1	1	2	–	–	2	2	–	–	3	3	–	–	–
CO2	–	2	–	–	2	3	2	–	3	3	–	–	–
CO3	2	2	–	–	3	1	–	–	3	3	–	–	–
CO4	2	2	–	–	2	2	–	–	3	3	–	–	–
CO5	–	2	–	–	2	2	–	–	3	3	–	–	–

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.
3. Soft Skills, Punam Agarwal, Blue Rose Publishers (24 September 2020)

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.
3. PERSONALITY DEVELOPMENT AND SOFT SKILLS, Barun K. Mitra, Basundhara Mitra, Oxford University Press (27 February 2024)

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

Course Outcomes:

- **CO1:** Organize and start writing technical reports and meeting minutes, using clear language and correct grammar.
- **CO2:** Draft reports, add relevant images, and thoroughly edit them for good grammar, clear language, and easy understanding.
- **CO3:** Create short, accurate summaries of information and deliver effective spoken presentations of technical reports.
- **CO4:** Efficiently use advanced features in word processing software (like creating tables of contents, tracking changes, and adding citations).
- **CO5:** Explain different types of intellectual property (like patents and copyrights) and understand how they are obtained and managed globally.

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

Mapping of COs to POs:

POs/ COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11	PS O1	PS O2
CO1	2	2	–	–	2	–	–	–	2	3	–	–	–
CO2	2	2	–	–	3	–	–	–	2	3	–	–	–
CO3	2	2	–	–	2	–	–	–	3	3	–	–	–
CO4	2	2	–	–	3	–	–	–	2	3	–	1	–
CO5	2	2	–	–	2	2	2	2	–	2	3	–	–

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.
4. Technical Writing: Processes & Products, Gerson / Gerson , Pearson Education India; 8th edition (1 January 2014); Pearson India

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).
3. Law Relating to Intellectual Property Rights (IPR) by MK Bhandari, Central law publication, 2024.

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/