COURSE STRUCTURE
&
DETAILED SYLLABUS
(R25 Regulation)

For
M. TECH - COMPUTER SCIENCE & ENGINEERING (Course Code-58)
I & II M.Tech (CSE)
(Applicable for Batches Admitted from 2025-2026)

Department of
COMPUTER SCIENCE & ENGINEERING
(Applicable for Batches Admitted from 2025-2026)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
JNTU-GV, COLLEGE OF ENGINEERING, VIZIANAGARAM (A)
JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY- GURAJADA VIZIANAGARAM
VIZIANAGARAM - 535003, ANDHRA PRADESH, INDIA

COURSE STRUCTURE

I Year – I SEMESTER

S.No.	Course Code	Course Title	L	T	P	C
1		**Program Core–1** Data Structures and Algorithm Analysis	3	1	0	4
2		**Program Core–2** Artificial Intelligence & Machine Learning	3	1	0	4
3		**Program Core–3** Mathematical Foundations of Computer Science	3	1	0	4
4		**Professional Elective –I**	3	0	0	3
5		**Professional Elective –II**	3	0	0	3
6		**Laboratory–1** Data Structures and Algorithm Analysis lab	0	1	2	2
7		**Laboratory–2** Artificial Intelligence & Machine Learning Lab	0	1	2	2
8		Seminar-I	0	0	2	1
		TOTAL	**15**	**5**	**6**	**23**

List of Professional Elective Courses in I Semester (Electives–I)

S. No.	Course Code	Course Title
1		Advanced Computer Networks
2		Recommender Systems
3		Mean Stack Technologies
4		Computer Organization and Operating Systems
5		Any minimum12 weeks MOOCS/NPTEL courses suggested by BOS

@Students can opt any one course from the above list

List of Professional Elective Courses in I Semester (Electives–II)

S. No.	Course Code	Course Title
1		Cryptography & Network Security
2		Digital Image Processing
3		Object Oriented Software Engineering
4		High Performance Computing
5		Any minimum12 weeks MOOCS/NPTEL courses suggested by BOS

@Students can opt any one course from the above list

I Year – II SEMESTER

Sl. No.	Course Code	Course Title	L	T	P	C
1		**Program Core–4** Deep Learning	3	1	0	4
2		**Program Core–5** Block Chain Technologies	3	1	0	4
3		**Program Core–6** Introduction to Quantum computing	3	1	0	4
4		**Professional Elective –III**	3	0	0	3
5		**Professional Elective –IV**	3	0	0	3
6		**Laboratory–3** Deep Learning lab	0	1	2	2
7		**Laboratory–4** Quantum computing lab	0	1	2	2
8		Seminar–II	0	0	2	1
		TOTAL	**15**	**5**	**6**	**23**

List of Professional Elective Courses in II Semester (Electives III)

S. No.	Course Code	Course Title
1		Soft computing
2		Internet of Things
3		Social Network Analysis
4		DevOps
5		Any minimum12 weeks MOOCS/NPTEL courses suggested by BOS

@Students can opt any one course from the above list

List of Professional Elective Courses in II Semester (Electives IV)

S. No.	Course Code	Course Title
1		Natural Language processing
2		Generative AI
3		Cyber Security
4		Design Patterns
5		Any minimum12 weeks MOOCS/NPTEL courses suggested by BOS

@Students can opt any one course from the above list

II Year – I SEMESTER

Sl. No.	Course Code	Course Title	L	T	P	C
1	O255801	Research Methodology and IPR / *Swayam 12 week MOOC course – RM&IPR*	3	0	0	3
2	O255802	Summer Internship/ Industrial Training (8-10 weeks)*	-	-	-	3
3	O255803	Comprehensive Viva[#]	-	-	-	2
4	O255804	Dissertation Part – A[$]	-	-	20	10
		TOTAL	**3**	**-**	**20**	**18**

* Student attended during summer / year break and assessment will be done in 3rd Semester.

Comprehensive viva can be conducted courses completed up to second semester.

$ Dissertation – Part A, internal assessment

II Year – II SEMESTER

Sl. No.	Course Code	Course Title	L	T	P	C
1	P255801	Dissertation Part – B[%]	-	-	32	16
		TOTAL	**-**	**-**	**32**	**16**

% External Assessment

I Year-I Semester

L	T	P	C
3	1	0	4

DATA STRUCTURES AND ALGORITHM ANALYSIS

Course Objectives:

The main objective of the course is to

1. Provide an understanding of algorithm design techniques
2. Enhance problem-solving skills by applying appropriate data structures and algorithms to computational problems

Course Outcomes:

At the end of the course, student will be able to

1. Design effective algorithms based on Divide and Conquer and Greedy methods.
2. Discuss various problems suitable to Dynamic programming.
3. Demonstrate various searching, sorting and hash techniques and be able to apply and solve problems of real life.
4. Design and implement variety of data structures including linked lists, binary trees, heaps, graphs and search trees.
5. Ability to compare various search trees and find solutions for IT related problems.

UNIT – I:

Divide and Conquer: General Method, Binary Search, Merge sort, Quick sort, Strassen's Matrix Multiplication. The Greedy Method: The general Method, knapsack problem, minimum-cost spanning Trees, Job sequencing within deadlines, Single Source Shortest Path Algorithm.

UNIT – II:

Dynamic Programming: The general method, All pairs-shortest paths, Optimal Binary search tree, 0/1 knapsack, Traveling salesperson problem.

UNIT – III:

Dictionaries, ADT, The List ADT, Stack ADT, Queue ADT, Hash Table Representation, Hash Functions, Collision Resolution-Separate Chaining, Open Addressing- Linear Probing, Double Hashing

UNIT – IV:

Priority queues- Definition, ADT, Realising a Priority Queue Using Heaps, Definition, Insertion, Deletion, Search Trees- Binary Search Trees, Definition, ADT, Implementation, Operations- Searching, Insertion, Deletion

UNIT – V:

Search Trees- AVL Trees, Definition, Height of AVL Tree, Operations- Insertion, Deletion and Searching. Introduction to Red-Black and Splay Trees, B-Trees, Height of B-Tree, Insertion, Deletion and Searching, Comparison of Search Trees

Text Books:

1. Data Structures: A Pseudocode Approach with C, 2nd Edition, Richard F.Gilberg, Behrouz A. Forouzon, Cengage Learning, 2004
2. Data Structures, Algorithms and Applications in java, 2nd Edition, Sartaj Sahni, University Press/Orient BlackSwan, 2005
3. Ellis Horowitz, Sartaj Sahni and Sanguthevar Rajasekaran, "Fundamentals of Computer Algorithms", Second Edition, Universities Press, India, 2021.

Reference Books:

1. Data Structures And Algorithm Analysis, 2nd Edition, Mark Allen Weiss, Pearson, 2002
2. Data Structures And Algorithms in C++, 3rd Edition, Adam Drozdek, Cengage Learning, 2005
3. C and Data Structures: A Snap Shot Oriented Treatise Using Live Engineering Examples, 1st Edition, N. B. Venkateswarulu, E.V. Prasad, S Chand & Co, 2009
4. Classic Data Structures, 2nd Edition, Debasis Samantha, PHI Learning, 2009

I Year-I Semester

L	T	P	C
3	1	0	4

ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

Course Objectives

- To gain foundational proficiency in Artificial Intelligence and Machine Learning techniques.
- To understand heuristic and probabilistic approaches to intelligent systems.
- To build predictive and classification models using decision trees, neural networks, and clustering methods.
- To grasp essential tools and concepts of fuzzy logic and probabilistic reasoning.

Course Outcomes

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

1. Formulate problems suitable for AI and ML solutions.
2. Understand how AI agents make decisions and solve problems optimally under constraints.
3. Represent knowledge formally and reason logically or probabilistically.
4. Understand classical supervised learning models and performance evaluation.
5. Gain a broad understanding of advanced ML and AI paradigms, including neural, fuzzy, and unsupervised models.

UNIT I:

Introduction to Artificial Intelligence and Machine Learning: History and evolution of AI, Applications and impact of AI in various domains, Intelligent agents: structure, environment, and rationality, Difference between AI, ML, and Deep Learning. Types of learning: supervised, unsupervised, reinforcement. ML pipeline: data collection, preprocessing, feature engineering, model training, and evaluation

UNIT II:

Problem Solving and Search Techniques: Problem representation and state space, Uninformed (blind) search: BFS, DFS, DLS, Iterative Deepening. Informed (heuristic) search: Greedy, A*

algorithm. Constraint Satisfaction Problems (CSPs), Game playing: Minimax, Alpha-Beta pruning, two-player games.

UNIT III:

Knowledge Representation and Reasoning: Propositional and Predicate Logic. Inference rules, Resolution, Semantic Tableau. Knowledge representation structures: Semantic networks, Frames, Scripts. Conceptual dependency theory, Case grammars, Introduction to Probabilistic reasoning: Bayesian networks, certainty factor theory.

UNIT IV:

Supervised Machine Learning: Learning types and workflow recap. Decision Trees: entropy, Gini, overfitting, pruning, Random Forests and ensemble basics, Linear Regression and Logistic Regression, Bias–variance tradeoff and model evaluation (accuracy, precision, recall, F1, confusion matrix).

UNIT V:

Neural Networks, SVMs, and Unsupervised Learning: Perceptron and introduction to Neural Networks, MLPs and Backpropagation (conceptual). Support Vector Machines (Linear and Kernelized – overview), Clustering: K-means, Hierarchical, Fuzzy C-means, EM Clustering, Overview: Fuzzy logic, Reinforcement Learning, NLP basics. Tools: Scikit-learn, TensorFlow basics

Textbooks:

1. Stuart Russell & Peter Norvig, Artificial Intelligence: A Modern Approach, 2nd Edition, Pearson Education, 2003.
2. SarojKaushik, Artificial Intelligence, CENGAGE Learning,1st Edition, 2011 .

Reference Books:

1. Tom M. Mitchell, Machine Learning, 1st Edition, Tata McGraw-Hill (TMH), 1997.
2. Deepak Khemani, Artificial Intelligence, 1st Edition, Tata McGraw-Hill (TMH), 2013.
3. Peter Harrington, Machine Learning in Action, 1st Edition, Manning Publications / DreamTech Press, 2012.
4. M.N. Murthy, V.S. Ananthanarayana, Machine Learning: Theory and Practice, Universities Press (India), 1st Edition, 2024.

I Year-I Semester

L	T	P	C
3	1	0	4

MATHEMATICAL FOUNDATIONS OF COMPUTER SCIENCE

Course Objectives:

Students undergoing this course are expected to:

1. Equip students with the knowledge of mathematical reasoning and proof techniques for correctness of algorithms and program verification
2. Cultivate analytical skills to apply mathematical tools and models in problem solving, algorithm analysis, and computational complexity.

Course Outcomes:

After completion of the course, students should be able to:

1. Apply equivalence formulas, tautological implications in finding normal forms, and theory of Inference in Statement Calculus and predicates, and explain Mathematical Induction Principle and apply the same
2. Apply skill in equivalences and inference theory in Predicate Calculus
3. Explain the properties of relations, POSETS, LATTICES, functions and apply the same in solving the problems.
4. Apply the principles of counting and probability to solve elementary and advanced probability problems.
5. Identify the properties of graphs and related structures and solve the related problems

UNIT – I:

Mathematical logic: Fundamentals (statements and notations, connectives, Truth tables), Tautologies, Equivalence of formulas, Tautological implications, Normal forms, Theory of Inference.

UNIT – II:

Predicate Calculus: Predicate logic, statement functions, variables and quantifiers, free and bound variables. Inference Theory of the Predicate Calculus: Logical implication involving quantifiers, Statements with more than one variable.

UNIT – III:

Relations : Relations, Properties of Relations, Equivalence relations, partial orders, Lattices, properties of Lattices, Special types of Lattices (Proofs not required).

UNIT – IV:

Counting, Probability, Discrete random variable, Continuous random variable, Moment generating function, Markov's inequality, Chebyshev's inequality, The geometric and binomial distributions, The tail of the binomial distribution.

UNIT – V:

Graph Theory: Basic Concepts of Graphs, Matrix representation of graphs: Adjacency Matrices, Incidence Matrices, Isomorphic Graphs, Paths and Circuits, Eulerin& Hamiltonian graphs, Planar Graphs, Graph coloring

Text Books:

1. Trembly J.P. and Manohar.P, "Discrete Mathematical Structures with applications to computer science," Tata McGraw Hill, New Delhi, 2017
2. Kolman B, Busoy R.C, Ross S.C, Discrete Mathematical Structures, 5th Edition, Prentice Hall, 2004.
3. Probability and Statistics with Reliability, 2nd edition, K. Trivedi, Wiley, 2011

Reference Books:

1. J.L. Mott, A. Kandel, T.P Baker, "Discrete Mathematics for Computer Scientists and Mathematicians," Prentice Hall India, 2nd Edition 2015
2. Kenneth H. Rosen, "Discrete Mathematics and its Applications," Tata McGraw Hill, New Delhi, 7th edition, 2017
3. V. Krishnamurthy, "Combinatories: Theory and Applications", East-West Press.

4. Seymour Lipschutz, M.Lipson, “Discrete Mathematics” Tata Mc Graw Hill, 2005.
5. D. S. Chandrasekharaiah, “Mathematical Foundation of Computer Science” Prism Publications, 2009.

I Year-I Semester

L	T	P	C
3	1	0	4

ADVANCED COMPUTER NETWORKS

(Professional Elective –I)

Course Objectives:

Students undergoing this course are expected to:

1. Explore advanced concepts in network design and analysis,
2. Familiarize students with wireless, mobile, and sensor networks, including their unique challenges and protocols.

Course Outcomes:

After completion of the course, students should be able to:

1. Illustrate reference models with layers, protocols, and interfaces.
2. Describe routing algorithms, subnetting, and addressing in IPv4 and IPv6.
3. Analyze basic network protocols and their use in network design and implementation.
4. Describe concepts related to wireless networks such as WLANs, WiMAX, IEEE 802.11, cellular and satellite systems.
5. Describe emerging network trends such as MANETs and Wireless Sensor Networks (WSNs).

UNIT – I:

Network layer: Network Layer design issues: store-and forward packet switching, services provided transport layers, implementation connection less services, implementation connection oriented services, comparison of virtual – circuit and datagram subnets, Routing Algorithms-shortest path routing, flooding, distance vector routing, link state routing, Hierarchical routing, congestion control algorithms : Approaches to congestion control, Traffic aware routing, Admission control, Traffic throttling, choke Packets, Load shedding, Random early detection, Quality of Service, Application requirements, Traffic shaping, Leaky and Token buckets.

UNIT – II:

Internetworking and IP protocols: How networks differ, How networks can be connected, internetworking, tunneling, The network layer in the internet, IPV4 Protocol, IP addresses, Subnets, CIDR, classful and Special addressing, network address translation (NAT),IPV6 Address structure address space, IPV6 Advantages, packet format, extension Headers, Transition from IPV4 to IPV6 , Internet Control Protocols-IMCP, ARP, DHCP.

UNIT – III:

Transport Layer Protocols: Introduction, Services, Port numbers, User Datagram Protocol: User datagram, UDP services, UDP Applications, Transmission control Protocol: TCP services, TCP features, Segment, A TCP connection, State transition diagram, Windows in TCP, Flow control and error control, TCP Congestion control, TCP Timers, SCTP: SCTP services SCTP
features, packet format, An SCTP association, flow control, error control

UNIT – IV:

Wireless LANS: Introduction, Architectural comparison, Access control, The IEEE 802.11 Project: Architecture, MAC sub layer, Addressing Mechanism, Physical Layer, Bluetooth: Architecture, Bluetooth Layers Other Wireless Networks: WIMAX: Services, IEEE project 802.16, Layers in project 802.16, Cellular Telephony: Operations, First Generation (1G), Second Generation (2G), Third Generation (3G), Fourth Generation (4G), Satellite Networks: Operation, GEO Satellites, MEO satellites, LEO satellites

UNIT – V:

Emerging trends in Computer networks: Mobile computing: Motivation for mobile computing, Protocol stack issues in mobile computing environment, mobility issues in mobile computing, security issues in mobile networks, MOBILE Ad Hoc Networks: Applications of Ad Hoc Networks, Challenges and Issues in MANETS, MAC Layer Issues Routing Protocols in MANET, Transport Layer Issues, Ad hoc Network Security

Wireless Sensor Networks: WSN functioning, Operating system support in sensor devices, WSN characteristics, sensor network operation, Sensor Architecture: Cluster management, Wireless Mesh Networks: WMN design, Issues in WMNs, Computational Grids, Grid Features, Issues in Grid construction design, Grid design features,P2P Networks: Characteristics of P2P Networks,

Classification of P2P systems, Gnutella, BitTorrent, Session Initiation Protocol(SIP) , Characteristics and addressing, Components of SIP, SIP establishment, SIP security.

Text Books:

1. Computer networks, 4th edition, Andrew S Tanenbaum, Pearson, 2012
2. Behrouz A. Forouzan, Data Communications and Networking, 5th Edition, McGraw-Hill Education, 2013.
3. Jochen Schiller, Mobile Communications, 2nd Edition, Pearson Education, 2012.

Reference Books:

1. Computer networks, A system Approach, 5th ed, Larry L Peterson and Bruce S Davie, Elsevier-2012.
2. C. Siva Ram Murthy and B. S. Manoj, Ad Hoc Wireless Networks: Architectures and Protocols, Pearson Education, 2004.

I Year-I Semester

L	T	P	C
3	1	0	4

RECOMMENDER SYSTEMS

(Professional Elective –I)

Course Objectives:

Students undergoing this course are expected to:

- Introduce the fundamental concepts and types of recommender systems, including content-based, collaborative, and hybrid approaches.
- Provide knowledge of evaluation metrics and methodologies to assess the quality and effectiveness of recommender systems.

Course Outcomes:

After completion of the course, students should be able to:

1. Compare different types of Recommender Systems.
2. Understand various issues related to Recommender System development.
3. Design a recommender system for a given problem.
4. Relate data collected from a Recommender System to understand user preferences and/or behaviour.
5. Develop and manage recommender system knowledge bases to efficiently store and retrieve user and item information for personalized recommendations.

UNIT-I:

Introduction to Recommender Systems: What is Recommendation engine?, Need for recommender systems, Framework of recommendation systems, Domain, Purpose, Context, Personalization, how will you target your users?, Personalized vs.Non-Personalized, Semi/Segment - Personalized, Privacy, users data and trustworthiness.

Recommender Systems Function, Techniques, Recommender Systems and Human Computer Interaction, Conversational Systems, Visualization, Issues working with RSs data sets: The cold-start problem.

UNIT-II:

Collaborative filtering-based Recommender System: Understanding ratings and rating data, User-based nearest-neighbor recommendation: Similarity Function, User-Based Algorithms Item-based nearest neighbor recommendation: Similarity Function, Item-Based Algorithms, Further model-based and preprocessing-based approaches, Comparing User-Based and Item-Based recommendations, data drift and concept drift.

UNIT-III:

Content-based Recommender System: Architecture of Content-based Systems, Advantages and Drawbacks of Content-based Filtering, Content representation and content similarity, Item profiles, discovering features of data, obtaining item features from tags, representing item profiles, Learning User Profiles and Filtering, Similarity-based retrieval, Classification algorithms, Knowledge base recommendation: Knowledge representation and reasoning, constraint-based recommenders, Case based recommenders.

UNIT-IV:

Neighbourhood -based Recommendation Methods: Advantages of Neighbourhood Approaches, Neighbourhood-based Recommendation, User-based Rating Prediction, User-based Classification Regression Vs Classification, Item-based Recommendation, User-based Vs Item based Recommendation, Rating Normalization, Similarity Weight Computation, Neighbourhood Selection

UNIT-V:

Constraint-based Recommenders Development of Recommender Knowledge Bases, User Guidance in Recommendation Processes, Calculating Recommendations

Context-Aware Recommender Systems Trust Context in Recommender Systems, Modeling Contextual Information in Recommender Systems, Paradigms for Incorporating Context in Recommender Systems: Contextual Pre- Filtering, Contextual Post-Filtering, Contextual Modeling, Combining Multiple Approaches, Additional Issues in Context-Aware Recommender Systems.

Text Books:

1. Charu C. Aggarwal, Recommender Systems: The Textbook, Springer, 2016.

2. Dietmar Jannach, Markus Zanker, Alexander Felfernig, Gerhard Friedrich, Recommender Systems: An Introduction, Cambridge University Press, 2010.
3. Francesco Ricci, Lior Rokach, and Bracha Shapira (Eds.), Recommender Systems Handbook, Springer, 2nd Edition, 2015.

Reference Books:

1. Michael D. Ekstrand, John T. Riedl, Joseph A. Konstan, Collaborative Filtering Recommender Systems, Foundations and Trends® in Human–Computer Interaction, 2011.
2. Guy Shani and Asela Gunawardana, Evaluating Recommendation Systems, in Recommender Systems Handbook (Springer, 2011).
3. Prem Melville and Vikas Sindhwani, Recommender Systems, in Encyclopedia of Machine Learning, Springer, 2010.

I Year-I Semester

L	T	P	C
3	1	0	4

MEAN STACK TECHNOLOGIES

(Professional Elective –I)

Course Objectives:

Students undergoing this course are expected to:

1. Translate user requirements into the overall architecture and implementation of new systems and Manage Project and coordinate with the Client.
2. Writing optimized front end code HTML and JavaScript.
3. Monitor the performance of web applications & infrastructure and Troubleshooting web application with a fast and accurate a resolution
4. Design and implementation of Robust and Scalable Front End Applications.

Course Outcomes:

After completion of the course, students should be able to:

- Identify the Basic Concepts of Web & Markup Languages.
- Develop web Applications using Scripting Languages & Frameworks.
- Make use of Express JS and Node JS frameworks
- Illustrate the uses of web services concepts like restful, react js.
- Adapt to Deployment Techniques & Working with cloud platform.

UNIT - I:

Introduction to Web: Internet and World Wide Web, Domain name service, Protocols: HTTP, FTP, SMTP. Html5 concepts, CSS3, Anatomy of a web page. XML: Document type Definition, XML schemas, Document object model, XSLT, DOM and SAX Approaches.

UNIT - II:

JavaScript: The Basic of JavaScript: Objects, Primitives Operations and Expressions, Control Statements, Arrays, Functions, Constructors, Pattern Matching using Regular Expressions. Angular Java Script Angular JS Expressions: ARRAY, Objects, $eval, Strings, Angular JS Form Validation & Form Submission, Single Page Application development using Angular JS

UNIT - III:

Node.js: Introduction, Advantages, Node.js Process Model, Node JS Modules. Express.js: Introduction to Express Framework, Introduction to Nodejs , What is Nodejs, Getting Started with Express, Your first Express App, Express Routing, Implementing MVC in Express, Middleware, Using Template Engines, Error Handling , API Handling , Debugging, Developing Template Engines, Using Process Managers, Security & Deployment.

UNIT IV:

RESTful Web Services: Using the Uniform Interface, Designing URIs, Web Linking, Conditional Requests. React Js: Welcome to React, Obstacles and Roadblocks, React's Future, Keeping Up with the Changes, Working with the Files, Pure React, Page Setup, The Virtual DOM, React Elements, ReactDOM, Children, Constructing Elements with Data, React Components, DOM Rendering, Factories

UNIT - V:

Mongo DB: Introduction, Architecture, Features, Examples, Database Creation & Collection in Mongo DB. Deploying Applications: Web hosting & Domains, Deployment Using Cloud Platforms.

Text Books:

1. Jeffrey C. Jackson, Web Technologies: A Computer Science Perspective, Pearson Education, 2011.
2. Paul Deitel, Harvey Deitel, Abbey Deitel, Internet & World Wide Web – How to Program, 5th Edition, Pearson Education, 2018.
3. Brad Dayley, Learning AngularJS, O'Reilly Media, 2015.
4. Mario Casciaro and Luciano Mammino, Node.js Design Patterns, 3rd Edition, Packt Publishing, 2020.
5. Alex Banks and Eve Porcello, Learning React: Modern Patterns for Developing React Apps, 3rd Edition, O'Reilly Media, 2023.
6. Kristina Chodorow, MongoDB: The Definitive Guide, 3rd Edition, O'Reilly Media, 2019.

Reference Books:

1. Wes Bos, Fullstack Advanced React & GraphQL, 2020.

2. Ethan Brown, Web Development with Node and Express: Leveraging the JavaScript Stack, 2nd Edition, O'Reilly, 2019.
3. Boris Paskhaver, Modern JavaScript for the Impatient Developer, Addison-Wesley, 2021.
4. Adam Freeman, Pro AngularJS, Apress, 2014.
5. Leonard Richardson and Mike Amundsen, RESTful Web APIs, O'Reilly, 2013.

I Year-I Semester

L	T	P	C
3	1	0	4

COMPUTER ORGANIZATION AND OPERATING SYSTEMS

(Professional Elective –I)

Course Objectives:

Students undergoing this course are expected to:

- Provide a comprehensive understanding of computer organization and architectural design.
- Explore how operating systems efficiently manage and optimize hardware resources.
- Examine the interaction between hardware architecture and system software for performance enhancement.
- Enable students to analyze and design high-performance computing systems.
- Prepare learners for advanced research in computer systems and operating system design.

Course Outcomes:

After completion of the course, students should be able to:

1. Understand the basic structure and functioning of a computer system.
2. Design and analyze scheduling and synchronization algorithms considering system architecture.
3. Evaluate memory hierarchy, cache, and virtual memory management techniques for performance optimization.
4. Analyze I/O subsystem design, device management, and file system organization in operating systems.
5. Analyze advanced architectures and OS support for multiprocessing and virtualization.

UNIT-I:

Fundamentals of Computer Organization: Functional units of a computer system: CPU, memory, I/O, and control unit, Instruction codes, instruction cycle, addressing modes, and instruction set design, Arithmetic and logic micro-operations; hardware implementation of arithmetic operations, Control unit design: hardwired and microprogrammed approaches.

UNIT-II:

Process Management and CPU Scheduling: Process concept, states, PCB structure, and operations on processes, CPU scheduling algorithms: FCFS, SJF, Priority, Round Robin, Process synchronization and inter-process communication, Classical synchronization problems-Producer-Consumer, Dining Philosopher's problem, semaphores and monitors, Deadlocks: prevention, avoidance, detection, and recovery.

UNIT-III:

Memory System and Management: Memory hierarchy: main memory, cache memory, and secondary storage, Cache mapping techniques, replacement policies, and write strategies, Virtual memory concepts and address translation, Operating system memory management: paging, segmentation, and demand paging, Page replacement algorithms.

UNIT-IV:

Input–Output and File Systems: Disk structure and scheduling algorithms, File system implementation: allocation methods, directory structure, and access control, I/O subsystem: organization, interfaces, and communication techniques (programmed, interrupt-driven, DMA), I/O channels and processors, Operating system I/O management: buffering, device drivers.

UNIT-V:

Advanced Architectures and OS Design: Pipelining, instruction-level parallelism, and superscalar architectures, Multiprocessor and multicore architectures, Distributed and real-time operating systems.

Text Books:

1. M. Morris Mano & Rajib Mall, Computer System Architecture, 3rd Edition, Pearson Education, 2017.
2. Abraham Silberschatz, Peter B. Galvin, Greg Gagne, Operating System Concepts, 10th Edition, Wiley, 2021.

Reference Books:

1. John L. Hennessy, David A. Patterson, Christos Kozyrakis, Computer Architecture: A Quantitative Approach, 7th Edition, Morgan Kaufmann, 2025.

2. Andrew S. Tanenbaum, Herbert Bos, Modern Operating Systems, 5th Edition, Pearson Education, 2022.
3. Dhananjay M. Dhamdhere, Operating Systems: A Concept-Based Approach, 3rd Edition, McGraw Hill Education, 2017.

I Year-I Semester

L	T	P	C
3	1	0	4

CRYPTOGRAPHY & NETWORK SECURITY

(Professional Elective –II)

Course Objectives:

Students undergoing this course are expected to:

1. The concepts of classical encryption techniques and concepts of finite fields and number theory
2. Working principles and utilities of various cryptographic algorithms including secret key cryptography, hashes and message digests, and public key algorithms
3. Design issues and working principles of various authentication protocols and PKI standards

Course Outcomes:

After completion of the course, students should be able to:

1. Identify information security goals and acquire fundamental knowledge on the concepts of finite fields and number theory
2. Compare and apply different encryption and decryption techniques to solve problems related to confidentiality and authentication
3. Apply the knowledge of cryptographic checksums and evaluate the performance of different message digest algorithms for verifying the integrity of varying message sizes.
4. Apply different digital signature algorithms to achieve authentication and create secure applications
5. Apply network security basics, analyze different attacks on networks and evaluate the performance of security protocols like SSL, IPSec, and PGP

UNIT- I:

Introduction to Security: Security Attacks, Security Services, Security Mechanisms, Fundamental Security Design Principles, Attack Surfaces and Attack Trees, a Model for Network Security

Mathematics of Cryptography: Algebraic Structures (Groups, Rings, Fields and Galois Fields), Divisibility and the Division Algorithm, The Euclidean Algorithm, Modular Arithmetic, Prime

Numbers, Fermat's and Euler's Theorems, Testing for Primality, The Chinese Remainder Theorem, Discrete Logarithms

UNIT- II:

Classical Encryption Techniques: Symmetric Cipher Model, Substitution Techniques, Transposition Techniques, Rotor Machines, Steganography

Block Ciphers: Traditional Block Cipher Structure, The Data Encryption Standard, The Strength of DES, Block Cipher Design Principles, Advanced Encryption Standard, AES Structure, AES Transformation Functions, AES Key Expansion, Multiple Encryption and Triple DES, Block Cipher Modes of Operation

UNIT- III:

Public-Key Cryptography: Principles of Public-Key Cryptosystems, The RSA Algorithm, Diffie-Hellman Key Exchange, Elgamal Cryptographic System, Elliptic Curve Cryptography

Cryptographic Hash Functions: Applications of Cryptographic Hash Functions, Requirements and Security, Secure Hash Algorithm (SHA)

Message Authentication Codes: Requirements for Message Authentication Codes, HMAC, CMAC

UNIT- IV:

Digital Signatures: Digital Signatures, Elgamal Digital Signature Scheme, Schnorr Digital Signature Scheme, NIST Digital Signature Algorithm, Elliptic Curve Digital Signature Algorithm

Key Management and Distribution: Symmetric Key Distribution Using Symmetric Encryption, Symmetric Key Distribution Using Asymmetric Encryption, Distribution of Public Keys, X.509 Certificates, Public-Key Infrastructure

User Authentication: Remote User-Authentication Principles, Remote User-Authentication Using Symmetric Encryption, Kerberos, Remote User-Authentication Using Asymmetric Encryption:

UNIT -V:

Transport-Level Security: Web Security Considerations, Transport Layer Security, Secure Shell (SSH)

Electronic Mail Security: S/MIME, Pretty Good Privacy

IP Security: IP Security Overview, Encapsulating Security Payload, Combining Security Associations, Internet Key Exchange

Text Book:

1. Cryptography and Network Security, William Stallings, 8th Edition, 2022, Pearson Education

Reference Books:

1. Cryptography, Network Security and Cyber Laws, Bernard L. Menezes, Ravinder Kumar, Cengage Learning, 1st edition, 2018.
2. Cryptography and Network Security, Behrouz A Forouzan, Debdeep Mukhopadhyaya, 3rd Edition, 2015, Mc-GrawHill.
3. Network Security Illustrated, Jason Albanese, Wes Sonnenreich, 1st edition, 2003, McGrawHill.

E-Resources:

1. https://nptel.ac.in/courses/106/105/106105031/ lecture by Dr.Debdeep Mukhopadhyay IIT Kharagpur [VideoLecture]
2. https://nptel.ac.in/courses/106/105/106105162/lecture by Dr. Sourav Mukhopadhyay IIT Kharagpur [VideoLecture]
3. https://www.mitel.com/articles/web-communication-cryptography-and-network-securityweb articles by Mitel Power Connections

I Year-I Semester

L	T	P	C
3	1	0	4

DIGITAL IMAGE PROCESSING

(Professional Elective –II)

Course Objectives:

Students undergoing this course are expected to:

- Describe and explain basic principles of digital image processing.
- Design and implement algorithms that perform basic image processing (e.g. noise removal and image enhancement).
- Design and implement algorithms for advanced image analysis (e.g. image compression, image segmentation).
- Assess the performance of image processing algorithms and systems.

Course Outcomes:

After completion of the course, students should be able to:

- Demonstrate the components of image processing
- Explain various filtration techniques.
- Apply image compression techniques.
- Discuss the concepts of wavelet transforms.
- Analyze the concept of morphological image processing.

UNIT - I:

Introduction: Fundamental steps in Image Processing System, Components of Image Processing System, Elements of Visual Perception, Image Sensing and acquisition, Image sampling & Quantization, Basic Relationship between pixels. Image Enhancement Techniques: Spatial Domain Methods: Basic grey level transformation, Histogram equalization, Image subtraction, image averaging.

UNIT - II:

Spatial filtering: Smoothing, sharpening filters, Laplacian filters, Frequency domain filters, Smoothing and sharpening filters, Homomorphism is filtering. Image Restoration & Reconstruction:

Model of Image Degradation/restoration process, Noise models, Spatial filtering, Inverse filtering, Minimum mean square Error filtering, constrained least square filtering, Geometric mean filter, Image reconstruction from projections. Color Fundamentals, Color Models, Color Transformations.

UNIT - III: Image Compression: Redundancies- Coding, Interpixel, Psycho visual; Fidelity, Source and Channel Encoding, Elements of Information Theory; Loss Less and Lossy Compression; Run length coding, Differential encoding, DCT, Vector quantization, Entropy coding, LZW coding; Image Compression Standards-JPEG, JPEG 2000, MPEG; Video compression.

UNIT - IV:

Wavelet Based Image Compression: Expansion of functions, Multi-resolution analysis, Scaling functions, MRA refinement equation, Wavelet series expansion, Discrete Wavelet Transform (DWT), Continuous, Wavelet Transform, Fast Wavelet Transform, 2-D wavelet Transform, JPEG-2000 encoding.

UNIT - V:

Image Segmentation: Discontinuities, Edge Linking and boundary detection, Thresholding, Region Based Segmentation, Watersheds; Introduction to morphological operations; binary morphology-erosion, dilation, opening and closing operations, applications; basic gray-scale morphology operations; Feature extraction; Classification; Object recognition. Digital Image Watermarking: Introduction, need of Digital Image Watermarking, applications of watermarking in copyright protection and Image quality analysis.

Text Books:

1. Digital Image Processing. 4th ed. Gonzalez, R.C. and Woods, R.E. India: Person Education, (2018)

Reference Books:

1. Digital Image Processing. John Wiley, Pratt, W. K, (2001)
2. Digital Image Processing, Jayaraman, S., Veerakumar, T. and Esakkiranjan, S. (2009),Tata McGraw-Hill

I Year-I Semester

L	T	P	C
3	1	0	4

OBJECT ORIENTED SOFTWARE ENGINEERING

(Professional Elective –II)

Course Objectives:

Students undergoing this course are expected to:

1. Understand the fundamental principles, processes, and methodologies of software engineering with a focus on object-oriented paradigms and development life cycles.
2. Apply object-oriented concepts, Unified Modeling Language (UML), and design patterns to model, design, and implement efficient software systems.
3. Evaluate and manage software projects by employing suitable metrics, testing strategies, CASE tools, and project management techniques to ensure quality and reliability.

Course Outcomes:

After completion of the course, students should be able to:

1. Explain the fundamental concepts of software engineering, software process models, and their applicability to different project environments.
2. Apply object-oriented principles such as abstraction, encapsulation, inheritance, and polymorphism to analyze and model software systems.
3. Design object-oriented software solutions using UML models, design patterns, and object-oriented programming concepts.
4. Apply appropriate object-oriented testing strategies and design effective test cases for class and interclass testing.
5. Evaluate and measure object-oriented software quality using suitable metrics and employ CASE tools for project estimation and management.

UNIT - I:

Introduction to Software Engineering: Software, Software Crisis, Software Engineering definition, Evolution of Software Engineering Methodologies, Software Engineering Challenges. Software

Processes: Software Process, Process Classification, Phased development life cycle, Software Development Process Models, Process, use, applicability and Advantages/limitations.

UNIT - II:

Object oriented Paradigm, Object oriented Concepts, Classes, Objects, Attributes, Methods and services, Messages, Encapsulation, Inheritance, Polymorphism, Identifying the elements of object model, management of object oriented Software projects, Object Oriented Analysis, Domain Analysis, Generic Components of OOA model,OOA Process, Object Relationship model, Object Behavior Model.

UNIT - III:

Object Oriented Design: Design for Object- Oriented systems, The Generic components of the OO design model, The System design process, The Object design process, Design Patterns, Object Oriented Programming.

UNIT - IV:

Object Oriented testing: Broadening the view of Testing, Testing of OOA and OOD models, Object-Oriented testing strategies, Test case design for OO software, testing methods applicable at the class level, Interclass test case design.

UNIT - V:

Technical Metrics for Object Oriented Systems: The Intent of Object Oriented metrics, The distinguishing Characteristics, Metrics for the OO Design model, Class-Oriented metrics, Operation-Oriented Metrics, Metrics foe Object Oriented testing, Metrics for Object Oriented projects. CASE Tools.

Text Books:

1. Stephen R. Schach, Object-Oriented and Classical Software Engineering, 7th Edition, McGraw-Hill, 2006
2. Timothy Lethbridge & Robert Laganière, Object-Oriented Software Engineering: Practical Software Development Using UML and Java,
3. Roger S. Pressman & Bruce R. Maxim, Software Engineering: A Practitioner's Approach, 9th Edition, McGraw-Hill, 2020

Reference Books:

1. Ivica Crnkovic et al., Component-Based Software Engineering: 7th International Symposium (Springer, 2004)
2. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, 1994
3. Grady Booch, James Rumbaugh, Ivar Jacobson, The Unified Modeling Language User Guide, 2nd Edition, Addison-Wesley, 2005
4. Bertrand Meyer, Object-Oriented Software Construction, 2nd Edition, Prentice Hall, 1997

I Year-I Semester

L	T	P	C
3	1	0	4

HIGH PERFORMANCE COMPUTING

(Professional Elective –II)

Course Objectives:

Students undergoing this course are expected to:

- Develop an understanding of parallel programming models, such as shared-memory, distributed-memory, and hybrid models.
- Provide knowledge of HPC tools, libraries, and frameworks, such as MPI, OpenMP, CUDA, and GPU computing.

Course Outcomes:

After completion of the course, students should be able to:

1. Describe different parallel architectures, inter-connect networks, programming models
2. Develop an efficient parallel algorithm to solve given problem
3. Analyze and measure performance of modern parallel computing systems
4. Build the logic to parallelize the programming task
5. Evaluate the performance of parallel algorithms and CUDA implementations in terms of speedup, efficiency, and scalability.

UNIT – I:

Introduction: Motivating Parallelism, Scope of Parallel Computing, Parallel Programming Platforms: Implicit Parallelism, Trends in Microprocessor and Architectures, Limitations of Memory, System Performance, Dichotomy of Parallel Computing Platforms, Physical Organization of Parallel Platforms, Communication Costs in Parallel Machines, Scalable design principles, Architectures: N-wide superscalar architectures, Multi- core architecture.

UNIT – II:

Parallel Programming: Principles of Parallel Algorithm Design: Preliminaries, Decomposition Techniques, Characteristics of Tasks and Interactions, Mapping Techniques for Load Balancing,

Methods for Containing Interaction Overheads, Parallel Algorithm Models, The Age of Parallel Processing, the Rise of GPU Computing, A Brief History of GPUs, Early GPU.

UNIT – III:

Basic Communication: Operations- One-to-All Broadcast and All-to-One Reduction, All-to-All Broadcast and Reduction, All-Reduce and Prefix-Sum Operations, Scatter and Gather, All-to-All Personalized Communication, Circular Shift, Improving the Speed of Some Communication Operations. Programming shared address space platforms: threads- basics, synchronization, OpenMP programming

UNIT – IV:

Analytical Models: Sources of overhead in Parallel Programs, Performance Metrics for Parallel Systems, and The effect of Granularity on Performance, Scalability of Parallel Systems, Minimum execution time and minimum cost, optimal execution time. Dense Matrix Algorithms: Matrix Vector Multiplication, Matrix-Matrix Multiplication.

UNIT – V:

Parallel Algorithms- Sorting and Graph : Issues in Sorting on Parallel Computers, Bubble Sort and its Variants, Parallelizing Quick sort, All-Pairs Shortest Paths, Algorithm for sparse graph, Parallel Depth-First Search, Parallel Best First Search. CUDA Architecture: CUDA Architecture, Using the CUDA Architecture, Applications of CUDA Introduction to CUDA C-Write and launch CUDA C kernels, Manage GPU memory, Manage communication and synchronization, Parallel programming in CUDA- C.

Text Books:

1. Ananth Grama, Anshul Gupta, George Karypis, and Vipin Kumar, "Introduction to Parallel Computing", 2nd edition, Addison-Wesley, 2003, ISBN: 0-201-64865-2
2. Jason sanders, Edward Kandrot, “CUDA by Example”, Addison-Wesley, ISBN-13: 978-0-13-138768-3

Reference Books

1. Kai Hwang, ”Scalable Parallel Computing”, McGraw Hill 1998, ISBN:0070317984

3. Shane Cook, “CUDA Programming: A Developer's Guide to Parallel Computing with GPUs”, Morgan Kaufmann Publishers Inc. San Francisco, CA, USA 2013 ISBN: 9780124159884
4. David Culler Jaswinder Pal Singh, ”Parallel Computer Architecture: A Hardware/Software Approach”, Morgan Kaufmann,1999, ISBN 978-1-55860-343-1
5. Rod Stephens, “Essential Algorithms”, Wiley, ISBN: ISBN: 978-1-118-61210-1

I Year-I Semester

L	T	P	C
0	1	2	3

DATA STRUCTURES AND ALGORITHM ANALYSIS LAB

Course Objectives:

Students undergoing this course are expected to:

- Enhance understanding of algorithm analysis, including time and space complexity using asymptotic notations.
- Equip students with the skills to apply appropriate data structures and algorithms to solve real-world computational problems.

Course Outcomes:

After completion of the course, students should be able to:

1. Ability to write and analyze algorithms for algorithm correctness and efficiency
2. Master a variety of advanced abstract data type (ADT) and data structures and their Implementation.
3. Demonstrate various searching, sorting and hash techniques and be able to apply and solve problems of real life
4. Design and implement variety of data structures including linked lists, binary trees, heaps, graphs and search trees
5. Ability to compare various search trees and find solutions for IT related problems

Experiment– 1: Write a java program to perform various operations on single linked list

Experiment– 2: Write a java program for the following

a) Reverse a linked list

b) Sort the data in a linked list

c) Remove duplicates

d) Merge two linked lists

Experiment–3: Write a java program to perform various operations on doubly linked list

Experiment– 4: Write a java program to perform various operations on circular linked list

Experiment– 5: Write a java program for performing various operations on stack using linked list

Experiment– 6: Write a java program for performing various operations on queue using linked list

Experiment– 7: Write a java program for the following using stack

a) Infix to postfix conversion.

b) Expression evaluation.

c) Obtain the binary number for a given decimal number.

Experiment– 8: Write a java program to implement various operations on Binary Search Tree Using Recursive and Non-Recursive methods.

Experiment– 9: Write a java program to implement the following for a graph.

a) BFS b) DFS

Experiment– 10: Write a java program to implement Merge & Heap Sort of given elements

Experiment– 11: Write a java program to implement Quick Sort of given elements

Experiment– 12: Write a java program to implement various operations on AVL trees

Experiment– 13: Write a java program to perform the following operations:

a) Insertion into a B-tree b) Searching in a B-tree

Experiment– 14: Write a java program to implementation of recursive and non-recursive functions to Binary tree Traversals

Experiment– 15: Write a java program to implement all the functions of Dictionary (ADT) using Hashing

I Year-I Semester

L	T	P	C
0	1	2	3

ARTIFICIAL INTELLIGENCE & MACHINE LEARNING LAB

Course Objectives

- To practice AI and ML algorithms using Python and its libraries.
- To understand neural networks, clustering, and reinforcement learning through hands-on coding.
- To utilize TensorFlow, Scikit-learn, and Keras for model development.

Course Outcomes

1. Upon completion of this lab course, students will be able to:
2. Implement supervised and unsupervised learning models using Python.
3. Apply algorithms to solve classification, regression, and clustering problems.
4. Explore tools like TensorFlow and Keras for real-world applications.
5. Use NLP libraries for basic language tasks.

List of Experiments

PartA

1. Practice on Pandas, Numpy, matplot libraries for handling datasets
2. Implement
    a) Simple linear regression
    b) Multiple linear regression
    c) Polynomial regression
3. Implement decision tree algorithm for
    a) Binary classifier
    b) Multiclass classifier
4. Implement the single layer perceptron learning to learn the following Boolean functions
    a) AND
    b) OR

c) NOT

5. Implement multi-layer perceptron learning (i.e back propagation) to learn the following Boolean functions
    a) XOR
    b) XNOR
6. Implement Naïve Bayes.
7. Implement SVM.
8. Implement K-means clustering.

Additional Experiments:

9. Implement Random Forest.
10. Kears : Handwritten digit recognition using MNIST

Part B: AI Lab and NLP (TensorFlow, Keras, NLTK)

1. Implement A* and Greedy Best-First search on a 10×10 grid with obstacles. Compare nodes Expanded and path cost using Manhattan and Euclidean heuristics.
2. Build a Tic-Tac-Toe AI using mini-max and alpha–beta pruning.
   Compare nodes expanded and playing strength.
3. Use spaCy or NLTK to extract subject–verb–object triples from sentences and build a simple semantic network visualization using NetworkX.

Reference Books:

1. Peter Flach, Machine Learning: The Art and Science, 1st Edition, Cambridge University Press, 2012.
2. Chris Albon, ML with Python Cookbook, 1st Edition, O'Reilly Media, 2018.
3. NLTK & Scikit-learn documentation.

I Year-II Semester

L	T	P	C
3	1	0	4

DEEP LEARNING

Course Objectives:

Students undergoing this course are expected to:

- Understand and internalize deep learning methods for sequential data, deep recurrent/memory networks, and advanced architectures such as deep Turing machines.
- Apply deep learning mechanisms to real-world learning problems in domains such as vision, NLP, and sequence modeling.
- Critically evaluate current research, open issues, and emerging directions in deep learning.

Course Outcomes:

After completion of the course, students should be able to:

1. Demonstrate foundational knowledge of learning techniques, network layers, and their roles in deep learning models.
2. Explain and derive neural network training methods, loss functions, optimization algorithms, and regularization strategies.
3. Classify and compare different deep learning architectures such as feedforward, convolutional, recurrent, autoencoders, and probabilistic neural networks.
4. Design and implement deep models (e.g. CNN, RNN, autoencoders) using suitable frameworks or tools.
5. Critically assess the limitations, open problems, and research trends in deep learning.

UNIT - I:

Introduction: Various paradigms of learning problems, Perspectives and Issues in deep learning framework, review of fundamental learning techniques. Feed forward neural network: Artificial Neural Network, activation function, multi-layer neural network.

UNIT - II:

Training Neural Network: Risk minimization, loss function, back propagation, regularization, model selection, and optimization.

Conditional Random Fields: Linear chain, partition function, Markov network, Belief propagation, Training CRFs, Hidden Markov Model, Entropy.

UNIT - III:

Deep Learning: Deep Feed Forward network, regularizations, training deep models, dropouts, Convolution Neural Network, Recurrent Neural Network, and Deep Belief Network.

UNIT - IV:

Probabilistic Neural Network: Hopfield Net, Boltzmann machine, RBMs, Sigmoid net, Auto encoders.

UNIT - V:

Applications: Object recognition, sparse coding, computer vision, natural language processing. Introduction to Deep Learning Tools: Caffe, Tensor Flow, Torch.

Text Books:

1. Ian Goodfellow, Yoshua Bengio & Aaron Courville, Deep Learning, MIT Press, 2016
2. Christopher M. Bishop, Pattern Recognition and Machine Learning, 1st Edition, Springer, 2006

Reference Books:

1. B. Yegnanarayana, Artificial Neural Networks, Prentice-Hall of India / PHI Learning, 2009 (1st/earlier edition)
2. Gene H. Golub & Charles F. Van Loan, Matrix Computations, 4th Edition, Johns Hopkins University Press, 2013
3. Satish Kumar, Neural Networks: A Classroom Approach, Tata McGraw-Hill, 2004

I Year-II Semester

L	T	P	C
3	1	0	4

BLOCKCHAIN TECHNOLOGIES

Course Objectives:

Students undergoing this course are expected to:

1. Architect sensor networks for various application setups.
2. Devise appropriate data dissemination protocols and model links cost.

Course Outcomes:

After completion of the course, students should be able to:

1. Discuss the Cryptographic primitives used in Blockchain
2. Discuss about various technologies borrowed in blockchain
3. Illustrate various models for blockchain
4. Discuss about Ethereum
5. Discuss about Hyperledger Fabric

UNIT – I:

INTRODUCTION TO BLOCKCHAIN: Introduction, history of Bitcoin and origins of Blockchain, Fundamentals of Blockchain and key components (Chapter 1-book1), Permission and Permission-less platforms(Chapter 1-book2), Introduction to Cryptography, SHA256 and ECDSA, Hashing and Encryption, Symmetric/ Asymmetric keys, Private and Public Keys(Chapter 3-book2).

UNIT–II:

TECHNOLOGIES BORROWED IN BLOCKCHAIN: Technologies Borrowed in Blockchain–hash pointers—Digital cash etc.-Bitcoin blockchain-Wallet–Blocks Merkley Tree - hardness of mining - Transaction verifiability - Anonymity -forks - Double spending - Mathematical analysis of properties of Bitcoin -Bitcoin-the challenges and solutions. (Chapter 3-book2).

UNIT – III:

CONSENSUS MECHANISMS: Consensus Algorithms: Proof of Work (PoW) as randomoracle-Formal treatment of consistency-Liveness and Fairness-Proof of Stake (PoS) based Chains -Hybrid models (PoW + PoS), Byzantine Models of fault tolerance. (Chapter 1-book2)

UNIT – IV:

ETHEREUM: Ethereum- Ethereum Virtual Machine(EVM)-Wallets for Ethereum-Solidity-Smart Contracts (Chapter 5-book1),- The Turing Completeness of Smart Contract Languages and verification challenges-Using smart contracts to enforce legal contracts-Comparing Bitcoin scripting vs. Ethereum Smart Contracts-Some attacks on smart contracts (Chapter 6 and Chapter 7-book2)

UNIT – V:

HYPERLEDGER FABRIC: Hyper ledger fabric- the plug and play platform and mechanisms in permissioned blockchain - Beyond Cryptocurrency – applications of blockchain in cyber security-integrity of information- E-Governance and other contract enforcement mechanisms-Limitations of block chain as a technology and myths vs reality of blockchain technology (Chapter 16-book1), (Chapter 9 -book2)

Text Books:

1. Blockchain Technology Chandramouli Subramanian, Asha A George, Abhilash K A and Meena Karthikeyan, University Press, 2020.
2. Mastering Blockchain - Distributed ledger technology, decentralization, and smart contracts explained, Imran Bashir,2nd Edition,2018, pakct publication

Reference Books:

1. Shukla, M.Dhawan, S.Sharma, S.Venkatesan "Blockchain Technology: Cryptocurrency and Applications", Oxford University Press, 2019.
2. Cryptography and network security principles and practice, William Stallings, Pearson, 8th edition,

WEB REFERENCES:

1. https://drive.google.com/file/d/1PtYaDmWYaqPVGjKDnMYGWO5eoI5wMPtJ/view
2. https://archive.nptel.ac.in/courses/106/104/106104220/
3. https://www.tutorialspoint.com/blockchain/index.htm

I Year-II Semester

L	T	P	C
3	1	0	4

INTRODUCTION TO QUANTUM COMPUTING

Course Objectives:

Students undergoing this course are expected to:

1. Introduce fundamental concepts of quantum mechanics and its mathematical formalism.
2. Explore quantum computing and communication principles and technologies.

Course Outcomes:

After completion of the course, students should be able to:

1. Describe the Historical development of quantum theory and its relevance to modern computing
2. Define Qubits and Compare the Classical vs. quantum information
3. Explain the Classical computing review and limitations
4. Demonstrate the principles and techniques of Quantum error correction
5. Discuss the working, applications and potential of Quantum sensors in real-world scenarios

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

Text Books

1. Nielsen, M. A., & Chuang, I. L., Quantum Computation and Quantum Information, 10th Anniversary Edition, Cambridge University Press, 2010 (Reprinted 2021).

Reference Books

1. Yanofsky, N. S., & Mannucci, M. A., Quantum Computing for Computer Scientists, Cambridge University Press, 2008 (Latest reprint 2020).
2. Benenti, G., Casati, G., & Strini, G., Principles of Quantum Computation and Information,
3. Vol. I: Basic Concepts, World Scientific, 2004 (Reprint 2019).
4. Vol. II: Basic Tools and Special Topics, World Scientific, 2007 (Reprint 2020).
5. Pittenger, A. O., An Introduction to Quantum Computing Algorithms, Birkhäuser, 2000 (Reprint 2019).

I Year-II Semester

L	T	P	C
3	1	0	4

SOFT COMPUTING

(Professional Elective-III)

Course Objectives:

Students undergoing this course are expected to:

- Provide an understanding of the mathematical foundations and principles behind soft computing techniques.
- Equip students with the skills to apply soft computing techniques

Course Outcomes:

After completion of the course, students should be able to:

1. Learn soft computing techniques and their applications.
2. Analyze various neural network architectures.
3. Define the fuzzy systems
4. Understand the genetic algorithm concepts and their applications.
5. Identify and select a suitable Soft Computing technology to solve the problem; construct a solution and implement a Soft Computing solution

UNIT – I:

Introduction to Soft Computing, Artificial neural networks, biological neurons, Basic models of artificial neural networks, Connections, Learning, Activation Functions, McCulloch and Pitts Neuron, Hebb network.

UNIT – II:

Perceptron networks, Learning rule, Training and testing algorithm, Adaptive Linear Neuron, Back propagation Network, Architecture, Training algorithm

UNIT – III:

Fuzzy logic, fuzzy sets, properties, operations on fuzzy sets, fuzzy relations, operations on fuzzy relations, Fuzzy membership functions, fuzzification, Methods of membership, value assignments, intuition, inference, rank ordering, Lambda –Cuts for fuzzy sets , Defuzzification methods

UNIT – IV:

Truth values and Tables in Fuzzy Logic, Fuzzy propositions, Formation of fuzzy rules, Decomposition of rules, Aggregation of rules, Fuzzy Inference Systems, Mamdani and Sugeno types, Neuro-fuzzy hybrid systems, characteristics, classification

UNIT – V:

Introduction to genetic algorithm, operators in genetic algorithm, coding, selection, crossover, mutation, stopping condition for genetic algorithm flow, Genetic-neuro hybrid systems, Genetic Fuzzy rule based system

Text Books:

1. S. N. Sivanandam and S. N. Deepa, Principles of soft computing–John Wiley & Sons, 2007.
2. Timothy J. Ross, Fuzzy Logic with engineering applications, John Wiley & Sons, 2016.

Reference Books:

1. N.K. Sinhaand M.M. Gupta, Soft Computing & Intelligent Systems: Theory & Applications-Academic Press /Elsevier. 2009.
2. Simon Haykin, Neural Network-A Comprehensive Foundation-Prentice Hall International, Inc.1998
3. R. Eberhart and Y. Shi, Computational Intelligence: Concepts to Implementation, Morgan Kaufman/Elsevier, 2007.
4. Driankov D., Hellendoorn H. and Reinfrank M., An Introduction to Fuzzy Control Narosa Pub., 2001.
5. BartKosko, Neural Network and Fuzzy Systems-Prentice Hall, Inc., Englewood Cliffs, 1992
6. Goldberg D.E, Genetic Algorithms in Search , Optimization , and Machine Learning Addison Wesley, 1989

I Year-II Semester

L	T	P	C
3	1	0	4

INTERNET OF THINGS
(Professional Elective-III)

Course Objectives:

Students undergoing this course are expected to:

- Understand the fundamentals of Internet of Things (IoT) and its building blocks along with their characteristics.
- Learn recent application domains of IoT in everyday life.
- Explore integration of Cyber Physical Systems with IoT for intelligent automation and real-world applications

Course Outcomes:

After completion of the course, students should be able to:

1. Understand Characteristics and Design of Internet of Things (IoT).
2. Compare various M2M and IoT architectures.
3. Study various Cloud Storage Models for IoT.
4. Design IoT Systems using Python and embedded devices.
5. Apply Data Analytics and Cyber Physical System concepts for IoT-based smart environments.

UNIT-I:

Introduction: Internet of Things – Definition & Characteristics of IoT, Physical Design of IoT, Logical Design of IoT, IoT Enabling Technologies, IoT Levels & Deployment Templates.

Domain Specific IoTs: Home, Cities, Environment, Energy Systems, Logistics, Agriculture, Health & Lifestyle.

UNIT-II:

IoT & M2M: Introduction to M2M, Difference between IoT and M2M, SDN and NFV for IoT, Need for IoT Systems Management, Simple Network Management Protocol (SNMP), Limitations of

SNMP, Network Operator Requirements, NETCONF, YANG, IoT Systems Management with NETCONF-YANG, NETOPEER.

UNIT-III:

IoT Platforms Design Methodology: IoT Design Methodology, Case Study on IoT System for Weather Monitoring.

Python for IoT: Motivation for Using Python, IoT Systems – Logical Design using Python, Installing Python, Python Data Types & Data Structures, Control Flow, Functions, Modules, Packages, File Handling, Date/Time Operations, Classes, Python Packages of Interest for IoT.

UNIT-IV:

IoT Physical Devices & Cloud Integration: IoT Physical Devices & Endpoints – Raspberry Pi: About the Board, Linux on Raspberry Pi, Raspberry Pi Interfaces, Programming Raspberry Pi with Python, Other IoT Devices.

Physical Servers & Cloud Offerings – Introduction to Cloud Storage Models & Communication APIs, WAMP – AutoBahn for IoT, Xively Cloud for IoT, Python Web Application Framework – Django, Amazon Web Services for IoT.

Cyber Physical Systems (CPS): Introduction to CPS, Relationship between IoT and CPS, Architecture of CPS, Sensing, Computation, and Actuation Loop, Control and Communication in CPS, Embedded Systems and Real-time Operation in CPS, Applications of CPS in Smart Cities, Healthcare, and Industrial Automation.

UNIT-V:

Data Analytics for IoT: Introduction, Apache Hadoop, Using Hadoop MapReduce for Batch Data Analysis, Apache Oozie, Apache Spark, Apache Storm, Using Apache Storm for Real-Time Data Analysis.

Text Books:

1. A. Bahga and V. Madisetti, Internet of Things – A Hands-On Approach, 1st Edition, University Press (India), 2014.
2. Raj Kamal, Internet of Things: Architecture and Design Principles, 2nd Edition, McGraw Hill Education, 2022.
3. Raj Rajkumar and Insup Lee, Cyber-Physical Systems: Foundations, Principles and Applications, 1st Edition, CRC Press, 2016.

Reference Books:

1. K. A. Lambert and B. L. Juneja, Fundamentals of Python: First Programs, 2nd Edition, Cengage Learning, 2019.
2. Rajkumar Buyya and Amir V. Dastjerdi, Internet of Things: Principles and Paradigms, 1st Edition, Morgan Kaufmann (Elsevier), 2016.
3. Olivier Hersent, David Boswarthick, and Omar Elloumi, The Internet of Things: Key Applications and Protocols, 2nd Edition, Wiley, 2012.
4. Edward A. Lee and Sanjit A. Seshia, Introduction to Embedded Systems: A Cyber-Physical Systems Approach, 2nd Edition, MIT Press, 2017.

I Year-II Semester

L	T	P	C
3	1	0	4

SOCIAL NETWORK ANALYSIS

(Professional Elective-III)

Course Objectives:

Students undergoing this course are expected to:

- The learning objective of the course Social Network Analysis is to provide students with essential knowledge of network analysis applicable to real world data, with examples from today's most popular social networks.

Course Outcomes:

After completion of the course, students should be able to:

1. Demonstrate social network analysis and measures.
2. Analyze random graph models and navigate social networks data
3. Apply the network topology and Visualization tools.
4. Analyze the experiment with small world models and clustering models.
5. Compare the application driven virtual communities from social network Structure.

UNIT - I:

Social Network Analysis: Preliminaries and definitions, Erdos Number Project, Centrality measures, Balance and Homophily.

UNIT - II:

Random graph models: Random graphs and alternative models, Models of network growth, Navigation in social Networks, Cohesive subgroups, Multidimensional Scaling, Structural equivalence, roles and positions.

UNIT - III:

Network topology and diffusion, Contagion in Networks, Complex contagion, Percolation and information, Navigation in Networks Revisited.

UNIT - IV:

Small world experiments, small world models, origins of small world, Heavy tails, Small Diameter, Clustering of connectivity, The ErdosRenyi Model, Clustering Models.

UNIT - V:

Network structure -Important vertices and page rank algorithm, towards rational dynamics in networks, basics of game theory, Coloring and consensus, biased voting, network formation games, network structure and equilibrium, behavioral experiments, Spatial and agent-based models.

Text Books:

1. S. Wasserman and K. Faust. Social Network Analysis: Methods and Applications (Cambridge, Cambridge University Press, 1994)
2. D. Easley and J. Kleinberg, Networks, Crowds and Markets: Reasoning about a highly connected world-2010

Reference Books:

1. Social Network Analysis: Methods and Applications (Structural Analysis in the Social Sciences) by Stanley Wasserman, Katherine Faust, 1994.

I Year-II Semester

L	T	P	C
3	1	0	4

DEVOPS

(Professional Elective-III)

Course Objectives:

Students undergoing this course are expected 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.

Course Outcomes:

After completion of the course, students should be able to:

1. Explain DevOps Life cycle process
2. Demonstrate the concept of Code coverage
3. Explain Jenkins, jenkins workflow, jenkins master slave architecture, Jenkins Pipelines
4. Discuss the concept of Dockers Command and running containers
5. Analyze and troubleshoot configuration and deployment issues in automated environments.

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-

CODECOVERAGE: 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: Importance of Continuous Delivery, CONTINUOUS DEPLOYMENT CD Flow, Containerization with Docker: Introduction to Docker, Docker installation, Docker commands, Images & Containers, Docker File, running containers, working with containers and publish to Docker Hub.

Testing Tools: Introduction to Selenium and its features, Java Script testing

UNIT – V:

Configuration Management - ANSIBLE: Introduction to Ansible, Ansible tasks Roles, Jinja2 templating, Vaults, Deployments using Ansible. CONTAINERIZATION USING UBERNETES(OPENSHIFT): Introduction to Kubernetes Namespace & Resources, CI/CD - On OCP, BC, DC& Config Maps, Deploying Apps on Open shift Container Pods. Introduction to Puppet master and Chef

Text Books:

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

Reference Books:

1. Len Bass, Ingo Weber, Liming Zhu. DevOps: A Software Architect's Perspective. Addison Wesley; ISBN-10

2. Gene Kim Je Humble, Patrick Debois, John Willis. The DevOps Handbook, 1st Edition, IT Revolution Press, 2016.
3. Verona , Joakim Practical DevOps,1st Edition, Packt Publishing, 2016.
4. Joakim Verona. Practical Devops, Second Edition. In gram short title; 2nd edition (2018). ISBN10: 1788392574
5. Deepak Gaikwad, Viral Thakkar. DevOps Tools from Practitioner's View point. Wiley publications, ISBN:9788126579952

Web Resources:

1. https://archive.nptel.ac.in/courses/106/104/106104220/
2. https://www.tutorialspoint.com/blockchain/index.htm

I Year-I Semester

L	T	P	C
3	1	0	4

NATURAL LANGUAGE PROCESSING

(Professional Elective-IV)

Course Objectives:

Students undergoing this course are expected to:

- Explore statistical and machine learning approaches for NLP tasks, including language modeling, classification, and sequence labeling.
- Provide knowledge of advanced NLP techniques, such as word embeddings, transformer models, attention mechanisms, and contextual representations.

Course Outcomes:

After completion of the course, students should be able to:

1. Demonstrate a given text with basic Language features
2. Design an innovative application using NLP components
3. Explain a rule based system to tackle morphology/syntax of a language
4. Design a tag set to be used for statistical processing for real-time applications
5. Compare and contrast the use of different statistical approaches for different types of NLP applications

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

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.

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, Nitin Indurkhya and Fred J. Damerau, Chapman and Hall/CRC Press, 2010.

4. Natural Language Processing and Information Retrieval, 3rdEdition, Tanveer Siddiqui, U.S. Tiwary, Oxford University Press, 2008.

I Year-II Semester

L	T	P	C
3	1	0	4

GENERATIVE AI

(Professional Elective-IV)

Course Objectives:

Students undergoing this course are expected to:

- Learn Python and TensorFlow skills for Generative AI.
- Study techniques for cleaning and preparing data for Generative AI tasks.
- Understand and apply prompt engineering for improving interaction with large language models.

Course Outcomes:

After completion of the course, students should be able to:

1. Implement Python and TensorFlow basics, including data handling and preprocessing techniques.
2. Implement Generative AI models such as GANs, VAEs, LSTM networks, and Transformer models for image, text, and music generation tasks.
3. Evaluate model performance and experiment with hyperparameters and optimization techniques to enhance Generative AI outcomes.
4. Develop innovative applications in image, text, and music generation.
5. Apply prompt engineering techniques for effective communication and control of large language models.

UNIT – I:

Introduction to Generative AI: Historical Overview of Generative Modelling, Difference between Generative and Discriminative Modeling, Importance of Generative Models in AI and ML, Types of Generative Models – GANs, VAEs, Autoregressive and Diffusion Models.

Understanding Probabilistic Modelling and Generative Processes, Challenges in Generative Modelling, Future of Generative AI, Ethical Aspects of AI, Responsible AI, Use Cases.

UNIT – II:

Generative Models for Text: Language Models Basics, Building Blocks of Language Models, Transformer Architecture, Encoder–Decoder Models, Attention Mechanisms, Text Generation, Models like BERT and GPT, Autoencoding, Regression Models, ChatGPT, RLHF, Retrieval Augmented Generation (RAG), Multimodal LLMs, Issues such as Hallucination.

Prompt Engineering: Concept and Importance of Prompt Engineering, Types of Prompts (Zero-shot, Few-shot, Chain-of-Thought, Role-based), Designing Effective Prompts for Text and Code Generation, Context Optimization and Prompt Templates, Prompt Tuning and Fine-tuning, Reinforcement Learning from Human Feedback (RLHF) and Prompt Refinement, Evaluation of Prompts, Tools for Prompt Testing and Iteration, Applications in Chatbots, Code Assistants, and Creative Generation.

UNIT – III:

Generation of Images: Introduction to Generative Adversarial Networks, Adversarial Training Process, Nash Equilibrium, Variational Autoencoders, Encoder–Decoder Architectures, Stable Diffusion Models, Transformer-based Image Generation, CLIP, Visual Transformers (ViT), DALL-E 2 and DALL-E 3, GPT-4V, Issues of Image Generation such as Mode Collapse and Stability.

UNIT – IV:

Generation of Painting, Music, and Play: Variants of GAN – DCGAN, CycleGAN; Using CycleGAN to Generate Paintings, Neural Style Transfer, Music Generation using RNN and MuseGAN, Autonomous Agents, Deep Q-Learning, Actor–Critic Networks.

UNIT – V:

Open Source Models and Programming Frameworks: Training and Fine-tuning of Generative Models, GPT-4 All, Transfer Learning and Pretrained Models, Training Vision Models, Google Copilot, Programming with LLMs, LangChain Framework, Open Source Models (LLaMA, Mistral, Falcon), Programming with TimeSformer, Deployment using Hugging Face.

Text Books:

1. Denis Rothman, Transformers for Natural Language Processing and Computer Vision, 3rd Edition, Packt Publishing, 2024.

2. Jason D. Brownlee, Prompt Engineering for Generative AI: Building Effective Prompts for LLMs, 1st Edition, Machine Learning Mastery, 2024.

Reference Books:

1. David Foster, Generative Deep Learning, 2nd Edition, O'Reilly Media, 2024.
2. Altaf Rehmani, Generative AI for Everyone, BlueRose One, 2024.
3. Ankur Patel and Ajay Uppili Arasanipalai, Applied Generative AI with ChatGPT and OpenAI API, 2nd Edition, Packt Publishing, 2024.
4. Lilian Weng, Prompt Engineering and Advanced LLM Techniques, OpenAI Educational Series, 2024.

I Year-II Semester

L	T	P	C
3	1	0	4

CYBER SECURITY

(Professional Elective-IV)

Course Objectives:

Students undergoing this course are expected to:

- Familiarize students with cyber attack types
- Develop an understanding of network security principles

Course Outcomes:

After completion of the course, students should be able to:

1. Understand key terms and concepts in cyber security fundamentals.
2. Gain knowledge about attacker techniques and motivations.
3. Differentiate various governing bodies of cyber laws.
4. Understand principles of malicious code.
5. Obtain comprehensive knowledge on Defense and Analysis Techniques including forensic tools.

UNIT-I:

Cyber Security Fundamentals: Information Assurance Fundamentals, Basic Cryptography, Symmetric Encryption, Public Key Encryption, The Domain Name System (DNS), Microsoft Windows Security Principles: Windows Tokens, Window Messaging, Windows Program Execution

UNIT-II:

Cyber Attacks and Application Security: Introduction to cyber-attacks, application security (design, development and testing), operations security, monitoring, identifying threats and remediating them, Principles of data security - Confidentiality, Integrity and Availability, Data Privacy, Data breaches, preventing attacks and breaches with security controls, Compliance standards, Computer Ethics.

UNIT-III:

Cyber Security Management and Law: Cyber Security Management Security Planning - Business Continuity Planning - Handling Incidents - Risk Analysis - Dealing With Disaster –Legal Issues – Protecting programs and Data.

Introduction to Cyber Law: Cyber Law, Need for Cyber Law Jurisprudence of Indian Cyber, Law, Evolution of Cyber Crime

UNIT-IV:

Malicious Code and Exploitation: Self-Replicating Malicious Code, Evading Detection and Elevating Privileges, Stealing Information and Exploitation, Research ideas in Cyber Security.

UNIT-V:

Defense, Forensics and Analysis Techniques: Memory Forensics, Honeypots, Malicious Code Naming, Automated Malicious Code Analysis Systems, **Introduction to Cyber Forensics and Forensic Tools** – Disk Imaging, File Recovery, Log Analysis, and Network Forensics. Overview of Tools such as Autopsy, FTK Imager, EnCase, Wireshark, Volatility, and Sleuth Kit. Cyber Security Current Trends and Standards.

Text Books:

1. Cyber Security Essentials (James Graham, Richard Howard, Ryan Olson) — 1st Edition 2011.
2. The Basics of Cyber Safety: Computer and Mobile Device Safety Made Easy (John Sammons & Michael Cross) — 1st Edition September 2016.

Reference Books:

1. Nelson, Phillips, and Steuart, Guide to Computer Forensics and Investigations, 6th Edition, Cengage Learning, 2022.
2. Chuck Easttom, Computer Security Fundamentals, 4th Edition, Pearson, 2022.
3. William Stallings, Network Security Essentials: Applications and Standards, 7th Edition, Pearson, 2023.

I Year-II Semester

L	T	P	C
3	1	0	4

DESIGN PATTERNS

(Professional Elective-IV)

Course Objectives:

Students undergoing this course are expected to:

- Demonstration of patterns related to object oriented design.
- Describe the design patterns that are common in software applications.

Course Outcomes:

After completion of the course, students should be able to:

1. Construct a design consisting of a collection of modules
2. Exploit well-known design patterns (such as Iterator, Observer, Factory and Visitor). Analyze
3. Distinguish between different categories of design patterns. Analyze
4. Ability to understand and apply common design patterns to incremental/iterative development.
5. identify appropriate patterns for design of given problem

UNIT – I:

What is a Design Pattern, Design Patterns in Smalltalk MVC, Describing Design Patterns, The Catalogue of Design Patterns, Organizing The Cato log, How Design Patterns solve Design Problems, How to Select a Design pattern, How to Use a Design Pattern.

UNIT – II:

A Case Study: Designing a Document Editor, Design Problems , Document Structure, Formatting , Embellishing the User Interface, Supporting Multiple Look-and-Feel Standards, Supporting Multiple Window Systems, User Operations Spelling Checking and Hyphenation, Summary, Creational

Patterns, Abstract Factory, Builder , Factory Method, Prototype, Singleton, Discussion of Creational Patterns.

UNIT – III:

Structural Pattern Part-I, Adapter, Bridge, Composite. Structural Pattern Part-II, Decorator, Facade, Flyweight, Proxy.

UNIT – IV:

Behavioral Patterns Part: I, Chain of Responsibility, Command, Interpreter, Iterator. Behavioral Patterns Part: II, Mediator, Memento, Observer, Discussion of Behavioral Patterns.

UNIT – V:

Behavioral Patterns Part: III, State, Strategy, Template Method, Visitor, Discussion of Behavioral Patterns. What to Expect from Design Patterns, A Brief History, The Pattern Community, An Invitation, A Parting Thought.

Text Book:

1. Erich Gamma, Richard Helm, Ralph Johnson & John Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software, 1st Edition, Addison-Wesley, 1994

Reference Books:

1. Mark Grand, Patterns in Java, Volume 1: A Catalog of Reusable Design Patterns Illustrated with UML, 2nd Edition, Wiley, 2002
2. Mark Grand, Java Enterprise Design Patterns, Volume III, Wiley

I Year-II Semester

L	T	P	C
0	1	2	3

DEEP LEARNING LAB

Course Objectives:

Students undergoing this course are expected to:

- Implement and experiment with core deep learning models (FFNN, CNN, RNN, RBM/AE, Boltzmann Machines) and training algorithms.
- Understand and apply training strategies (losses, backprop, regularization, dropout, optimization) and probabilistic models (CRF, HMM).
- Use modern DL frameworks (TensorFlow / Keras, PyTorch) to build end-to-end solutions for vision & NLP tasks and evaluate model performance.

Course Outcomes:

After completion of the course, students should be able to:

1. Explain the fundamentals of deep learning paradigms and various neural network architectures.
2. Apply appropriate training methods (loss functions, backpropagation, optimization, and regularization) for deep models.
3. Design and implement convolutional, recurrent, and deep belief networks for real-world data.
4. Analyze and evaluate probabilistic neural network models such as Boltzmann Machines, RBMs, and Autoencoders.
5. Utilize deep learning frameworks such as TensorFlow, Keras, PyTorch, and Theano to develop solutions for applications in computer vision and natural language processing.

Experiment- 1: — Intro environment & Python basics for DL- Set up environment (TF, PyTorch), run "Hello Neural Net". Install packages, run simple NumPy operations, train tiny FFNN on XOR Sample Dataset: Synthetic XOR.

Experiment- 2: —Build & train an MLP for digit classification - Implement from-scratch (NumPy) one-hidden-layer NN; then implement with Keras (Sequential) & compare. Sample Dataset: MNIST.

Experiment- 3: —Implement backprop manually for a small network; compare optimizers (SGD, Momentum, Adam). Track training loss curves; experiment with learning rates and batch sizes. Sample Dataset: Fashion-MNIST or small regression dataset.

Experiment- 4: —Implement L1/L2 regularization, dropout, early stopping, k-fold validation. Demonstrate overfitting/underfitting and apply fixes. Sample Dataset: CIFAR-10 subset or UCI dataset.

Experiment- 5: —Implement HMM basics and train a linear-chain CRF for sequence labeling. Implement Viterbi decoding for HMM; use a library (e.g., sklearn-crfsuite or pystruct) to train CRF for POS tagging / simple NER. Sample Dataset: Small POS/NER dataset (CoNLL subset) or synthetic sequence labels.

Experiment- 6: —Build basic CNNs (Conv → Pool → FC), visualize filters/feature maps. Train LeNet-like model on MNIST; transfer to CIFAR-10. Visualize intermediate activations. Sample Dataset: MNIST, CIFAR-10.

Experiment- 7: —Use augmentation, batch normalization, residual connections (simple). Compare baseline vs. augmented training; implement a small ResNet block. Sample Dataset: CIFAR-10.

Experiment- 8: —Implement RNN/LSTM for sequence prediction and language modeling. Character-level language model, next-word prediction; compare RNN vs LSTM. Sample Dataset: Tiny Shakespeare, Penn Treebank subset, or custom text.

Experiment- 9: —Implement simple autoencoder and visualize compressed representations. (DBNs can be explored via an RBM library), Train denoising autoencoder on MNIST; visualize latent space with t-SNE. Sample Dataset: MNIST.

Experiment- 10: Implement Hopfield net for pattern storage and recall. Store few binary patterns; test recall from noisy inputs. Sample Dataset: Synthetic binary patterns.

Experiment- 11: —Implement/troubleshoot RBM training (contrastive divergence) and stack into DBN. Train RBM on small binary or binarized MNIST; sample reconstructions. Sample Dataset: Binarized MNIST.

Experiment- 12: —Implement different AE flavors and compare reconstructions & latent representations. Train AE variants and visualize latent clusters. Sample Dataset: CIFAR-10 / MNIST.

Experiment- 13: Build an end-to-end classifier with preprocessing, augmentation, training, evaluation, and deployment stub (Flask). Train on a 5-class dataset, export model, build simple REST API. Sample Dataset: Caltech101 subset or custom images.

Experiment- 14: — Build text classification or NER pipeline using embeddings + LSTM/Transformer-lite. Use pre-trained embeddings (GloVe) or small Transformer block; evaluate. Sample Dataset: IMDB (sentiment) or CoNLL (NER).

Experiment- 15: — Install/run a simple model in Caffe / Torch (if desired) to understand legacy frameworks. Run pre-configured example model and observe differences to TF/PyTorch.

Text Books:

1. François Chollet, Deep Learning with Python, 3rd Edition, Manning Publications, 2024.
2. Ian Goodfellow, Yoshua Bengio, and Aaron Courville, Deep Learning, 1st Edition, MIT Press, 2016.
3. Aurélien Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 3rd Edition, O'Reilly Media, 2023.
4. Sudharsan Ravichandiran, Deep Learning with PyTorch: Build, Train, and Tune Neural Networks Using Python Tools, 2nd Edition, Packt Publishing, 2022.

Reference Books:

1. Prajit Dhar and Raghav Bali, Practical Deep Learning: A Python-Based Introduction, 2nd Edition, Apress, 2023.
2. Eli Stevens, Luca Antiga, and Thomas Viehmann, Deep Learning with PyTorch, 2nd Edition, Manning Publications, 2023.

3. Anirudh Koul, Siddha Ganju, and Meher Kasam, Practical Deep Learning for Cloud, Mobile, and Edge, 2nd Edition, O'Reilly Media, 2023.
4. Pavel Shumskiy, Modern Deep Learning: Practical Neural Networks with Python, TensorFlow, and PyTorch, 1st Edition, BPB Publications, 2024.

I Year-II Semester

L	T	P	C
0	1	2	2

QUANTUM COMPUTING LAB

Course Objectives:

Students undergoing this course are expected to:

1. To introduce the historical background and interdisciplinary foundations of Quantum Computing.
2. To provide a strong foundation in linear algebra, probability, and quantum physics required for quantum computing

Course Outcomes:

After completion of the course, students should be able to:

1. Differentiate between classical and quantum computation paradigms
2. Apply mathematical and physical principles to analyze quantum systems
3. Design and simulate quantum circuits using quantum programming frameworks
4. Implement basic quantum algorithms (Deutsch, Grover, Shor, etc.) and evaluate their performance
5. Demonstrate understanding of quantum error correction and quantum cryptographic protocols

Experiment– 1: Classical vs Quantum Bits Simulation

- Implement a basic program to show difference between bit and qubit (superposition of states).

Experiment– 2: Single Qubit Gates

- Implement and visualize operations of Pauli-X, Y, Z, Hadamard, Phase gates on Bloch Sphere

Experiment– 3: Multiple Qubit Gates

- Design and execute a CNOT gate and show how entanglement arises

Experiment– 4: Superposition & Measurement

- Apply Hadamard to |0⟩, measure in both computational and Hadamard basis.

Experiment– 5: Bell State Preparation & Verification

- Generate Bell states using Hadamard + CNOT and verify entanglement.

Experiment– 6: Deutsch Algorithm Implementation

- Implement and test Deutsch's Algorithm for constant and balanced functions

Experiment– 7: Deutsch–Jozsa Algorithm

- Demonstrate exponential speed-up over classical algorithms

Experiment– 8: Grover's Algorithm

- Implement Grover's search for finding a marked element in an unsorted dataset.

Experiment– 9: Shor's Algorithm (Simulation)

- Demonstrate factoring small numbers (e.g., 15) on a quantum simulator.

Experiment– 10: Quantum Cryptography & Teleportation

- Simulate BB84 protocol and quantum teleportation using entangled qubits.

Text Books:

1. Michael A. Nielsen & Isaac L. Chuang, Quantum Computation and Quantum Information, 10th Anniversary Edition, Cambridge University Press, 2011.
2. H.-Y. Wong, Introduction to Quantum Computing: From a Layperson's Guide, 2nd Edition, Springer, 2024.
3. Rajkumar Buyya & S. S. Gill, Quantum Computing: Principles & Paradigms, 1st Edition, Morgan Kaufmann, 2025.
4. Chris Bernhardt, Quantum Computing for Everyone, MIT Press, 2019.

Reference Books:

1. Noson S. Yanofsky & Mirco A. Mannucci, Quantum Computing for Computer Scientists, 1st Edition, Cambridge University Press, 2019.
2. Giuseppe Benenti, Giuliano Casati & Giulio Strini, Principles of Quantum Computation and Information, Vol. I: Basic Concepts, Vol. II: Basic Tools and Special Topics, World Scientific, 2019.
3. Pittenger, A. O., An Introduction to Quantum Computing Algorithms, 1st Edition, Springer, 2016.