MARRI LAXMAN REDDY
INSTITUTE OF TECHNOLOGY AND MANAGEMENT
(AN AUTONOMOUS INSTITUTION)
(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad)
Accredited by NAAC with 'A' Grade & Recognized Under Section 2(f) & 12(B) of the UGC act,1956

M.TECH - Computer Science and Engineering

COURSE STRUCTURE (MT25)

Applicable From 2025-26 Admitted Batch

S. No	Category	Breakup of credits (Total 68 credits)
1	Programme Core	22
2	Professional Elective	15
3	Open Elective	3
4	Project Work	28
5	Audit Course	-
	Total Credits	**68**

MARRI LAXMAN REDDY
INSTITUTE OF TECHNOLOGY AND MANAGEMENT
(AN AUTONOMOUS INSTITUTION)
(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad)
Accredited by NBA and NAAC with 'A' Grade & Recognized Under Section2(f) & 12(B)of the UGC act,1956

Department of Computer Science and Engineering

MLRS-MT25 Course Structure
Applicable From 2025-26 Admitted Batch

I YEAR I SEMESTER (I SEMESTER)

S. No.	Course Code	Course Title	Course Area	Hours Per Week			Credits	Scheme of Examination Maximum Marks		
				L	T	P		Internal (CIA)	External (SEE)	Total
		Theory								
1	2515801	Mathematical Foundations of Computer Science	CS	3	0	0	3	40	60	100
2	2515802	Advanced Data Structures	CS	3	0	0	3	40	60	100
3	Professional Elecive-I	1. Database Programming with PL/SQL.- 2. Deep Learning 3. Natural Language Processing. 4. Advanced UNIX Programming.	CS	3	0	0	3	40	60	100
4	Professional Elecive-II	1. Applied Cryptography 2. Software Quality Engineering 3. Mining Massive Datasets. 4. Agile Methodologies.	CS	3	0	0	3	40	60	100
		Laboratory								
1	2515871	Advanced Data Structures Lab	CS	0	0	4	2	40	60	100
2		Professional Elective-I Lab	CS	0	0	4	2	40	60	100
3	2510001	Research Methodology & IPR	HS	2	0	0	2	40	60	100
		Audit-I								
1	Audit Course	Audit Course-I	AC	2	0	0	0	100	-	100
	Total Credits			16	0	8	18	280	520	800

I YEAR II SEMESTER (II SEMESTER)

S. No.	Course Code	Course Title	Course Area	Hours Per Week			Credits	Scheme of Examination Maximum Marks		
				L	T	P		Internal (CIA)	External (SEE)	Total
		Theory								
1	2525803	Advanced Algorithms	CS	3	0	0	3	40	60	100
2	2525804	Advanced Computer Architecture	CS	3	0	0	3	40	60	100
3	Professional Elecive-III	1. Enterprise Cloud Concepts 2. Cyber Security 3. Parallel Computing 4. Large Language Models	CS	3	0	0	3	40	60	100
4	Professional Elecive-IV	1. Bioinformatics 2. Adhoc Sensor Networks. 3. Robotic Process Automation 4. Generative AI	CS	3	0	0	3	40	60	100
		Laboratory								
1	2525872	Advanced Algorithms Lab	CS	0	0	4	2	40	60	100
2		Professional Elecive-III Lab	CS	0	0	4	2	40	60	100
3	2525873	Mini Project with Seminar	CS	0	0	4	2	100	-	100
		Audit-II								
1	2520003	Audit Course-II	AC	2	0	0	0	100	-	100
Total Credits				14	0	12	18	280	520	800

II YEAR I SEMESTER (III SEMESTER)

S. No.	Course Code	Course Title	Course Area	Hours Per Week			Credits	Scheme of Examination Maximum Marks		
				L	T	P		Internal (CIA)	External (SEE)	Total
		Theory								
1	Professional Elecive-V	1. Digital Forensics 2. Advanced Operating Systems 3. Quantum Computing 4. Prompt Engineering	CS	3	0	0	3	40	60	100
3		Open Elective		3	0	0	3	40	60	100
4	2535874	Dissertation Work Review-I	CS	0	0	18	6	100	-	100
Total Credits				3	0	18	12	180	120	300

II YEAR II SEMESTER (IV SEMESTER)

S. No.	Course Code	Course Title	Course Area	Hours Per Week			Credits	Scheme of Examination Maximum Marks		
				L	T	P		Internal (CIA)	External (SEE)	Total
1	2545875	Dissertation Work Review-III	CS	0	0	18	6	100	-	100
2	2545876	Dissertation Viva-Voice	CS	0	0	42	14	-	100	100
	Total Credits			0	0	60	20	100	100	800

Professional Elective I

S No	Course Code	Course Name
1	2515805	Database Programming with PL/SQL
2	2515806	Deep Learning
3	2515807	Natural Language Processing
4	2515808	Advanced UNIX Programming
5	2515877	Database Programming with PL/SQL Lab
6	2515878	Deep Learning Lab
7	2515879	Natural Language Processing Lab
8	2515880	Advanced UNIX Programming Lab

Professional Elective II

S No	Course Code	Course Name
1	2515809	Applied Cryptography
2	2515810	Software Quality Engineering
3	2515811	Mining Massive Datasets
4	2515811	Agile Methodologies

Professional Elective III

S No	Course Code	Course Name
1	2525812	Enterprise Cloud Concepts
2	2525813	Cyber Security
3	2525814	Parallel Computing
4	2525815	Large Language Models
5	2525881	Enterprise Cloud Concepts Lab
6	2525882	Cyber Security Lab
7	2525883	Parallel Computing Lab
8	2525884	Large Language Models Lab

Professional Elective IV

S No	Course Code	Course Name
1	2525816	Bioinformatics
2	2525817	Adhoc Sensor Networks
3	2525818	Robotic Process Automation
4	2525818	Generative AI

Professional Elective V:

S No	Course Code	Course Name
1	2535819	Digital Forensics
2	2535820	Advanced Operating Systems
3	2535821	Quantum Computing
4	2535822	Prompt Engineering

Audit Course-I & II :

S No	Course Code	Course Name
1	2510002	English for Research Paper Writing
2	2510003	Disaster Management
3	2510004	Sanskrit for Technical Knowledge
4	2510005	Value Education
5	2520006	Constitution of India
6	2520007	Pedagogy Studies
7	2520008	Stress Management by Yoga
8	2520009	Personality Development Through Life Enlightenment Skills

Open Electives for Other Departments:

S No	Course Code	Course Name
1	25X5823	Intellectual Property Rights
2	25X5824	Introduction to Generative AI
3	25X5825	Intrusion Detection Systems
4	25X5826	Digital Forensics

I-I

MATHEMATICAL FOUNDATIONS OF COMPUTER SCIENCE (PC-I)

M.Tech CSE I Year I Sem. LTP C
3 0 0 3

Pre-requisites: An understanding of Math in general is sufficient.

Course Objectives: To learn

1. Introduces the elementary discrete mathematics for computer science and engineering.
2. Topics include formal logic notation, methods of proof, induction, sets, relations, graph theory, permutations and combinations, counting principles; recurrence relations and generating functions.

Course Outcomes: After learning the contents of this paper the student must be able to

1. Apply the principles of propositional and predicate logic to solve problems in computer science and engineering domains.
2. Analyse set-theoretic structures, relations, and functions to establish mathematical foundations for advanced algorithms and computational models in real world scenarios.
3. Make Use of mathematical induction, recursion, and program verification techniques in terms of complexity and correctness for the challenging algorithms.
4. Utilize Bayes' theorem, and advanced counting techniques to optimize combinatorial problems of the problems in multidisciplinary domains.
5. Create mathematical models using recurrence relations, generating functions, and graph-theoretic techniques to solve real-world research-oriented problems.

UNIT-I

The Foundations Logic and Proofs: Propositional Logic, Applications of Propositional Logic, Propositional Equivalence, Predicates and Quantifiers, Nested Quantifiers, Rules of Inference, Introduction to Proofs, Proof Methods and Strategy.

UNIT-II

Basic Structures, Sets, Functions, Sequences, Sums, Matrices and Relations: Sets, Functions, Sequences & Summations, Cardinality of Sets and Matrices Relations, Relations and Their Properties, n-ary Relations and Their Applications, Representing Relations, Closures of Relations, Equivalence Relations, Partial Orderings.

UNIT-III

Algorithms, Induction and Recursion: Algorithms, The Growth of Functions, Complexity of Algorithms. Induction and Recursion: Mathematical Induction, Strong Induction and Well-Ordering, Recursive Definitions and Structural Induction, Recursive Algorithms, Program Correctness.

UNIT-IV

Discrete Probability and Advanced Counting Techniques:
An Introduction to Discrete Probability. Probability Theory, Bayes' Theorem, Expected Value and Variance. Advanced Counting Techniques:
Recurrence Relations, Solving Linear Recurrence Relations, Divide-and-Conquer Algorithms and Recurrence Relations, Generating Functions, Inclusion-Exclusion, Applications of Inclusion-Exclusion.

UNIT-V

Graphs: Graphs and Graph Models, Graph Terminology and Special Types of Graphs, Representing Graphs and Graph Isomorphism, Connectivity, Euler and Hamilton Paths, Shortest-Path Problems, Planar Graphs, Graph Coloring.
Trees: Introduction to Trees, Applications of Trees, Tree Traversal, Spanning Trees, Minimum Spanning Trees.

TEXT BOOKS:

1. Discrete Mathematical Structures with Applications to Computer Science: J.P. Tremblay, R. Manohar, McGraw-hill, 1st ed.
2. Discrete Mathematics for Computer Scientists & Mathematicians: Joe l. Mott, Abraham Kandel, Teodore P. Baker, Prentis Hall of India, 2nd ed.

REFERENCE:

1. Discrete and Combinatorial Mathematics - an applied introduction: Ralph.P. Grimald, Pearson education, 5th edition.
2. Discrete Mathematical Structures: Thomas Kosy, Tata McGraw Hill publishing co.

ADVANCED DATA STRUCTURES (PC-II)

M.Tech CSE I Year I Sem.

L	T	P	C
3	0	0	3

Prerequisites

1. A course on "Data Structures".

Course Objectives

1. Introduces the heap data structures such as leftist trees, binomial heaps, Fibonacci and min- max heaps.
2. Introduces a variety of data structures such as disjoint sets, hash tables, search structures and digital search structures.

Course Outcomes

1. Apply heap and hashing techniques for efficient data organization and retrieval in real-world applications of databases and memory management.
2. Analyze balanced and multiway search trees for dynamic searching operations in multidisciplinary domains of network routing and AI-based search systems.
3. Implement digital search structures such as tries and suffix trees for software engineering and compiler design.
4. Evaluate collision resolution strategies and hashing methods for research in cryptography and distributed systems.
5. Design efficient pattern-matching solutions using classical and advanced algorithms for industry applications in cybersecurity and text analytics.

UNIT - I

Heap Structures

Introduction, Min-Max Heaps, Leftist trees, Binomial Heaps, Fibonacci heaps.

UNIT - II

Hashing and Collisions

Introduction, Hash Tables, Hash Functions, different Hash Functions:- Division Method, Multiplication Method, Mid-Square Method, Folding Method, Collisions

UNIT - III

Search Structures: OBST, AVL trees, Red-Black trees, Splay trees,Multiway Search Trees,B-trees, 2-3 trees

UNIT - IV

Digital Search Structures,Digital Search trees, Binary tries and Patricia, Multiway Tries, Suffix trees, Standard Tries, Compressed Tries

UNIT - V

Pattern matching,Introduction, Brute force, the Boyer –Moore algorithm, Knuth-Morris-Pratt algorithm, Naïve String, Harspool, Rabin Karp

TEXT BOOKS:

1. Fundamentals of data structures in C++ Sahni, Horowitz, Mehatha, Universities Press.
2. Introduction to Algorithms, TH Cormen, PHI

REFERENCES:

1. Design methods and analysis of Algorithms, SK Basu, PHI.
2. Data Structures & Algorithm Analysis in C++, Mark Allen Weiss, Pearson Education.
3. Fundamentals of Computer Algorithms, 2nd Edition, Ellis Horowitz, Sartaj Sahni, Sanguthevar Rajasekaran, Universities Press.

DATABASE PROGRAMMING WITH PL/SQL (PROFESSIONAL ELECTIVE - I)

M.Tech CSE I Year I Sem. **L T P C**
3 0 0 3

Course Objectives:

1. Knowledge on significance of SQL fundamentals.
2. Evaluate functions and triggers of PL/SQL
3. Knowledge on control structures, packages in PL/SQL and its applications

Course Outcomes:

1. Apply PL/SQL block structures, variables, control statements, and exception handling to solve problems.
2. Design collections and bulk operations for efficient data processing in e-commerce transactions.
3. Develop functions and procedures with proper transaction handling and parameter passing business logic modules of ERP systems.
4. Construct and manage PL/SQL packages with appropriate definer, invoker rights for components of inventory, payroll systems.
5. Make use of triggers to automate database events for auditing financial transactions and enforcing data integrity in databases.

Unit I

PL/SQL Basics

Block Structure, Behavior of Variables in Blocks, Basic Scalar and Composite Data Types, Control Structures, Exceptions, Bulk Operations, Functions, Procedures, and Packages, Transaction Scope

Unit II

Language Fundamentals & Control Structures

Lexical Units, Variables and Data Types, Conditional Statements, Iterative Statements, Cursor Structures, Bulk Statements, Introduction to Collections, Object Types: Varray and Table Collections, Associative Arrays, Oracle Collection API

Unit III

Functions and Procedures

Function and Procedure Architecture, Transaction Scope, Calling Subroutines, Positional Notation, Named Notation, Mixed Notation, Exclusionary Notation, SQL Call Notation, Functions, Function Model Choices, Creation Options, Pass-by-Value Functions, Pass-by-Reference Functions, Procedures, Pass-by-Value Procedures, Pass-by-Reference Procedures, Supporting Scripts.

Unit IV

Packages

Package Architecture, Package Specification, Prototype Features, Serially Reusable Precompiler Directive, Variables, Types, Components: Functions and Procedures, Package Body, Prototype Features, Variables, Types, Components: Functions and Procedures, Definer vs. Invoker Rights Mechanics, Managing Packages in the Database Catalog, Finding, Validating, and Describing Packages, Checking Dependencies, Comparing Validation Methods: Timestamp vs. Signature.

Unit V

Triggers

Introduction to Triggers, Database Trigger Architecture, Data Definition Language Triggers, Event Attribute Functions, Building DDL Triggers, Data Manipulation Language Triggers, Statement-Level Triggers, Row-Level Triggers, Compound Triggers, INSTEAD OF Triggers, System and Database Event Triggers, Trigger Restrictions, Maximum Trigger Size, SQL Statements, LONG and LONG RAW Data Types.

TEXT BOOKS:

1. Oracle Database 12c PL/SQL Programming Michael McLaughlin, McGraw Hill Education.

REFERENCES:

1. Benjamin Rosenzweig, Elena Silvestrova Rakhimov, Oracle PL/SQL by example Fifth Edition.
2. Dr. P. S. Deshpande, SQL & PL / SQL for Oracle 11g Black Book.

DEEP LEARNING (PROFESSIONAL ELECTIVE - I)

M.Tech CSE I Year I Sem.

L	T	P	C
3	0	0	3

Course Objectives:

1. To understand complexity of Deep Learning algorithms and their limitations
2. To be capable of performing experiments in Deep Learning using real-world data.

Course Outcomes:

1. Apply feed-forward and backpropagation techniques in optimizing predictive models for real-time image or speech recognition.
2. Design convolutional, recurrent, and attention-based neural network architectures for image classification and LSTMs in sequence prediction.
3. Develop deep unsupervised learning models, including autoencoders, variational autoencoders, and GANs, in generating synthetic images or enhancing feature representation in medical imaging.
4. Analyze deep learning models for computer vision tasks in building automated video surveillance, self-driving car vision and captioning systems.
5. Evaluate word embeddings, sentiment analysis and dialogue generation in social media and intelligent text summarization.

UNIT - I

Introduction: Feed forward Neural networks, Gradient descent and the back propagation algorithm, Unit saturation, the vanishing gradient problem, and ways to mitigate it. RelU Heuristics for avoiding bad local minima, Heuristics for faster training, Nestors accelerated gradient descent, Regularization, Dropout

UNIT - II

Convolutional Neural Networks: Architectures, convolution/pooling layers, Recurrent Neural Networks: LSTM, GRU, Encoder Decoder architectures. Deep Unsupervised Learning: Auto encoders, Variational Auto-encoders, Adversarial Generative Networks, Auto-encoder and DBM Attention and memory models, Dynamic Memory Models

UNIT - III

Applications of Deep Learning to Computer Vision: Image segmentation, object detection, automatic image captioning, Image generation with Generative adversarial networks, video to text with LSTM models, Attention Models for computer vision tasks

UNIT -IV

Applications of Deep Learning to NLP: Introduction to NLP and Vector Space Model of Semantics, Word Vector Representations: Continuous Skip-Gram Model, Continuous Bag-of-Words model (CBOW), Glove, Evaluations and Applications in word similarity

UNIT -V

Analogy reasoning: Named Entity Recognition, Opinion Mining using Recurrent Neural Networks: Parsing and Sentiment Analysis using Recursive Neural Networks: Sentence Classification using Convolutional Neural Networks, Dialogue Generation with LSTMs

TEXT BOOKS:

1. Deep Learning by Ian Goodfellow, Yoshua Bengio and Aaron Courville, MIT Press.
2. The Elements of Statistical Learning by T. Hastie, R. Tibshirani, and J. Friedman, Springer.
3. Probabilistic Graphical Models. Koller, and N. Friedman, MIT Press.

REFERENCES:

1. Bishop, C, M., Pattern Recognition and Machine Learning, Springer, 2006.
2. Yegnanarayana, B., Artificial Neural Networks PHI Learning Pvt. Ltd, 2009.
3. Golub, G.,H., and Van Loan, C.,F., Matrix Computations, JHU Press,2013.
4. Satish Kumar, Neural Networks: A Classroom Approach, Tata McGraw-Hill Education, 2004.

Extensive Reading:

1. https://www.deeplearning.net
2. https://www.deeplearningbook.org/
3. https://developers.google.com/machine-learning/crash-course/ml-intro
4. www.cs.toronto.edu/~fritz/absps/imagenet.pdf
5. https://neuralnetworksanddeeplearning.com/

NATURAL LANGUAGE PROCESSING (PROFESSIONAL ELECTIVE - I)

M.Tech CSE I Year I Sem.

L	T	P	C
3	0	0	3

Prerequisites:

1. Data structures, finite automata and probability theory.

Course Objectives:

1. Introduction to some of the problems and solutions of NLP and their relation to linguistics and statistics.

Course Outcomes:

1. Apply morphological analysis techniques for improving text preprocessing in information retrieval systems.
2. Construct syntactic parsers using treebanks and parsing algorithms to develop multilingual grammar checking tools.
3. Implement semantic parsing and word sense disambiguation to enhance question-answering and knowledge extraction systems.
4. Analyze predicate-argument structures and meaning representation systems to support automated reasoning in intelligent dialogue systems.
5. Evaluate n-gram and cross-lingual models for text prediction and machine translation applications.

UNIT - I

Finding the Structure of Words: Words and Their Components, Issues and Challenges, Morphological Models

Finding the Structure of Documents: Introduction, Methods, Complexity of the Approaches, Performances of the Approaches

UNIT - II

Syntax Analysis: Parsing Natural Language, Treebanks: A Data-Driven Approach to Syntax, Representation of Syntactic Structure, Parsing Algorithms, Models for Ambiguity Resolution in Parsing, Multilingual Issues

UNIT - III

Semantic Parsing: Introduction, Semantic Interpretation, System Paradigms, Word Sense Systems, Software.

UNIT - IV

Predicate-Argument Structure, Meaning Representation Systems, Software.

UNIT - V

Discourse Processing: Cohesion, Reference Resolution, Discourse Cohesion and Structure Language Modeling: Introduction, N-Gram Models, Language Model Evaluation, Parameter Estimation, Language Model Adaptation, Types of Language Models, Language-Specific Modeling Problems, Multilingual and Cross Lingual Language Modeling

TEXT BOOKS:

1. Multilingual natural Language Processing Applications: From Theory to Practice – Daniel M. Bikel and Imed Zitouni, Pearson Publication.
2. Natural Language Processing and Information Retrieval: Tanvier Siddiqui, U.S. Tiwary.

REFERENCES:

1. Speech and Natural Language Processing - Daniel Jurafsky & James H Martin, Pearson Publications.

ADVANCED UNIX PROGRAMMING (PROFESSIONAL ELECTIVE - I)

M.Tech CSE I-I Sem

L	T	P	C
3	0	0	3

Course Objectives:

1. To understand and make effective use of Linux utilities and Shell scripting language (bash) to solve Problems.
2. To implement in C some standard Linux utilities such as ls, mv, cp etc. using system calls.
3. To develop the skills necessary for systems programming including file system programming, process and signal management, and Inter Process Communication.
4. To develop the basic skills required to write network programs using Sockets.

Course Outcomes:

1. Apply Linux file handling, process, networking, and text utilities to automate system administration tasks and backup management.
2. Implement shell scripts using control structures, variables, functions, and I/O redirection to develop automated system monitoring tools.
3. Analyze file system structures, file metadata, and directory operations to optimize storage management and access control in multi-user environments.
4. Design interprocess communication mechanisms, including pipes, FIFOs, message queues, and semaphores, for concurrent and client-server applications.
5. Evaluate shared memory and socket-based IPC solutions for building scalable networked applications and multi-client server systems.

UNIT- I

Linux Utilities - File handling utilities, Security by file permissions, Process utilities, Disk utilities, Networking commands, Filters, Text processing utilities and Backup utilities.

Shell programming with Bourne again shell (bash) - Introduction, shell responsibilities, pipes and Redirection, here documents, running a shell script, the shell as a programming language, shell meta characters, file name substitution, shell variables, command substitution, shell commands, the environment, quoting, test command, control structures, arithmetic in shell, shell script examples, interrupt processing, functions, debugging shell scripts.

UNIT- II

Files and Directories - File Concept, File types, File System Structure, file metadata-Inodes, kernel support for files, system calls for file I/O operations- open, creat, read, write, close, lseek, dup2, file status information-stat family, file and record locking- fcntl function, file permissions - chmod, fchmod, file ownership-chown, lchown, fchown, links-soft links and hard links – symlink, link, unlink.

Directories

- Creating, removing and changing Directories- mkdir, rmdir, chdir, obtaining current working directory- getcwd, Directory contents, Scanning Directories-opendir, readdir, closedir, rewinddir functions.

UNIT- III

Process – Process concept, Layout of a C program image in main memory, Process environment-environment list, environment variables, getenv, setenv, Kernel support for process, process identification, process control - process creation, replacing a process image, waiting for a process, process termination, zombie process, orphan process, system call interface for process management-fork, vfork, exit, wait, waitpid, exec family, Process Groups, Sessions and Controlling Terminal, Differences between threads and processes.

Signals – Introduction to signals, Signal generation and handling, Kernel support for signals, Signal function, unreliable signals, reliable signals, kill, raise, alarm, pause, abort, sleep functions.

UNIT- IV

Interprocess Communication - Introduction to IPC, IPC between processes on a single computer system, IPC between processes on different systems, pipes-creation, IPC between related processes using unnamed pipes, FIFOs-creation, IPC between unrelated processes using FIFOs (Named pipes), differences between unnamed and named pipes, popen and pclose library functions. **Message Queues** - Kernel support for messages, APIs for message queues, client/server example. **Semaphores** - Kernel support for semaphores, APIs for semaphores, file locking with semaphores.

UNIT- V

Shared Memory - Kernel support for shared memory, APIs for shared memory, shared memory example. **Sockets** - Introduction to Berkeley Sockets, IPC over a network, Client- Server model, Socket address structures (Unix domain and Internet domain),Socket system calls for connection oriented protocol and connectionless protocol, example-client/server programs-Single Server-Client connection, Multiple simultaneous clients, Socket options- setsockopt and fcntl system calls, Comparison of IPC mechanisms.

TEXT BOOKS:

1. Unix System Programming using C++, T. Chan, PHI.
2. Advanced Programming in the Unix Environment, 2nd edition, W. R. Stevens and S. A. Rago, Pearson Education.
3. Unix Concepts and Applications, 4th Edition, Sumitabha Das, TMH.
4. Unix Network Programming, W. R. Stevens, PHI.

REFERENCE BOOKS:

1. C Programming Language, Kernighan and Ritchie, PHI.
2. Beginning Linux Programming, 4th Edition, N. Matthew, R. Stones, Wrox, Wiley India Edition.
3. Unix for programmers and users, 3rd Edition, Graham Glass, King Ables, Pearson.
4. System Programming with C and Unix, A. Hoover, Pearson.
5. Unix System Programming, Communication, Concurrency and Threads, K. A. Robbins and S. Robbins, Pearson Education.
6. Unix shell Programming, S. G. Kochan and P. Wood, 3rd edition, Pearson Education.
7. Shell Scripting, S. Parker, Wiley India Pvt. Ltd.
8. Unix and Shell programming, B. A. Forouzan and R. F. Gilberg, Cengage Learning.
9. Linux System Programming, Robert Love, O'Reilly, SPD.

APPLIED CRYPTOGRAPHY (PROFESSIONAL ELECTIVE - II)

M.Tech CSE I Year I Sem.

L	T	P	C
3	0	0	3

Course Objectives: Knowledge on significance of cryptographic protocols and symmetric and public key algorithms

Course Outcomes:

1. Apply substitution, transposition, XOR, and one-time pad techniques to secure data communication in confidential messaging systems.
2. Analyze symmetric and public-key algorithms, key lengths, and cipher modes to design secure encryption systems for financial transactions.
3. Implement public-key algorithms and digital signature schemes such as RSA, DSA, and ECC to ensure authentication and integrity in e-commerce platforms.
4. Design advanced cryptographic protocols, including zero-knowledge proofs, secret sharing, and oblivious transfer, for secure multiparty computations.
5. Evaluate real-world cryptographic standards and protocols such as Kerberos, PGP, and PKCS for securing enterprise communication and payment systems.

Unit I

Foundations:

Terminology, Steganography, Substitution Ciphers and Transposition Ciphers, Simple XOR, One- Time Pads, Computer Algorithms, Large Numbers,

Cryptographic Protocols: Protocol Building Blocks

Introduction to Protocols, Communications Using Symmetric Cryptography, One-Way Functions, One-Way Hash Functions, Communications Using Public-Key Cryptography, Digital Signatures, Digital Signatures with Encryption, Random and Pseudo-Random-Sequence Generation

Unit II

Cryptographic Techniques

Key length: Symmetric Key length, Public key length, comparing symmetric and public key length.

Algorithm types and modes: Electronic Codebook Mode, Block Replay, Cipher Block Chaining Mode, Stream Cipher, Self-Synchronizing Stream Ciphers, Cipher-Feedback Mode, Synchronous Stream Ciphers, Output-Feedback Mod, Counter Mode, Other Block-Cipher Modes.

Unit III

Public-Key Algorithms

Background, Knapsack Algorithms, RSA, Pohlig-Hellman, Rabin, ElGamal, McEliece, Elliptic Curve Cryptosystems, LUC, Finite Automaton Public-Key Cryptosystems

Public-Key Digital Signature Algorithms: Digital Signature Algorithm (DSA), DSA Variants, Gost Digital Signature Algorithm, Discrete Logarithm Signature Schemes, Ong-Schnorr-Shamir, ESIGN

Unit IV

Special Algorithms for Protocols

Multiple-Key Public-Key Cryptography, Secret-Sharing Algorithms, Subliminal Channel, Undeniable Digital Signatures, Designated Confirmer Signatures, Computing with Encrypted Data, Fair Coin Flips, One-Way Accumulators, All-or-Nothing Disclosure of Secrets, Fair and Failsafe Cryptosystems, Zero-Knowledge Proofs of Knowledge, Blind Signatures, Oblivious Transfer, Secure Multiparty Computation, Probabilistic Encryption, Quantum Cryptography

Unit V

Real World Approaches

IBM Secret key management protocol, ISDN, Kerberos, KryptoKnight, Privacy enhanced mail (PEM), Message security protocol (MSP), PGP, Public-Key Cryptography Standards (PKCS), Universal Electronic Payment System (UEPS).

TEXT BOOKS:

1. Bruce Schneier, Applied Cryptography, Second Edition: Protocols, Algorithms, and Source Code in C (cloth).

SOFTWARE QUALITY ENGINEERING (PROFESSIONAL ELECTIVE - II)

M.Tech CSE I Year I Sem.

L	T	P	C
3	0	0	3

Course Objectives:

Knowledge on significance of Quality, quality assurance, quality engineering.

Course Outcomes:

1. Apply substitution, transposition, XOR, and one-time pad techniques to secure data communication in confidential messaging systems.
2. Analyze symmetric and public-key algorithms, key lengths, and cipher modes to design secure encryption systems for financial transactions.
3. Implement public-key algorithms and digital signature schemes such as RSA, DSA, and ECC to ensure authentication and integrity in e-commerce platforms.
4. Design advanced cryptographic protocols, including zero-knowledge proofs, secret sharing, and oblivious transfer, for secure multiparty computations.
5. Evaluate real-world cryptographic standards and protocols such as Kerberos, PGP, and PKCS for securing enterprise communication and payment systems.

Unit I Software Quality

Quality: perspectives and expectations, Quality frameworks and ISO-9126, correctness and defects: Definitions, properties and Measurements, A historical perspective of quality, software quality.

Unit II Quality Assurance

Classification: QA as dealing with defects, Defect prevention- Education and training, Formal method, Other defect prevention techniques, Defect Reduction - Inspection: Direct fault detection and removal, Testing: Failure observation and fault removal, other techniques and risk identification, Defect Containment- software fault tolerance, safety assurance and failure containment

Unit III Quality Engineering

Quality Engineering: Activities and process, Quality planning: Goal setting and Strategy formation, Quality assessment and Improvement, Quality engineering in software process.

Unit IV Test Activities, Management and Automation

Test planning and preparation, Test execution, Result checking and measurement, Analysis and follow- up, Activities People and Management, Test Automation.

Unit V Coverage and usage testing based on checklist and partitions

Checklist based testing and its limitations, Testing for partition Coverage, Usage based Statistical testing with Musa's operational profiles, Constructing operational profiles

Case Study: OP for the cartridge Support Software

TEXT BOOKS:

1. Jeff Tian, Software Quality Engineering, Testing, Quality Assurance, and Quantifiable improvement
2. Richard N. Taylor, Software Architecture: Foundations, Theory, and Practice

MINING MASSIVE DATASETS (PROFESSIONAL ELECTIVE - II)

M.Tech CSE I Year I Sem.

L	T	P	C
3	0	0	3

Prerequisites:

1. Students should be familiar with Data mining, algorithms, basic probability theory and Discrete math.

Course Objectives:

1. This course will cover practical algorithms for solving key problems in mining of massive datasets.
2. This course focuses on parallel algorithmic techniques that are used for large datasets.
3. This course will cover stream processing algorithms for data streams that arrive constantly, page ranking algorithms for web search, and online advertisement systems that are studied in detail.

Course Outcomes:

1. Apply MapReduce and distributed file systems to process and analyze large-scale datasets for real-time analytics applications.
2. Implement similarity search and streaming data techniques to detect near-duplicate documents and monitor real-time data streams.
3. Analyze link structures, frequent itemset, and clustering algorithms to improve recommendation and ranking systems for e-commerce platforms.
4. Design web advertising and recommendation system algorithms, including collaborative filtering and dimensionality reduction, for personalized marketing.
5. Evaluate social network graph mining techniques, including graph clustering and partitioning, to extract insights for social media analytics.

UNIT I

Data Mining-Introduction-Definition of Data Mining-Statistical Limits on Data Mining,
MapReduce and the New Software Stack-Distributed File Systems, MapReduce, Algorithms Using MapReduce.

UNIT II

Similarity Search: Finding Similar Items**-**Applications of Near-Neighbor Search, Shingling of Documents, Similarity-Preserving Summaries of Sets, Distance Measures.
Streaming Data: Mining Data Streams-The Stream Data Model , Sampling Data in a Stream, Filtering Streams

UNIT III

Link Analysis-PageRank, Efficient Computation of PageRank, Link Spam
Frequent Itemsets-Handling Larger Datasets in Main Memory, Limited-Pass Algorithms, Counting Frequent Items in a Stream.
Clustering-The CURE Algorithm, Clustering in Non-Euclidean Spaces, Clustering for Streams and Parallelism

UNIT IV

Advertising on the Web-Issues in On-Line Advertising, On-Line Algorithms, The Matching Problem, The Adwords Problem, Adwords Implementation.**Recommendation Systems-**A Model for Recommendation Systems,Content-Based Recommendations, Collaborative Filtering, Dimensionality Reduction, The NetFlix Challenge.

UNIT V

Mining Social-Network Graphs-Social Networks as Graphs, Clustering of Social-Network Graphs, Partitioning of Graphs, Simrank, Counting Triangles

TEXT BOOKS:

1. Jure Leskovec, Anand Rajaraman, Jeff Ullman, Mining of Massive Datasets, 3rd Edition.

REFERENCE BOOKS:

1. Jiawei Han & Micheline Kamber , Data Mining – Concepts and Techniques 3rd Edition Elsevier.
2. Margaret H Dunham, Data Mining Introductory and Advanced topics, PEA.
3. Ian H. Witten and Eibe Frank, Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann.

AGILE METHODOLOGIES (PROFESSIONAL ELECTIVE - II)

M.Tech CSE I Year I Sem.

L	T	P	C
3	0	0	3

Course Objectives

- Knowledge on concepts of Agile development, releasing, planning and developing

Course Outcomes

1. Apply Agile and Extreme Programming principles, including XP lifecycle and team practices, to improve software project adaptability and delivery efficiency.
2. Implement collaborative techniques such as pair programming, continuous integration, and iteration demos to enhance team productivity in software development projects.
3. Analyze release management processes, including version control, fast builds, and documentation practices, to ensure high-quality software delivery.
4. Design effective planning strategies, including risk management, iteration planning, and story estimation, for predictable and efficient software development.
5. Evaluate development practices such as test-driven development, refactoring, and exploratory testing to optimize software performance and maintainability.

UNIT - I

Introduction Extreme Programming (XP) - Agile Development
Why Agile - Understanding Success, Beyond Deadlines, Importance of Organizational Success, Introduction to Agility How to Be Agile - Agile methods, Don't make your own method, Road to mastery Understanding XP (Extreme Programming) - XP life cycle, XP team, XP Concepts Adopting XP - Knowing whether XP is suitable, Implementing XP, assessing Agility Practicing XP - Thinking - Pair Programming, Energized work, Informative Workspace, Root cause Analysis, Retrospectives

UNIT - II

Collaborating
Trust, Sit together, Real customer involvement, Ubiquitous language, meetings, coding standards, Iteration demo, Reporting

UNIT - III

Releasing
Bugfree Release, Version Control, fast build, continuous integration, Collective ownership, Documentation

UNIT – IV

Planning
Version, Release Plan, Risk Management, Iteration Planning, Slack, Stories, Estimating

UNIT - V

Developing:
Incremental requirements, Customer tests, Test driven development, Refactoring,
Incremental design and architecture, spike solutions, Performance optimization, Exploratory testing

TEXT BOOK:

1. The art of Agile Development, James Shore and Shane Warden, 11th Indian Reprint, O'Reilly,2018.

REFERENCES:

1. Learning Agile, Andrew Stellman and Jennifer Greene, O'Reilly, 4th Indian Reprint, 2018.
2. Practices of an Agile Developer, Venkat Subramaniam and Andy Hunt, SPD, 5th Indian Reprint, 2015.
3. Agile Project Management - Jim Highsmith, Pearson Low price Edition 2004.

ADVANCED DATA STRUCTURES LAB (LAB - I)

M.Tech CSE I Year I Sem.

L	T	P	C
0	0	4	2

Prerequisites: A course on Computer Programming & Data Structures

Course Objectives:

1. Introduces the basic concepts of Abstract Data Types.
2. Reviews basic data structures such as stacks and queues.
3. Introduces a variety of data structures such as hash tables, search trees, tries, heaps, graphs,
4. and B-trees.
5. Introduces sorting and pattern matching algorithms.

Course Outcomes:

1. Implement AVL trees, and Red-Black trees to perform efficient insertion, deletion, and search operations for database indexing and memory management.
2. Utilize Merge Sort, Heap Sort, and Quick Sort to optimize data organization for large-scale computational applications.
3. Develop B-trees, Min-Max heaps, Leftist trees, and Binomial heaps to manage dynamic datasets in file systems and priority queue applications.
4. Experiment hashing techniques for implementing dictionaries to ensure fast data retrieval in information retrieval and compiler symbol tables.
5. Evaluate Knuth-Morris-Pratt, Brute Force, and Boyer-Moore, for applications in text search, bioinformatics, and network intrusion detection.

List of Programs

1. Write a program to perform the following operations:

a) Insert an element into a binary search tree.
b) Delete an element from a binary search tree.
c) Search for a key element in a binary search tree.

2. Write a program for implementing the following sorting methods:
   a) Merge sort b) Heap sort c) Quick sort

3. Write a program to perform the following operations:
   a) Insert an element into a B- tree.
   b) Delete an element from a B- tree.
   c) Search for a key element in a B- tree.

4. Write a program to perform the following operations:
   a) Insert an element into a Min-Max heap
   b) Delete an element from a Min-Max heap
   c) Search for a key element in a Min-Max heap

5. Write a program to perform the following operations:
   a) Insert an element into a Lefiist tree
   b) Delete an element from a Leftist tree
   c) Search for a key element in a Leftist tree
6. Write a program to perform the following operations:
   a) Insert an element into a binomial heap
   b) Delete an element from a binomial heap.
   c) Search for a key element in a binomial heap

7. Write a program to perform the following operations:
    a) Insert an element into a AVL tree.
    b) Delete an element from a AVL search tree.
    c) Search for a key element in a AVL search tree.

8. Write a program to perform the following operations:
    a) Insert an element into a Red-Black tree.
    b) Delete an element from a Red-Black tree.
    c) Search for a key element in a Red-Black tree.

9. Write a program to implement all the functions of a dictionary using hashing.

10. Write a program for implementing Knuth-Morris-Pratt pattern matching algorithm.

11. Write a program for implementing Brute Force pattern matching algorithm.

12. Write a program for implementing Boyer pattern matching algorithm.

TEXT BOOKS:

1. Fundamentals of Data structures in C, E. Horowitz, S. Sahni and Susan Anderson Freed, 2nd Edition, Universities Press
2. Data Structures Using C – A.S. Tanenbaum, Y. Langsam, and M.J. Augenstein, PHI/Pearson education.
3. Introduction to Data Structures in C, Ashok Kamthane, 1st Edition, Pearson.

REFERENCES:

1. The C Programming Language, B.W. Kernighan, Dennis M.Ritchie, PHI/Pearson Education
2. C Programming with problem solving, J.A. Jones & K. Harrow, Dreamtech Press
3. Data structures: A Pseudocode Approach with C, R.F. Gilberg And B.A. Forouzan, 2nd Edition, Cengage Learning

DATABASE PROGRAMMING WITH PL/SQL LAB (LAB - II)

M.Tech CSE/CS I Year I Sem.

L	T	P	C
0	0	4	2

Course Objectives:

1. Knowledge on significance of SQL fundamentals.
2. Evaluate functions and triggers of PL/SQL
3. Knowledge on control structures, packages in PL/SQL and its applications

Course Outcomes:

1. Implement PL/SQL programs to insert, retrieve, and manipulate data in database tables for real-world applications such as payroll and inventory management.
2. Develop procedures and functions to encapsulate reusable business logic for tasks like banking transactions, salary computation, and automated record updates.
3. Make use of exception handling mechanisms in PL/SQL to manage runtime errors and ensure robust and reliable database applications.
4. Design packages and triggers to enforce business rules and automate database operations in enterprise applications.
5. Utilize embedded PL/SQL in host languages to integrate database operations with application programs for real-time transaction processing systems.

List of Experiments:

1. Write a Pl/SQL program using FOR loop to insert ten rows into a database table.
2. Given the table EMPLOYEE (EmpNo, Name, Salary, Designation, DeptID), write a cursor to select the five highest paid employees from the table.
3. Illustrate how you can embed PL/SQL in a high-level host language such as C/Java And demonstrates how a banking debit transaction might be done.
4. Given an integer i, write a PL/SQL procedure to insert the tuple (i, 'xxx') into a given relation.
5. Write a PL/SQL program to demonstrate Exceptions.
6. Write a PL/SQL program to demonstrate Cursors.
7. Write a PL/SQL program to demonstrate Functions.
8. Write a PL/SQL program to demonstrate Packages.
9. Write PL/SQL queries to create Procedures.
10. Write PL/SQL queries to create Triggers.

DEEP LEARNING LAB (LAB - II)

M.Tech CSE/CS I Year I Sem.

L	T	P	C
0	0	4	2

Course Objectives:

1. To Build the Foundation of Deep Learning.
2. To Understand How to Build the Neural Network.
3. To enable students to develop successful machine learning concepts.

Course Outcomes:

1. Implement Python programs and set up IDE environments to execute deep learning workflows for computer vision and NLP tasks.
2. Make use of deep learning libraries such as TensorFlow, Keras, and PyTorch to build and train neural network models for real-world applications.
3. Design convolutional neural networks (CNNs) for image classification tasks such as MNIST digit recognition and other computer vision problems.
4. Utilize recurrent neural network (RNN) models with LSTM/GRU layers to perform sentiment analysis and other sequence modelling tasks in NLP applications.
5. Implement autoencoders and generative adversarial networks (GANs) to perform encoding, image generation, and unsupervised learning tasks in real-world datasets.

LIST OF EXPERIMENTS:

1. Setting up the Spyder IDE Environment and Executing a Python Program
2. Installing Keras, Tensorflow and Pytorch libraries and making use of them
3. Applying the Convolution Neural Network on computer vision problems
4. Image classification on MNIST dataset (CNN model with Fully connected layer)
5. Applying the Deep Learning Models in the field of Natural Language Processing
6. Train a sentiment analysis model on IMDB dataset, use RNN layers with LSTM/GRU notes
7. Applying the Autoencoder algorithms for encoding the real-world data
8. Applying Generative Adversial Networks for image generation and unsupervised tasks.

TEXT BOOKS:

1. Deep Learning by Ian Goodfellow, Yoshua Bengio and Aaron Courville, MIT Press.
2. The Elements of Statistical Learning by T. Hastie, R. Tibshirani, and J. Friedman, Springer.
3. Probabilistic Graphical Models. Koller, and N. Friedman, MIT Press.

REFERENCES:

1. Bishop, C, M., Pattern Recognition and Machine Learning, Springer, 2006.
2. Yegnanarayana, B., Artificial Neural Networks PHI Learning Pvt. Ltd, 2009.
3. Golub, G.,H., and Van Loan, C.,F., Matrix Computations, JHU Press,2013.
4. Satish Kumar, Neural Networks: A Classroom Approach, Tata McGraw-Hill Education, 2004.

Extensive Reading:

1. http://www.deeplearning.net
2. https://www.deeplearningbook.org/
3. https://developers.google.com/machine-learning/crash-course/ml-intro
4. www.cs.toronto.edu/~fritz/absps/imagenet.pdf
5. http://neuralnetworksanddeeplearning.com/

NATURAL LANGUAGE PROCESSING LAB (LAB - II)

M.Tech CSE I Year I Sem.

L	T	P	C
0	0	4	2

Prerequisites: Data structures, finite automata and probability theory

Course Objectives:

- To Develop and explore the problems and solutions of NLP.

Course Outcomes:

- Implement Python programs for text preprocessing tasks such as tokenization, stop-word removal, and stemming to prepare textual data for NLP applications like chatbots and search engines.
- Apply part-of-speech tagging and morphological analysis to extract syntactic information for tasks such as grammar checking and text parsing.
- Develop word analysis and word generation techniques for building applications like auto-completion and predictive text systems.
- Construct chunking and N-gram models, including smoothing techniques, to perform phrase detection and probabilistic language modelling for speech recognition and text prediction.
- Evaluate the performance of NLP preprocessing and language modelling methods for real-world applications such as document summarization and sentiment analysis.

List of Experiments

Implement the following using Python

1. Tokenization
2. Stemming
3. Stop word removal (a, the, are,.)
4. Word Analysis
5. Word Generation
6. Pos tagging
7. Morphology
8. chunking
9. N-Grams
10. N-Grams Smoothing

TEXT BOOKS:

1. Multilingual natural Language Processing Applications: From Theory to Practice – Daniel M. Bikel and Imed Zitouni, Pearson Publication
2. Natural Language Processing and Information Retrieval: Tanvier Siddiqui, U.S. Tiwary

REFERENCES:

1. Speech and Natural Language Processing - Daniel Jurafsky & James H Martin, Pearson Publications

ADVANCED UNIX PROGRAMMING LAB (LAB - II)

M.Tech CSE I Year I Sem.

L	T	P	C
0	0	4	2

Prerequisites: C programming

Course Objectives:

To develop proficiency in Unix/Linux system calls, shell scripting, and advanced programming techniques for system-level development and administration

Course Outcomes:

1. Effectively use advanced command-line features, manage file systems, and customize their user environment.
2. Design, code, test, and debug complex shell scripts to automate system administration tasks and perform repetitive operations efficiently.
3. Understand and utilize various UNIX system calls (file APIs, process APIs) to interact directly with the operating system kernel.
4. Implement process management concepts using C programming, including process creation (fork, exec, wait), process termination, and handling signals.
5. Gain practical experience with essential UNIX programming tools like gcc (compiler), gdb (debugger), make (build automation), and version control systems (git)

List of Experiments

I. Fundamentals & Shell Scripting

Advanced Shell Scripting: Complex awk, sed, grep usage, scripting for system administration, automation, input/output redirection, pipes, command substitution.

Unix Utilities: find, xargs, cron, rsync, tar, awk, sed for text processing and file management.

II. Core System Programming (System Calls)

File I/O: open, read, write, close, lseek, file permissions (chmod, chown), stat.
Process Management: fork, exec, wait, exit, process IDs (PIDs), parent/child relationships, zombie/orphan processes.
Signals: Signal handling (signal, sigaction), common signals (SIGINT, SIGTERM, SIGKILL, SIGCHLD).

III. Inter-Process Communication (IPC) & Threading

IPC Mechanisms: Pipes (named/unnamed), Message Queues, Shared Memory, Semaphores.
Multithreading: POSIX Threads (Pthreads) - thread creation, joining, mutual exclusion (mutexes), condition variables, thread synchronization.

IV. Networking & Socket Programming

Socket APIs: socket, bind, listen, accept, connect, send, recv.
Protocol Implementation: TCP (stream sockets) and UDP (datagram sockets) client-server applications.

V. Advanced Topics & Debugging

Debugging & Profiling: Using gdb, valgrind, strace, ltrace, perf.
File Systems: Basic understanding of file system structure (inodes, directory entries).
System Monitoring: Using top, vmstat, iostat.

TEXT BOOKS:

1. Unix System Programming using C++, T. Chan, PHI.
2. Advanced Programming in the Unix Environment, 2nd edition, W. R. Stevens and S. A. Rago, Pearson Education.
3. Unix Concepts and Applications, 4th Edition, Sumitabha Das, TMH.
4. Unix Network Programming, W. R. Stevens, PHI.

REFERENCES:

1. C Programming Language, Kernighan and Ritchie, PHI.

2. Beginning Linux Programming, 4th Edition, N. Matthew, R. Stones, Wrox, Wiley India Edition.

3. Unix for programmers and users, 3rd Edition, Graham Glass, King Ables, Pearson.

RESEARCH METHODOLOGY & IPR

M.Tech CSE I Year I Sem.

L	T	P	C
2	0	0	2

Prerequisite: None

Course Objectives:

1. To understand the research problem
2. To know the literature studies, plagiarism and ethics
3. To get the knowledge about technical writing
4. To analyze the nature of intellectual property rights and new developments
5. To know the patent rights

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

1. Identify research problems using systematic approaches to data collection, analysis, and instrumentation for solving real-world challenges in science and engineering.
2. Analyze literature effectively while adhering to research ethics to support academic writing and plagiarism-free publications.
3. Develop structured research proposals and technical reports for applications in academic presentations, project funding, and professional communication.
4. Examine intellectual property systems, including patents, copyrights, and designs, to apply them in innovation protection and technology commercialization.
5. Evaluate case studies on emerging IPR issues such as biological systems, software, and traditional knowledge to support entrepreneurship and multidisciplinary research.

UNIT-I

Meaning of research problem, Sources of research problem, Criteria Characteristics of a good research problem, Errors in selecting a research problem, Scope and objectives of research problem. Approaches of investigation of solutions for research problem, data collection, analysis, interpretation, Necessary instrumentations

UNIT-II

Effective literature studies approaches, analysis, Plagiarism, Research ethics

UNIT-III

Effective technical writing, how to write report, Paper Developing a Research Proposal, Format of research proposal, a presentation and assessment by a review committee

UNIT-IV

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. Procedure for grants of patents, Patenting under PCT.

UNIT-V

Patent Rights: Scope of Patent Rights. Licensing and transfer of technology. Patent information and databases. Geographical Indications. New Developments in IPR: Administration of Patent System. New developments in IPR; IPR of Biological Systems, Computer Software etc. Traditional knowledge Case Studies, IPR and IITs.

TEXT BOOKS:

1. Stuart Melville and Wayne Goddard, "Research methodology: an introduction for science & engineering students"
2. C.R. Kothari, Research Methodology, methods & techniques, 2nd edition, New age International publishers

REFERENCES:

1. Ranjit Kumar, 2nd Edition, "Research Methodology: A Step by Step Guide for beginners"
2. Halbert, "Resisting Intellectual Property", Taylor & Francis Ltd ,2007.
3. Mayall, "Industrial Design", McGraw Hill, 1992.
4. Niebel, "Product Design", McGraw Hill, 1974.
5. Asimov, "Introduction to Design", Prentice Hall, 1962.
6. Robert P. Merges, Peter S. Menell, Mark A. Lemley, "Intellectual Property in New Technological Age", 2016.
7. T. Ramappa, "Intellectual Property Rights Under WTO", S. Chand, 2008.

I-II

ADVANCED ALGORITHMS (PC - III)

M.Tech CSE I Year II Sem.

L	T	P	C
3	0	0	3

Pre-Requisites: UG level course in Algorithm Design and Analysis

Course Objectives:

1. Introduce students to the advanced methods of designing and analyzing algorithms.
2. The student should be able to choose appropriate algorithms and use it for a specific problem.
3. To familiarize students with basic paradigms and data structures used to solve advanced algorithmic problems.
4. Students should be able to understand different classes of problems concerning their computation difficulties.
5. To introduce the students to recent developments in the area of algorithmic design.

Course Outcomes: After completion of course, students would be able to:

1. Apply sorting, graph traversal, and shortest path algorithms to solve optimization problems in networks and real-time systems.
2. Analyze matroids, greedy algorithms, and graph matching techniques for applications in resource allocation and scheduling.
3. Implement flow networks, matrix computations, and divide-and-conquer strategies for solving large-scale scientific and engineering problems.
4. Evaluate dynamic programming, modular arithmetic, and fast polynomial/DFT algorithms for secure communications and cryptography.
5. Design solutions for NP-complete problems using advanced paradigms and recent data structures to support multidisciplinary research.

UNIT – I

Sorting:

Review of various sorting algorithms, topological sorting

Graph:

Definitions and Elementary Algorithms: Shortest path by BFS, shortest path in edge-weighted case (Dijkstra's), depth-first search and computation of strongly connected components, emphasis on correctness proof of the algorithm and time/space analysis, example of amortized analysis.

UNIT – II

Matroids:

Introduction to greedy paradigm, algorithm to compute a maximum weight maximal independent set. Application to MST.

Graph Matching:

Algorithm to compute maximum matching. Characterization of maximum matching by augmenting paths, Edmond's Blossom algorithm to compute augmenting path.

UNIT - III

Flow-Networks:

Maxflow-mincut theorem, Ford-Fulkerson Method to compute maximum flow, Edmond-Karp maximum-flow algorithm.

Matrix Computations:

Strassen's algorithm and introduction to divide and conquer paradigm, inverse of a triangular matrix, relation between the time complexities of basic matrix operations, LUP-decomposition.

UNIT - IV

Shortest Path in Graphs:

Floyd-Warshall algorithm and introduction to dynamic programming paradigm. More examples of dynamic programming.

Modulo Representation of integers/polynomials:

Chinese Remainder Theorem, Conversion between base-representation and modulo-representation. Extension to polynomials. Application: Interpolation problem.

Discrete Fourier Transform (DFT):

In complex field, DFT in modulo ring. Fast Fourier Transform algorithm. Schonhage-Strassen Integer Multiplication algorithm.

UNIT - V

Linear Programming: Geometry of the feasibility region and Simplex algorithm

NP-completeness: Examples, proof of NP-hardness and NP-completeness.

Recent Trends in problem solving paradigms using recent searching and sorting techniques by applying recently proposed data structures.

REFERENCES:

1. Cormen, Leiserson, Rivest, Stein, "Introduction to Algorithms".
2. Aho, Hopcroft, Ullman "The Design and Analysis of Computer Algorithms".
3. Kleinberg and Tardos."Algorithm Design".

ADVANCED COMPUTER ARCHITECTURE (PC - IV)

M.Tech CSE I Year II Sem.

L	T	P	C
3	0	0	3

Prerequisites: Computer Organization

Course Objectives:

1. To impart the concepts and principles of parallel and advanced computer architectures.
2. To develop the design techniques of Scalable and multithreaded Architectures.
3. To Apply the concepts and techniques of parallel and advanced computer architectures to design modern computer systems

Course Outcomes: Gain knowledge of

1. Apply models of parallelism, program partitioning, and scheduling techniques to optimize execution in distributed and high-performance systems.
2. Analyze performance metrics, speedup laws, and scalability approaches to evaluate the efficiency of parallel systems.
3. Implement shared-memory architectures, pipelining, and superscalar designs to improve throughput in modern processors.
4. Evaluate interconnection networks, cache coherence, and synchronization mechanisms for reliable multiprocessor and multicomputer systems.
5. Design vector and SIMD architectures to support advanced applications in machine learning, graphics, and scientific simulations.

UNIT - I

Theory of Parallelism, Parallel computer models, The State of Computing, Multiprocessors and Multicomputers, Multivector and SIMD Computers, PRAM and VLSI models, Architectural development tracks, Program and network properties, Conditions of parallelism, Program partitioning and Scheduling, Program flow Mechanisms, System interconnect Architectures.

UNIT - II

Principles of Scalable performance, Performance metrics and measures, Parallel Processing applications, Speed up performance laws, Scalability Analysis and Approaches, Hardware Technologies, Processes and Memory Hierarchy, Advanced Processor Technology, Superscalar and Vector Processors

UNIT - III

Shared-Memory Organizations, Sequential and weak consistency models, Pipelining and superscalar techniques, Linear Pipeline Processors, Non-Linear Pipeline Processors, Instruction Pipeline design, Arithmetic pipeline design, superscalar pipeline design.

UNIT - IV

Parallel and Scalable Architectures, Multiprocessors and Multicomputers, Multiprocessor system interconnects, cache coherence and synchronization mechanism, Three Generations of Multicomputers, Message-passing Mechanisms, Multivetor and SIMD computers.

UNIT - V

Vector Processing Principles, Multivector Multiprocessors, Compound Vector processing, SIMD computer Organizations, The connection machine CM-5.

TEXT BOOK:

1. Advanced Computer Architecture, Kai Hwang, 2nd Edition, Tata McGraw Hill Publishers.

REFERENCES:

1. Computer Architecture, J.L. Hennessy and D.A. Patterson, 4th Edition, ELSEVIER.
2. Advanced Computer Architectures, S.G.Shiva, Special Indian edition, CRC, Taylor &Francis.
3. Introduction to High Performance Computing for Scientists and Engineers, G. Hager and G. Wellein, CRC Press, Taylor & Francis Group.
4. Advanced Computer Architecture, D. Sima, T. Fountain, P. Kacsuk, Pearson education.
5. Computer Architecture, B. Parhami, Oxford Univ. Press.

ENTERPRISE CLOUD CONCEPTS (PROFESSIONAL ELECTIVE - III)

M.Tech CSE I Year II Sem.

L	T	P	C
3	0	0	3

Course Objectives: Knowledge on significance of cloud computing and its fundamental concepts and models.

Course Outcomes:

1. Design cloud delivery and deployment models for scalable e-learning and online retail platforms using public or hybrid clouds.
2. Analyze virtualization and resource replication mechanisms to optimize data center utilization in IT enterprises.
3. Evaluate cloud management systems and resource monitoring to ensure compliance and cost-effectiveness in enterprise cloud adoption.
4. Differentiate fundamental cloud architecture patterns to determine their suitability for handling dynamic financial transaction workloads.
5. Formulate cloud-centric enterprise strategies by integrating smart enterprise mechanisms to transform sectors such as healthcare and banking.

Unit - I

Understanding Cloud Computing:

Origins and influences, Basic Concepts and Terminology, Goals and Benefits, Risks and Challenges.

Fundamental Concepts and Models:

Roles and Boundaries, Cloud Characteristics, Cloud Delivery Models, Cloud Deployment Models.

Unit - II

Cloud-Enabling Technology:

Broadband Networks and Internet Architecture, Data Center Technology, Virtualization Technology

CLOUD COMPUTING MECHANISMS:

Cloud Infrastructure Mechanisms: Logical Network Perimeter, Virtual Server, Cloud Storage Device, Cloud Usage Monitor, Resource Replication

Unit - III

Cloud Management Mechanisms: Remote Administration System, Resource Management System, SLA Management System, Billing Management System, Case Study Example

Cloud Computing Architecture

Fundamental Cloud Architectures: Workload Distribution Architecture, Resource Pooling Architecture, Dynamic Scalability Architecture, Elastic Resource Capacity Architecture, Service Load Balancing Architecture, Cloud Bursting Architecture, Elastic Disk Provisioning Architecture, Redundant Storage Architecture, Case Study Example

Unit - IV

Cloud-Enabled Smart Enterprises

Introduction, Revisiting the Enterprise Journey, Service-Oriented Enterprises, Cloud Enterprises, Smart Enterprises, The Enabling Mechanisms of Smart Enterprises

Cloud-Inspired Enterprise Transformations

Introduction, The Cloud Scheme for Enterprise Success, Elucidating the Evolving Cloud Idea, Implications of the Cloud on Enterprise Strategy, Establishing a Cloud-Incorporated Business Strategy

UNIT-V Transitioning to Cloud-Centric Enterprises

The Tuning Methodology, Contract Management in the Cloud

Cloud-Instigated IT Transformations

Introduction, Explaining Cloud Infrastructures, A Briefing on Next-Generation Services, Service Infrastructures, Cloud Infrastructures, Cloud Infrastructure Solutions, Clouds for Business Continuity, The Relevance of Private Clouds, The Emergence of Enterprise Clouds

TEXT BOOKS:

1. Erl Thomas, Puttini Ricardo, Mahmood Zaigham, Cloud Computing: Concepts, Technology & Architecture 1st Edition,
2. Pethuru Raj, Cloud Enterprise Architecture, CRC Press

REFERENCE:

1. James Bond, The Enterprise Cloud, O'Reilly Media, Inc.

CYBER SECURITY (PROFESSIONAL ELECTIVE - III)

M.Tech CSE I Year II Sem.

L	T	P	C
3	0	0	3

Course objectives:

1. To understand various types of cyber-attacks and cyber-crimes
2. To learn threats and risks within context of the cyber security
3. To have an overview of the cyber laws & concepts of cyber forensics
4. To study the defensive techniques against these attacks

Course Outcomes:

1. Design cyber security concepts and threat models for secure communication systems in protecting organizational assets.
2. Analyze cyber laws, policies, and digital forensic techniques to investigate cybercrimes in preserving admissible digital evidence.
3. Evaluate mobile and wireless device vulnerabilities to recommend effective authentication and security policies in enterprise environments.
4. Examine organizational implications of cyber threats to formulate strategies mitigating risks in web services, social media, and intellectual property.
5. Develop privacy-preserving solutions and policies for sensitive domains such as healthcare and financial systems through real-world case analysis.

UNIT -I

Introduction to Cyber Security: Basic Cyber Security Concepts, layers of security, Vulnerability, threat, Harmful acts, Internet Governance – Challenges and Constraints, Computer Criminals, CIA Triad, Assets and Threat, motive of attackers, active attacks, passive attacks, Software attacks, hardware attacks, Cyber Threats-Cyber Warfare, Cyber Crime, Cyber terrorism, Cyber Espionage, etc., Comprehensive Cyber Security Policy.

UNIT - II

Cyberspace and the Law & Cyber Forensics: Introduction, Cyber Security Regulations, Roles of International Law. The INDIAN Cyberspace, National Cyber Security Policy.

Introduction, Historical background of Cyber forensics, Digital Forensics Science, The Need for Computer Forensics, Cyber Forensics and Digital evidence, Forensics Analysis of Email, Digital Forensics Lifecycle, Forensics Investigation, Challenges in Computer Forensics

UNIT - III

Cybercrime: Mobile and Wireless Devices: Introduction, Proliferation of Mobile and Wireless Devices, Trends in Mobility, Credit card Frauds in Mobile and Wireless Computing Era, Security Challenges Posed by Mobile Devices, Registry Settings for Mobile Devices, Authentication service Security, Attacks on Mobile/Cell Phones, Organizational security Policies and Measures in Mobile Computing Era, Laptops.

UNIT- IV

Cyber Security: Organizational Implications: Introduction, cost of cybercrimes and IPR issues, web threats for organizations, security and privacy implications, social media marketing: security risks and perils for organizations, social computing and the associated challenges for organizations

UNIT - V

Privacy Issues: Basic Data Privacy Concepts: Fundamental Concepts, Data Privacy Attacks, Data linking and profiling, privacy policies and their specifications, privacy policy languages, privacy in different domains- medical, financial, etc

Cybercrime: Examples and Mini-Cases

Examples: Official Website of Maharashtra Government Hacked, Indian Banks Lose Millions of Rupees, Parliament Attack, Pune City Police Bust Nigerian Racket, e-mail spoofing instances.

Mini-Cases: The Indian Case of online Gambling, An Indian Case of Intellectual Property Crime, Financial Frauds in Cyber Domain.

TEXT BOOKS:

1. Nina Godbole and Sunit Belpure, Cyber Security Understanding Cyber Crimes, Computer Forensics and Legal Perspectives, Wiley
2. B.B. Gupta, D.P. Agrawal, Haoxiang Wang, Computer and Cyber Security: Principles, Algorithm, Applications, and Perspectives, CRC Press, ISBN 9780815371335,2018.

REFERENCES:

1. Cyber Security Essentials, James Graham, Richard Howard and Ryan Otson, CRC Press.
2. Introduction to Cyber Security, Chwan-Hwa(john) Wu,J. David Irwin, CRC Press T&FGroup.

PARALLEL COMPUTING (PROFESSIONAL ELECTIVE - III)

M.Tech CSE I Year II Sem.

L	T	P	C
3	0	0	3

Prerequisites

1. Computer Organization & Architecture
2. Operating Systems
3. Programming for problem solving

Course Objectives

1. To introduce the foundations of parallel Computing
2. To learn various parallel computing architectures and programming models
3. To gain knowledge of writing efficient parallel programs

Course Outcomes

1. Apply basic parallel programming platforms and communication operations to implement scalable scientific applications.
2. Design analytical models with parallel algorithms for program efficiency in real-time systems.
3. Evaluate message passing (MPI) and shared memory (PThreads) paradigms for suitable programming models in heterogeneous computing environments.
4. Distinguish dense matrix and sorting algorithms to optimize performance in engineering simulations and data-intensive applications.
5. Create efficient parallel graph and search algorithms to solve large-scale problems in social network analysis.

Unit I

Parallel Computing: Introduction, Motivation and scope - Parallel Programming Platforms – Basic Communication Operations

Unit II

Principles of Parallel Algorithm Design - Analytical Modelling of Parallel Programs

Unit III

Programming using Message Passing Paradigm (MPI) – Programming Shared Address Space Platforms (PThreads)

Unit IV

Dense Matric Algorithms (Matrix-Vector Multiplication, Matrix-Matrix Multiplication) – Sorting Algorithms (Issues, Bubble Sort, Quick Sort, Bucket Sort, Enumeration Sort, Radix Sort)

Unit V

Graph Algorithms (Minimum Spanning Tree: Prim's Algorithm - Single-Source Shortest Paths: Dijkstra's Algorithm) Search Algorithms (DFS, BFS)

TEXT BOOK:

1. Introduction to Parallel Computing, Second Edition, Ananth Grama, George Karypis, Vipin Kumar, Anshul Gupta, Addison-Wesley, 2003, ISBN: 0201648652

REFERENCES:

1. Parallel Computing – Theory and Practice, Second Edition, Michaek J. Quinn, Tata McGraw- Hill Edition.
2. Parallel Computers – Architectures and Programming, V. Rajaraman, C. Siva Ram Murthy, PHI.

LARGE LANGUAGE MODELS (PROFESSIONAL ELECTIVE - III)

M.Tech CSE/CS I Year II Sem.

L	T	P	C
3	0	0	3

Course Objectives

1. To introduce the foundations of transformer architectures and their evolution into LLMs.
2. To equip students with skills to train, fine-tune, and deploy LLMs for various tasks.
3. To explore ethical, legal, and societal implications of LLMs in real-world applications.
4. To expose students to state-of-the-art LLM frameworks, evaluation techniques, and research trends.

Course Outcomes (COs)

1. Understand the architecture and inner workings of transformer-based LLMs.
2. Apply prompt engineering and fine-tuning techniques for domain-specific tasks.
3. Evaluate LLM performance using standard metrics and benchmarks.
4. Identify challenges in LLM training, deployment, and scaling.
5. Analyze ethical, legal, and societal implications of LLM usage.

1. Apply transformer architectures and pretraining objectives to build baseline natural language and generation systems.
2. Analyze fine-tuning and domain adaptation methods to customize LLMs for specialized industry tasks.
3. Evaluate prompt engineering strategies to optimize performance in real-world applications as chatbots and code assistants.
4. Distinguish evaluation metrics and deployment strategies for efficient approaches in scaling LLM-based enterprise solutions.
5. Create responsible LLM-driven applications by integrating fairness, safety, and retrieval-augmented generation techniques for domains such as healthcare, education, and legal services.

UNIT 1 – Foundations of Large Language Models

Introduction to LLMs: Definition, scope, and historical evolution from statistical NLP to transformers. The Transformer architecture: Attention mechanisms, self-attention, multi-head attention. Pretraining objectives: Masked language modeling (MLM), Causal language modeling (CLM). Evolution of LLMs: BERT, GPT series, T5, LLaMA, Mistral.

UNIT 2 – Training and Fine-Tuning LLMs

Pretraining datasets and tokenization: BPE, SentencePiece, WordPiece. Fine-tuning approaches: Full fine-tuning, LoRA, adapters, instruction tuning. Domain adaptation and few-shot/zero-shot learning. Data augmentation for LLMs and prompt-based tuning.

UNIT 3 – Prompt Engineering and Applications

Principles of prompt design: Zero-shot, few-shot, and chain-of-thought prompting. System prompts, role prompting, and context length optimization.

Use cases: Text generation, summarization, code generation, question answering, chatbots. Tools & frameworks: Lang Chain, Llama Index, Hugging Face Transformers.

UNIT 4 – Evaluation and Deployment of LLMs

Evaluation metrics: Perplexity, BLEU, ROUGE, METEOR, human evaluation.

Benchmark datasets: GLUE, SuperGLUE, HELM, BIG-bench.

Deployment strategies: API-based deployment, on-prem deployment, inference optimization.

Scaling and latency considerations; quantization and pruning for LLMs.

UNIT 5 – Ethics, Safety, and Future Directions

Bias, fairness, and toxicity in LLMs. Hallucination problem and mitigation techniques. Legal and regulatory issues: Copyright, data privacy, AI Act. Trends in LLM research: Multimodal LLMs, retrieval-augmented generation (RAG), open-source LLM ecosystems.

TEXT BOOKS:

1. Vaswani, A. et al. (2017) *Attention Is All You Need* – NIPS Conference Paper.
2. Lewis, P. et al. (2021) *Language Models are Few-Shot Learners* – OpenAI Research Paper.
3. Tunstall, L., von Werra, L., & Wolf, T. (2022) *Natural Language Processing with Transformers* – O'Reilly Media.

REFERENCE BOOKS:

1. Bommasani, R. et al. (2021) *On the Opportunities and Risks of Foundation Models* – Stanford CRFM.
2. Jurafsky, D., & Martin, J. H. (2023) *Speech and Language Processing* (3rd Edition draft) – Pearson.
3. Mollick, E., & Mollick, L. (2024) *Co-Intelligence: Living and Working with AI* – Little, Brown Spark.
4. Hugging Face Documentation – *https://huggingface.co/docs/*

BIOINFORMATICS (PROFESSIONAL ELECTIVE - IV)

M.Tech CSE I Year II Sem.

L	T	P	C
3	0	0	3

Course Objectives:

Knowledge on concepts of bioinformatics and biological motivations of sequence analysis

Course Outcomes:

1. Apply XML concepts to represent and manage biological data for knowledge extraction in genomics.
2. Analyze DNA and protein sequences using Perl or Bioperl scripts to automate motif discovery and sequence manipulation in bioinformatics research.
3. Evaluate different database architectures for storing and retrieving biological information.
4. Distinguish sequence alignment algorithms and gap-penalty models to determine accurate techniques for genomic sequence comparisons.
5. Create phylogenetic trees and protein structure visualizations by integrating clustering and prediction methods to study evolutionary relationships and structural biology.

UNIT -I: The Central Dogma & XML (Bio XML) for Bioinformatics: Watson's definition, information flow, from data to knowledge, Convergence, the organization of DNA, the organization of Proteins, Introduction, Differences between HTML and XML, fundamentals of XML, fundamentals of XML namespaces. Introduction to DTDs, Document type Declarations, declaring elements, declaring attributes, working with entities XML Schemas, Essential Concepts, working with simple types, working with complex types, Basic namespaces issues.

UNIT -II: Perl (Bioperl) for Bioinformatics: Representing sequence data, program to store a DNA sequence, concatenating DNA fragments, Transcription, Calculating the reverse complement in Perl, Proteins, files, reading proteins in files, Arrays, Flow control, finding motifs, counting Nucleotides, exploding strings into arrays, operating on strings, writing to files, subroutines and bugs.

UNIT -III: Databases: Flat file, Relational, object-oriented databases, object Relational and Hypertext, Data life cycle, Database Technology, Database Architecture, Database Management Systems and Interfaces.

UNIT -IV : Sequence Alignment Algorithms: Biological motivations of sequence analysis, the models for sequence analysis and their biological motivation, global alignment, local alignment, End free-space alignment and gap penalty, Sequence Analysis tools and techniques.

UNIT -V : Phylogenetic Analysis: Introduction, methods of Phylogenetic analysis, distance methods, the neighbor- Joining (NJ) method, The Fitch/ Margoliash method, character-based methods, Other methods, Tree evaluation and problems in phylogenetic analysis, Clustering, Protein structure visualization and Protein structure prediction.

TEXT BOOKS:

1. S.C. Rastogi, N. Mendiratta, "Bioinformatics Methods and Applications", CBS publications, 2004
2. James D. Tisdall, "Beginning Perl for Bioinformatics" O'Reilly media, 1st Edition, 2001

REFERENCE BOOKS:

1. D.R. Westhead, J.H. Parish, "Bioinformatics" Viva books private limited, New Delhi (2003)
2. Att Wood, "Bioinformatics" Pearson Education, 2004
3. Bryan Bergeron, M.D, "Bioinformatics Computing" Pearson Education, 2003

AD-HOC AND SENSOR NETWORKS (PROFESSIONAL ELECTIVE - IV)

M.Tech CSE I Year II Sem.

L	T	P	C
3	0	0	3

Prerequisites: Computer Networks

Course Objectives

1. To understand the challenges of routing in ad-hoc and sensor networks
2. To understand various broadcast, mutlicast and geocasting protocols in ad hoc and sensor networks
3. To understand basics of Wireless sensors, and Lower Layer Issues and Upper Layer Issues of WSN

Course Outcomes

1. Design efficient data communication systems in mobile ad hoc networks with routing algorithms.
2. Analyze rebroadcasting and multicasting schemes to minimize broadcast storms and improve transmission efficiency in dynamic MANET environments.
3. Evaluate geocasting protocols and TCP enhancements to ensure reliable end-to-end communication in mobile and location-aware ad hoc networks.
4. Distinguish wireless sensor network architectures and lower-layer protocols to optimize communication in environmental monitoring and smart cities.
5. Create adaptive transport and application layer mechanisms to support real-time data collection and decision.

UNIT - I

Introduction to Ad Hoc Networks

Characteristics of MANETs, Applications of MANETs and Challenges of MANETs. Routing in MANETs,Criteria for classification, Taxonomy of MANET routing algorithms, *Topology-based* routing algorithms-Proactive: DSDV, WRP; Reactive: DSR, AODV, TORA; Hybrid: ZRP; *Position- based* routing algorithms-Location Services-DREAM, Quorum-based, GLS; Forwarding Strategies,Greedy Packet, Restricted Directional Flooding-DREAM, LAR; Other routing algorithms-QoS Routing, CEDAR.

UNIT - II

Data Transmission

Broadcast Storm Problem, Rebroadcasting Schemes-Simple-flooding, Probability-based Methods, Area-based Methods, Neighbour Knowledge-based: SBA, Multipoint Relaying, AHBP. Multicasting: Tree-based: AMRIS, MAODV; Mesh-based: ODMRP, CAMP; Hybrid: AMRoute, MCEDAR.

UNIT - III

Geocasting

Data-transmission Oriented-LBM; Route Creation Oriented-GeoTORA, MGR**.**
TCP over Ad Hoc TCP protocol overview, TCP and MANETs, Solutions for TCP over Ad hoc

UNIT - IV

Basics of Wireless Sensors and Lower Layer Issues

Applications, Classification of sensor networks, Architecture of sensor network, Physical layer, MAC layer, Link layer, Routing Layer.

UNIT - V

Upper Layer Issues of WSN

Transport layer, High-level application layer support, Adapting to the inherent dynamic nature of WSNs.

TEXT BOOKS:

1. Ad Hoc and Sensor Networks – Theory and Applications, *Carlos Corderio Dharma P.Aggarwal,* World Scientific Publications, March 2006, ISBN – 981-256-681-3
2. Wireless Sensor Networks: An Information Processing Approach, Feng Zhao, Leonidas Guibas, Elsevier Science, ISBN – 978-1-55860-914-3 (Morgan Kauffman)

REFERENCES:

1. C. Siva Ram Murthy, B.S. ManojAd Hoc Wireless Networks: Architectures and Protocols
2. Taieb Znati Kazem Sohraby, Daniel Minoli, Wireless Sensor Networks: Technology, Protocols and Applications, Wiley.

ROBOTIC PROCESS AUTOMATION (PROFESSIONAL ELECTIVE - IV)

M.Tech CSE I Year II Sem.

L	T	P	C
3	0	0	3

Course Objectives:

Aim of the course is to make learners familiar with the concepts of Robotic Process Automation.

Course Outcomes:

1. Apply RPA concepts and Automation Anywhere features for automating repetitive business processes.
2. Analyze web control room components to manage enterprise-level automation tasks effectively.
3. Evaluate device pools, workloads, and administration settings to ensure secure, scalable, and efficient deployment of RPA solutions in organizations.
4. Distinguish recording tools and command libraries to optimize task automation in real-world scenarios.
5. Create advanced automation workflows using object cloning, error handling, PDF/FTP/PGP integration, and workflow designers for end-to-end digital transformation.

Unit I

Introduction to Robotic Process Automation & Bot Creation Introduction to RPA and Use cases – Automation Anywhere Enterprise Platform – Advanced features and capabilities – Ways to create Bots

Unit II

Web Control Room and Client Introduction - Features Panel - Dashboard (Home, Bots, Devices, Audit, Workload, Insights) - Features Panel – Activity (View Tasks in Progress and Scheduled Tasks) - Bots (View Bots Uploaded and Credentials)

Unit III

Devices (View Development and Runtime Clients and Device Pools) - Workload (Queues and SLA Calculator) - Audit Log (View Activities Logged which are associated with Web CR) - Administration (Configure Settings, Users, Roles, License and Migration) - Demo of Exposed API's – Conclusion – Client introduction and Conclusion.

Unit IV

Bot Creator Introduction – Recorders – Smart Recorders – Web Recorders – Screen Recorders - Task Editor – Variables - Command Library – Loop Command – Excel Command – Database Command - String Operation Command - XML Command

Unit V

Terminal Emulator Command - PDF Integration Command - FTP Command - PGP Command - Object Cloning Command - Error Handling Command - Manage Windows Control Command - Workflow Designer - Report Designer

TEXT BOOKS:

1. Learning Robotic Process Automation: Create Software robots and automate business processes with the leading RPA tool - UiPath: Create Software robots. with the leading RPA tool – UiPath Kindle Edition.

REFERENCES:

1. Robotic Process Automation A Complete Guide - 2020 Edition Kindle Edition.

GENERATIVE AI (PROFESSIONAL ELECTIVE - IV)

M.Tech CSE I Year II Sem.

L	T	P	C
3	0	0	3

Course Objectives

1. To introduce the foundations, evolution, and core concepts of AI, ML, DL, NLP, and Generative AI.
2. To develop understanding of advanced neural architectures and generative models such as GANs, VAEs, and Transformers.
3. To explore Large Language Models, prompt engineering, and their real-world applications.
4. To familiarize learners with frameworks, multimodal applications, and ethical considerations in Generative AI.

Course Outcomes

1. Apply foundational AI, deep learning, and generative modeling techniques to build transfer-learning based solutions for real-world applications.
2. Analyze advanced neural architectures and Transformers to determine their effectiveness in generating realistic data across different domains.
3. Evaluate large language models and prompt engineering strategies to optimize chatbot design, creative writing, and intelligent text generation systems.
4. Distinguish multi-agent systems and collaboration frameworks to orchestrate generative AI tasks in areas such as healthcare and finance.
5. Create multimodal generative AI applications using frameworks while addressing ethical, social, and legal challenges for responsible deployment.

UNIT 1

Foundations of AI and Generative Models

Introduction and historical evolution to Artificial Intelligence (AI), Machine Learning (ML), Natural Language Processing (NLP)and Deep Learning (DL), Structure of Artificial Neural Networks (ANNs), Mathematical and computational foundations of generative modeling, Overview of generative models and their applications across various domains; Importance of Generative AI in modern applications, Transfer learning and in advancing Generative AI

UNIT 2

Advanced Neural Architectures for Generative AI

Variational Autoencoders (VAEs): principles and applications, Generative Adversarial Networks (GANs): architecture and working principles; Transformer architecture and attention mechanisms (in detail); Long Short-Term Memory Networks (LSTMs) and the limitations of traditional RNNs/LSTMs, Advanced Transformer architectures and techniques, Pre-training and transfer learning strategies for generative models

UNIT 3

Large Language Models and Prompt Engineering

Overview of Large Language Models (LLMs), GPT architecture, variants, and working principles, Pré-training and fine-tuning GPT models for applications (e.g., chatbots, text generation), Case study: GPT-based customer support chatbot, BERT architecture, pre-training objectives, and fine-tuning, Prompt Engineering: Designing effective prompts, controlling model behavior, and improving output quality, Fine-tuning language models for creative writing and chatbot development

UNIT 4

Multi-Agent Systems and Generative AI Applications

Introduction to Multi-Agent Systems (MAS),Types of agents: reactive, deliberative, hybrid, and learning agents, Multi-agent collaboration and orchestration for generative tasks, Use cases: autonomous research assistants, cooperative creative generation, distributed problem-solving, Frameworks and tools: AutoGen, CrewAI, Hugging GPT for LLM-powered multi-agent systems, Generative AI applications: Art, Creativity, Image/Video generation, Music composition, Healthcare, Finance, Real-world case studies and deployment challenges

UNIT 5

Frameworks, Multimodal Applications, and Ethics

LangChain framework: components and LLM application development, Retrieval-Augmented Generation (RAG), Embeddings, Indexing networks, and Vector databases, Generative AI across modalities: Text, Code, Image, and Video generation, Image and Video generation using GANs and VAEs, Multimodal Generative AI: integration and training strategies, Ethical considerations: bias, fairness, trust, and responsible AI deployment, Social and legal implications of Generative AI, Risk mitigation strategies and real-world ethical case studies

TEXT BOOKS

1. Altaf Rehmani, Generative AI for Everyone: Understanding the Essentials and Applications of This Breakthrough Technology.
2. Charu C. Aggarwal, Neural Networks and Deep Learning: A Textbook. Joseph Babcock, Raghav Bali, Generative AI with Python and TensorFlow 2, 2024.

REFERENCE BOOKS

1. Josh Kalin, Generative Adversarial Networks Cookbook.
2. Jesse Sprinter, Generative AI in Software Development: Beyond the Limitations of Traditional Coding, 2024.

ONLINE REFERENCES

1. Fabian Gloeckle et al., Better & Faster Large Language Models via Multi-token Prediction, arXiv:2404.19737v1, 2024.Vaswani et al., Attention Is All You Need, NeurIPS 2017.

ADVANCED ALGORITHMS LAB (LAB - III)

M.Tech CSE I Year II Sem.

L	T	P	C
0	0	4	2

Course Objective: The student can able to attain knowledge in advanced algorithms.

Course Outcomes: The student can able to analyze the performance of algorithms

1. Apply brute-force and divide-and-conquer approaches to solve computational problems such as assignment tasks and large integer multiplication in scientific computing.
2. Analyze optimization strategies including greedy methods and Gaussian elimination to design efficient solutions for knapsack and system-of-equation problems in engineering applications.
3. Evaluate matrix factorization techniques like LU decomposition and transitive closure algorithms such as Warshall's for performance in real-time data processing.
4. Differentiate among string-matching algorithms including Rabin-Karp, KMP, and Horspool to identify the most suitable method for text mining and bioinformatics applications.
5. Create network flow-based solutions by implementing maximum flow algorithms to optimize logistics, communication, and resource allocation problems.

List of Experiments

- Implement assignment problem using Brute Force method
- Perform multiplication of long integers using divide and conquer method.
- Implement a solution for the knapsack problem using the Greedy method.
- Implement Gaussian elimination method.
- Implement LU decomposition
- Implement Warshall algorithm
- Implement the Rabin Karp algorithm.
- Implement the KMP algorithm.
- Implement Harspool algorithm
- Implement max-flow problem.

TEXT BOOK:

1. Design and Analysis of Algorithms, S.Sridhar, OXFORD University Press

REFERENCES:

1. Introduction to Algorithms, second edition, T.H. Cormen, C.E. Leiserson, R.L. Rivest and C.Stein,PHI Pvt. Ltd./ Pearson Education.
2. Fundamentals of Computer Algorithms, Ellis Horowitz, Satraj Sahni and Rajasekharam, Universities Press.
3. Design and Analysis of algorithms, Aho, Ullman and Hopcroft, Pearson education.

ENTERPRISE CLOUD CONCEPTS LAB (LAB -IV)

M.Tech CSE I Year II Sem.

L	T	P	C
0	0	4	2

Course Objectives:

Knowledge on significance of cloud computing and its fundamental concepts and models.

Course Outcomes:

1. Understand importance of cloud architecture
2. Illustrating the fundamental concepts of cloud security
3. Analyze various cloud computing mechanisms
4. Understanding the architecture and working of cloud computing.

List of Experiments:

1. Understand the concepts of virtualization and configure virtual machines using VirtualBox to simulate real-world computing environments.
2. Deploy and manage Linux virtual machines, perform basic operations, and troubleshoot common issues.
3. Analyze cloud computing scenarios using CloudSim for modeling and simulation of cloud infrastructure and resource allocation.
4. Create cloud instances in OpenStack environments, understand IaaS concepts, and implement cloud-based solutions.
5. Install and configure a single-node Hadoop cluster, perform basic data processing using HDFS and MapReduce.

E-Resources:

1. https://www.iitk.ac.in/nt/faq/vbox.htm

2. https://www.google.com/urlsa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwjqrNG0za73Ah XZt1YBHZ21DWEQFnoECAMQAQ&url=http%3A%2F%2Fwww.cs.columbia.edu%2F~sedwards%2Fclasses%2F2015%2F1102-fall%2Flinuxvm.pdf&usg=AOvVaw3xZPuF5xVgk-AQnBRsTtHz

3. https://www.cloudsimtutorials.online/cloudsim/

4.https://edwardsamuel.wordpress.com/2014/10/25/tutorial-creating-openstack-instance-in-

trystack/

5. https://www.edureka.co/blog/install-hadoop-single-node-hadoop-cluster

CYBER SECURITY LAB (LAB -IV)

M.Tech CSE I Year II Sem.

L	T	P	C
0	0	4	2

Prerequisites

1. A course on "Network Security and Cryptography".

Course Objectives:

1. To get practical exposure of Cyber security threats
2. Learn about cyber forensic tools

Course Outcomes:

1.	Interpret scan results for responsible security assessment for the open services and associated vulnerabilities of the Network.
2.	Design honeypot on a network to capture attacker activity and intrusion patterns for threat intelligence.
3.	Demonstrate symmetric/asymmetric encryption, hashing, and digital/PKI signatures and verify message integrity/authenticity.
4.	Build secure passwords programmatically using OpenSSL commands and evaluate password strength against common attack models.
5.	Formulate network traffic using Wireshark to identify protocols, detect anomalies, and reconstruct sessions for forensic investigation.

List of Experiments

1. Perform an Experiment for port scanning with NMAP.
2. Setup a honeypot and monitor the honeypot on the network
3. Install Jcrpt /Cryptool tool (or any other equivalent) and demonstrate Asymmetric, Symmetric crypto algorithm, Hash and Digital/PKI signatures.
4. Generate minimum 10 passwords of length 12 characters using open SSL command
5. Perform practical approach to implement Foot printing-Gathering target information using Dmitry-Dmagic, UAtester.
6. Working with sniffers for monitoring network communication (Wireshark).
7. Use Snort to perform real time traffic analysis and packet logging.
8. Perform email analysis using Autopsy tool.
9. Perform Registry analysis and get boot time logging using process monitor tool
10. Perform File type detection using Autopsy tool
11. Perform Memory capture and analysis using FTK imager tool
12. Perform Network analysis using the Network Miner tool

TEXT BOOKS

1. Real Digital Forensics for Handheld Devices, E. P. Dorothy, Auerback Publications, 2013.
2. Handbook of Digital Forensics and Investigation, E. Casey, Academic Press, 2010

REFERENCES:

1. The Basics of Digital Forensics: The Primer for Getting Started in Digital Forensics, J. Sammons, Syngress Publishing, 2012.
2. Malware Forensics Field Guide for Windows Systems: Digital Forensics Field Guides, C. H. Malin, E. Casey and J. M. Aquilina, Syngress, 2012
3. The Best Damn Cybercrime and Digital Forensics Book Period, J. Wiles and A.Reyes, Syngress, 2007.

PARALLEL COMPUTING LAB (LAB -IV)

M.Tech CSE/CS I Year II Sem.

L	T	P	C
0	0	4	2

Prerequisites

1. Computer Organization & Architecture
2. Operating Systems
3. Programming for problem solving

Course Objectives

1. To introduce the foundations of parallel Computing
2. To learn various parallel computing architectures and programming models
3. To gain knowledge of writing efficient parallel programs

Course Outcomes

1. Implement parallel programs using MPI, OpenMP, and Pthreads to solve computational problems efficiently.
2. Apply parallel programming constructs to implement sorting and graph for performance improvement.
3. Analyze the speedup and efficiency of parallel programs, with sequential implementations to evaluate performance gains.
4. Decompose computational problems into tasks suitable for parallel execution, and synchronize tasks effectively using threads and message passing.
5. Demonstrate the use of parallel computing in real-world applications and research scenarios, including matrix computations, graph processing, and scientific simulations.

List of Programs:

1. Design a parallel program to implement Matrix-Vector and Matrix-Matrix Multiplication using MPI library.
2. Design a parallel program to implement Bubble Sort using OpenMP and Pthread Programming Constructs.
3. Design a parallel program to implement Quick Sort using OpenMP and Pthread Programming Constructs.
4. Design a parallel program to implement Bucket Sort using OpenMP and Pthread Programming Constructs.
5. Design a parallel program to implement Prim's Algorithm using OpenMP and Pthread Programming Constructs.
6. Design a parallel program to implement DFS Algorithm using OpenMP and Pthread Programming Constructs.
7. Design a parallel program to implement BFS Algorithm using OpenMP and Pthread Programming Constructs.
8. Design a parallel program to implement Dijkstra's Algorithm using MPI library.

LARGE LANGUAGE MODELS LAB (LAB -IV)

M.Tech CSE/CS I Year II Sem.

L	T	P	C
0	0	4	2

Prerequisites: Python Programming

Course Objectives:
The primary objectives of an LLM lab course typically involve both theoretical understanding and practical application skills

Course Outcomes:

1. To provide hands-on experience using tools and frameworks (e.g., Python with PyTorch or TensorFlow) to implement, train, and fine-tune LLMs.
2. To cover recent advances in LLM research and engineering, such as prompting (zero-shot, few-shot), Retrieval-Augmented Generation (RAG), and parameter-efficient fine-tuning (PEFT) methods like LoRA.
3. To teach methods for systematically evaluating and benchmarking models using various metrics and human feedback (RLHF).
4. To discuss the challenges and ethical implications of LLMs, including hallucination, bias, privacy, and safety, and how to mitigate them.
5. To learn how to conceive, design, prototype, and test software applications that leverage LLM-based services to solve real-world problems

List of Programs:

1. Write a program to implement word Tokenizer, Sentence and Paragraph Tokenizers. Check how many words are there in any corpus. Also check how many distinct words are there?
2. Write a python program to eliminate stopwords using nltk
3. Write a python program to perform tokenization by word and sentence using nltk.
4. Write a program to implement Perceptron.
5. Write a Program to implement AND OR gates using Perceptron.
6. Applying the Autoencoder algorithms for encoding the real-world data.
7. Word Analysis.
8. Word Generation.
9. Building Chunker.
10. Building POS Tagger.
11. POS Tagging: Hidden Markov Model.
12. POS Tagging: Viterbi Decoding.
13. Demonstrate the use of Expectation Maximization based clustering algorithm.

II-I

DIGITAL FORENSICS (PROFESSIONAL ELECTIVE - V)

M.Tech CSE II Year III Sem.

L	T	P	C
3	0	0	3

Pre-Requisites: Cybercrime and Information Warfare, Computer Networks

Course Objectives:

1. provides an in-depth study of the rapidly changing and fascinating field of computer forensics.
2. Combines both the technical expertise and the knowledge required to investigate, detect and prevent digital crimes.
3. Knowledge on digital forensics legislations, digital crime, forensics processes and procedures, data acquisition and validation, e-discovery tools
4. E-evidence collection and preservation, investigating operating systems and file systems, network forensics, art of steganography and mobile device forensics

Course Outcomes: On completion of the course the student should be able to

1. Differentiate various types of cybercrimes in relevant to the forensic science, computer forensics, and digital forensics.
2. Analyze cyber criminalistic processes and adopt a holistic approach to investigate digital crimes, including criminal behavior and cyber-attack patterns.
3. Apply evidence collection, management, and presentation techniques in compliance with legal standards, including court orders, probable cause, and IT Act provisions.
4. Design computer and network forensic investigations using forensic workstations, software tools, and open-source network analysis tools to preserve digital evidence.
5. Demonstrate mobile forensics techniques, tools, and emerging trends, and implement search and seizure procedures for electronic evidence on mobile devices.

UNIT - I

Digital Forensics Science: Forensics science, computer forensics, and digital forensics.

Computer Crime: Criminalistics as it relates to the investigative process, analysis of cyber criminalistics area, holistic approach to cyber-forensics

UNIT - II

Cyber Crime Scene Analysis:

Discuss the various court orders etc., methods to search and seizure electronic evidence, retrieved and un-retrieved communications, Discuss the importance of understanding what court documents would be required for a criminal investigation.

UNIT - III

Evidence Management & Presentation:

Create and manage shared folders using operating system, importance of the forensic mindset, define the workload of law enforcement, Explain what the normal case would look like, Define who should be notified of a crime, parts of gathering evidence, Define and apply probable cause.

UNIT - IV

Computer Forensics: Prepare a case, Begin an investigation, Understand computer forensics workstations and software, Conduct an investigation, Complete a case, Critique a case,

Network Forensics: open-source security tools for network forensic analysis, requirements for preservation of network data.

UNIT - V

Mobile Forensics: mobile forensics techniques, mobile forensics tools.

Legal Aspects of Digital Forensics: IT Act 2000, amendment of IT Act 2008.

Recent trends in mobile forensic technique and methods to search and seizure electronic evidence

TEXT BOOKS:

1. John Sammons, The Basics of Digital Forensics, Elsevier
2. John Vacca, Computer Forensics: Computer Crime Scene Investigation, Laxmi Publications

REFERENCES:

1. William Oettinger, Learn Computer Forensics: A beginner's guide to searching, analyzing, and securing digital evidence, Packt Publishing; 1st edition (30 April 2020), ISBN 1838648178
2. Thomas J. Holt , Adam M. Bossler, Kathryn C. Seigfried-Spellar , Cybercrime and Digital Forensics: An Introduction, Routledge.

ADVANCED OPERATING SYSTEMS (PROFESSIONAL ELECTIVE - V)

M.Tech CSE II Year I Sem.

L	T	P	C
3	0	0	3

Course Objectives

1. To study, learn, and understand the main concepts of advanced operating systems (parallel processing systems, distributed systems, real time systems, network operating systems, and open source operating systems), Hardware and software features that support these systems.

Course Outcomes

1. Understand the design approaches of advanced operating systems
2. Analyze the design issues of distributed operating systems.
3. Evaluate design issues of multi-processor operating systems.
4. Identify the requirements Distributed File System and Distributed Shared Memory.
5. Formulate the solutions to schedule the real time applications.

UNIT - I

Architectures of Distributed Systems: System Architecture Types, Distributed Operating Systems, Issues in Distributed Operating Systems, Communication Primitives. Theoretical Foundations: Inherent Limitations of a Distributed System, Lamport's Logical Clocks, Vector Clocks, Causal Ordering of Messages, Termination Detection.

UNIT - II

Distributed Mutual Exclusion: The Classification of Mutual Exclusion Algorithms, Non-Token –
Based Algorithms: Lamport's Algorithm, The Ricart-Agrawala Algorithm, Maekawa's Algorithm, Token-Based Algorithms: Suzuki-Kasami's Broadcast Algorithm, Singhal's Heurisric Algorithm, Raymond's Heuristic Algorithm.

UNIT - III

Distributed Deadlock Detection: Preliminaries, Deadlock Handling Strategies in Distributed Systems, Issues in Deadlock Detection and Resolution, Control Organizations for Distributed Deadlock Detection, Centralized- Deadlock – Detection Algorithms, Distributed Deadlock Detection Algorithms, Hierarchical Deadlock Detection Algorithms

UNIT - IV

Multiprocessor System Architectures: Introduction, Motivation for multiprocessor Systems, Basic Multiprocessor System Architectures Multi Processor Operating Systems: Introduction, Structures of Multiprocessor Operating Systems, Operating Design Issues, Threads, Process Synchronization, Processor Scheduling.
Distributed File Systems: Architecture, Mechanisms for Building Distributed File Systems, Design Issues

UNIT - V

Distributed Scheduling: Issues in Load Distributing, Components of a Load Distributed Algorithm, Stability, Load Distributing Algorithms, Requirements for Load Distributing, Task Migration, Issues in task Migration Distributed Shared Memory: Architecture and Motivation, Algorithms for Implementing DSM, Memory Coherence, Coherence Protocols, Design Issues

TEXT BOOK:

1. Advanced Concepts in Operating Systems, Mukesh Singhal, Niranjan G. Shivaratri, Tata McGraw-Hill Edition 2001.

REFERENCE BOOK:

1. Distributed Systems: Andrew S. Tanenbaum, Maarten Van Steen, Pearson Prentice Hall, Edition – 2, 2007.

QUANTUM COMPUTING (PROFESSIONAL ELECTIVE - V)

M.Tech CSE II Year III Sem.

L	T	P	C
3	0	0	3

Course Objectives

1. To introduce the fundamentals of quantum computing
2. The problem-solving approach using finite dimensional mathematics

Course Outcomes

1. Understand basics of quantum computing
2. Understand physical implementation of Qubit
3. Understand Quantum algorithms and their implementation
4. Understand the Impact of Quantum Computing on Cryptography

Unit I

Introduction to Essential Linear Algebra

Some Basic Algebra, Matrix Math, Vectors and Vector Spaces, Set Theory

Complex Numbers

Definition of Complex Numbers, Algebra of Complex Numbers, Complex Numbers Graphically, Vector Representations of Complex Numbers, Pauli Matrice, Transcendental Numbers

Unit II

Basic Physics for Quantum Computing

The Journey to Quantum, Quantum Physics Essentials, Basic Atomic Structure, Hilbert Spaces, Uncertainty, Quantum States, Entanglement

Basic Quantum Theory

Further with Quantum Mechanics, Quantum Decoherence, Quantum Electrodynamics, Quantum Chromodynamics, Feynman Diagram Quantum Entanglement and QKD, Quantum Entanglement, Interpretation, QKE

Unit III

Quantum Architecture

Further with Qubits, Quantum Gates, More with Gates, Quantum Circuits, The D-Wave Quantum Architecture

Quantum Hardware

Qubits, How Many Qubits Are Needed? Addressing Decoherence, Topological Quantum Computing, Quantum Essentials

Unit IV

Quantum Algorithms

What Is an Algorithm? Deutsch's Algorithm, Deutsch-Jozsa Algorithm, Bernstein-Vazirani Algorithm, Simon's Algorithm, Shor's Algorithm, Grover's Algorithm

Unit V

Current Asymmetric Algorithms: RSA, Diffie-Hellman, Elliptic Curve

The Impact of Quantum Computing on Cryptography: Asymmetric Cryptography, Specific Algorithms, Specific Applications.

TEXT BOOKS:

1. Nielsen M. A., Quantum Computation and Quantum Information, Cambridge University Press
2. Dr. Chuck Easttom, Quantum Computing Fundamentals, Pearson

REFERENCES:

1. Quantum Computing for Computer Scientists by Noson S. Yanofsky and Mirco A. Mannucci
2. Benenti G., Casati G. and Strini G., Principles of Quantum Computation and Information, Vol. Basic Concepts, Vol
3. Basic Tools and Special Topics, World Scientific. Pittenger A. O., An Introduction to Quantum Computing Algorithms.

PROMPT ENGINEERING (PROFESSIONAL ELECTIVE - V)

M.Tech CSE/CS II Year III Sem.

L	T	P	C
3	0	0	3

Course Objectives:

1. To introduce the principles and techniques of effective prompt engineering for generative AI models.
2. To understand the architecture, capabilities, and evolution of large language models such as GPT-3.5, GPT-4, Gemini, and LLaMA.
3. To explore standard practices in structured and unstructured text generation using tools like ChatGPT.
4. To apply chunking, tokenization, and formatting techniques for improving text generation and manipulation.
5. To understand the role of embeddings, vector databases (FAISS, Pinecone), and Retrieval-Augmented Generation (RAG) in modern NLP systems.

Course Outcomes:

1. Explain and apply the core principles of prompt engineering for guiding generative AI outputs effectively.
2. Describe the underlying architecture and functionality of state-of-the-art large language models (LLMs).
3. Generate and manipulate structured outputs (JSON, YAML, CSV) using ChatGPT with advanced prompting techniques.
4. Implement text chunking, tokenization, and format control using tools like SpaCy, Tiktoken, and Python.
5. Utilize vector databases such as FAISS and Pinecone in Retrieval-Augmented Generation (RAG) pipelines for efficient information retrieval.

UNIT – I

Fundamentals and Principles of Prompting

Overview of the Five Principles of Prompting: Give Direction, Specify Format, Provide Examples, Evaluate Quality, Divide Labor.

UNIT – II

Introduction to Large Language Models for Text Generation

What Are Text Generation Models, Vector Representations: The Numerical Essence of Language, Transformer Architecture: Orchestrating Contextual Relationships, Probabilistic Text Generation: The Decision Mechanism, Historical Underpinnings: The Rise of Transformer Architectures, OpenAI's Generative Pretrained Transformers, GPT-3.5-turbo and ChatGPT, GPT-4, Google's Gemini, Meta's Llama and Open Source.

UNIT – III

Standard Practices for Text Generation with ChatGPT- Part-A

Generating Lists, Hierarchical List Generation, When to Avoid Using Regular Expressions, Generating JSON, YAML Filtering YAML Payloads, Handling Invalid Payloads in YAML, Diverse Format Generation with ChatGPT, Mock CSV Data, Universal Translation Through LLMs, Ask for Context, Text Style Unbundling, Identifying the Desired Textual Features, Generating New Content with the Extracted Features, Extracting Specific Textual Features with LLMs.

UNIT – IV

Standard Practices for Text Generation with ChatGPT- Part-B

Chunking Text, Benefits of Chunking Text, Scenarios for Chunking Text, Poor Chunking Example, Chunking Strategies, Sentence Detection Using SpaCy, building a Simple Chunking Algorithm in Python, Sliding Window Chunking, Text Chunking Packages, Text Chunking with Tiktoken, Encodings, Understanding the Tokenization of Strings.

UNIT – V

Vector Databases with FAISS and Pinecone

Retrieval Augmented Generation (RAG), Introducing Embeddings, Document Loading

Memory Retrieval with FAISS, RAG with Lang Chain, Hosted Vector Databases with Pinecone, Self- Querying, Alternative Retrieval Mechanisms.

TEXTBOOK:

1. Phoenix J, Taylor M. Prompt engineering for generative AI. " O'Reilly Media, Inc."; 2024 May 16.

REFERENCES:

1. Tunstall L, Von Werra L, Wolf T. Natural language processing with transformers. " O'Reilly Media, Inc."; 2022 Jan 26.
2. Foster D. Generative deep learning. " O'Reilly Media, Inc."; 2022 Jun 28.

INTELLECTUAL PROPERTY RIGHTS (OPEN ELECTIVE)

M.Tech CSE II Year I Sem.

L	T	P	C
3	0	0	3

Course Outcomes:

1. Categorize types of IPR in real-world business, research, and innovation scenarios.
2. **Demonstrate** the process of trademark selection, evaluation, and registration through case-based applications.
3. **Analyze** copyright and patent ownership, rights, and registration processes to resolve issues related to originality, reproduction, and transfer of intellectual property.
4. **Distinguish** between lawful and unlawful business practices in relevant to trade secret laws and unfair competition.
5. **Evaluate** emerging trends and international developments in trademarks, copyrights, patents, and trade secrets to recommend effective IP strategies for organizations.

UNIT – I

Introduction to Intellectual property: Introduction, types of intellectual property, international organizations, agencies and treaties, importance of intellectual property rights.

UNIT – II

Trade Marks: Purpose and function of trademarks, acquisition of trade mark rights, protectable matter, selecting, and evaluating trade mark, trade mark registration processes.

UNIT – III

Law of copyrights: Fundamental of copyright law, originality of material, rights of reproduction, rights to perform the work publicly, copyright ownership issues, copyright registration, notice of copyright, international copyright law.

Law of patents: Foundation of patent law, patent searching process, ownership rights and transfer

UNIT – IV

Trade Secrets: Trade secret law, determination of trade secret status, liability for misappropriations of trade secrets, protection for submission, trade secret litigation.

Unfair competition: Misappropriation right of publicity, false advertising.

UNIT – V

New development of intellectual property: new developments in trade mark law; copyright law, patent law, intellectual property audits.

International overview on intellectual property, international – trade mark law, copyright law, international patent law, and international development in trade secrets law.

TEXT & REFERENCE BOOKS:

1. Intellectual property right, Deborah. E. Bouchoux, Cengage learning.
2. Intellectual property right – Unleashing the knowledge economy, prabuddha ganguli, Tata McGraw Hill Publishing company ltd

GENERATIVE AI (OPEN ELECTIVE)

M.Tech CSE II Year I Sem.

L	T	P	C
3	0	0	3

Course Objectives

1. To introduce the foundations, evolution, and core concepts of AI, ML, DL, NLP, and Generative AI.
2. To develop understanding of advanced neural architectures and generative models such as GANs, VAEs, and Transformers.
3. To explore Large Language Models, prompt engineering, and their real-world applications.
4. To familiarize learners with frameworks, multimodal applications, and ethical considerations in Generative AI.

Course Outcomes

1. Design generative models using neural network architectures and transfer learning approaches.
2. **Implement** advanced neural architectures and Transformers to solve generative AI tasks in domains of text, image, and video.
3. **Evaluate** prompt engineering techniques for effective model control and fine-tuning in working principles of Large Language Models (LLMs)
4. **Examine** multi-agent system architectures and their orchestration for generative AI applications.
5. **Evaluate** multimodal frameworks, tools, and ethical considerations of Generative AI to recommend responsible, fair, and trustworthy deployment strategies in real-world scenarios.

UNIT 1

Foundations of AI and Generative Models

Introduction and historical evolution to Artificial Intelligence (AI), Machine Learning (ML), Natural Language Processing (NLP)and Deep Learning (DL), Structure of Artificial Neural Networks (ANNs), Mathematical and computational foundations of generative modeling, Overview of generative models and their applications across various domains; Importance of Generative AI in modern applications, Transfer learning and in advancing Generative AI

UNIT 2

Advanced Neural Architectures for Generative AI

Variational Autoencoders (VAEs): principles and applications, Generative Adversarial Networks (GANs): architecture and working principles; Transformer architecture and attention mechanisms (in detail); Long Short-Term Memory Networks (LSTMs) and the limitations of traditional RNNs/LSTMs, Advanced Transformer architectures and techniques, Pre-training and transfer learning strategies for generative models

UNIT 3

Large Language Models and Prompt Engineering

Overview of Large Language Models (LLMs), GPT architecture, variants, and working principles, Pré-training and fine-tuning GPT models for applications (e.g., chatbots, text generation), Case study: GPT-based customer support chatbot, BERT architecture, pre-training objectives, and fine-tuning, Prompt Engineering: Designing effective prompts, controlling model behavior, and improving output quality, Fine-tuning language models for creative writing and chatbot development

UNIT 4

Multi-Agent Systems and Generative AI Applications

Introduction to Multi-Agent Systems (MAS),Types of agents: reactive, deliberative, hybrid, and learning agents, Multi-agent collaboration and orchestration for generative tasks, Use cases: autonomous research assistants, cooperative creative generation, distributed problem-solving, Frameworks and tools: AutoGen, CrewAI, Hugging GPT for LLM-powered multi-agent systems, Generative AI applications: Art, Creativity, Image/Video generation, Music composition, Healthcare, Finance, Real-world case studies and deployment challenges

UNIT 5

Frameworks, Multimodal Applications, and Ethics

LangChain framework: components and LLM application development, Retrieval-Augmented Generation (RAG), Embeddings, Indexing networks, and Vector databases, Generative AI across modalities: Text, Code, Image, and Video generation, Image and Video generation using GANs and VAEs, Multimodal Generative AI: integration and training strategies, Ethical considerations: bias,fairness, trust, and responsible AI deployment, Social and legal implications of Generative AI, Risk mitigation strategies and real-world ethical case studies

TEXT BOOKS:

1. Altaf Rehmani, *Generative AI for Everyone: Understanding the Essentials and Applications of This Breakthrough Technology*.
2. Charu C. Aggarwal, *Neural Networks and Deep Learning: A Textbook*.Joseph Babcock, Raghav Bali, *Generative AI with Python and TensorFlow 2*, 2024.

REFERENCE BOOKS:

1. Josh Kalin, Generative Adversarial Networks Cookbook.
2. Jesse Sprinter, Generative AI in Software Development: Beyond the Limitations of Traditional Coding, 2024.

ONLINE REFERENCES:

1. Fabian Gloeckle et al., Better & Faster Large Language Models via Multi-token Prediction, arXiv:2404.19737v1, 2024.Vaswani et al., Attention Is All You Need, NeurIPS 2017.

INTRUSION DETECTION SYSTEMS (OPEN ELECTIVE)

M.Tech CSE II Year III Sem.

L	T	P	C
3	0	0	3

Prerequisites: Computer Networks, Computer Programming

Course Objectives:

1. Compare alternative tools and approaches for Intrusion Detection through quantitative analysis to determine the best tool or approach to reduce risk from intrusion.
2. Identify and describe the parts of all intrusion detection systems and characterize new and emerging IDS technologies according to the basic capabilities all intrusion detection systems share.

Course Outcomes:

1. Apply fundamental security concepts including firewalls, VPNs, and vulnerability assessment to mitigate threats against computers and networked systems.
2. Demonstrate classes of attacks across network, application, and human layers, and relate them to corresponding attacker profiles and automated threats.
3. Analyze intrusion detection models using signature-based solutions and assess cost-sensitive in IDS performance.
4. Examine anomaly detection algorithms based on software vulnerabilities and payload analysis.
5. Evaluate malware detection strategies, alert correlation, insider threats, and evolving techniques to recommend future collaborative security approaches.

UNIT - I

The state of threats against computers, and networked systems-Overview of computer security solutions and why they fail-Vulnerability assessment, firewalls, VPN's -Overview of Intrusion Detection and Intrusion Prevention, Network and Host-based IDS

UNIT - II

Classes of attacks - Network layer: scans, denial of service, penetration Application layer: software exploits, code injection-Human layer: identity theft, root access-Classes of attackers-Kids/hackers/sop Hesitated groups-Automated: Drones, Worms, Viruses

UNIT - III

A General IDS model and taxonomy, Signature-based Solutions, Snort, Snort rules, Evaluation of IDS, Cost sensitive IDS

UNIT - IV

Anomaly Detection Systems and Algorithms-Network Behavior Based Anomaly Detectors (rate based)-Host-based Anomaly Detectors-Software Vulnerabilities-State transition, Immunology, Payload Anomaly Detection

UNIT - V

Attack trees and Correlation of alerts- Autopsy of Worms and Botnets-Malware detection - Obfuscation, polymorphism- Document vectors.

Email/IM security issues-Viruses/Spam-From signatures to thumbprints to zero day detection-Insider Threat issues-Taxonomy-Masquerade and Impersonation Traitors, Decoys and Deception-Future: Collaborative Security

TEXT BOOKS:

1. Peter Szor, The Art of Computer Virus Research and Defense, Symantec Press ISBN 0-321-30545-3.
2. Markus Jakobsson and Zulfikar Ramzan, Crimeware, Understanding New Attacks and Defenses.

REFERENCE BOOKS:

1. Saiful Hasan, Intrusion Detection System, Kindle Edition.
2. Ankit Fadia, Intrusion Alert: An Ethical Hacking Guide to Intrusion Detection.

Online Websites/Materials:

1. https://www.intechopen.com/books/intrusion-detection-systems/ Online Courses:
1. https://www.sans.org/course/intrusion-detection-in-depth
2. https://www.cybrary.it/skill-certification-course/ids-ips-certification-training-course

DIGITAL FORENSICS (OPEN ELECTIVE)

M.Tech CSE II Year III Sem.

L	T	P	C
3	**0**	**0**	**3**

Pre-Requisites: Cybercrime and Information Warfare, Computer Networks

Course Objectives:

1. provides an in-depth study of the rapidly changing and fascinating field of computer forensics.
2. Combines both the technical expertise and the knowledge required to investigate, detect and prevent digital crimes.
3. Knowledge on digital forensics legislations, digital crime, forensics processes and procedures, data acquisition and validation, e-discovery tools
4. E-evidence collection and preservation, investigating operating systems and file systems, network forensics, art of steganography and mobile device forensics

Course Outcomes: On completion of the course the student should be able to

1. Apply the principles of forensic science, computer forensics, and digital forensics to cybercrimes using a holistic investigative approach.
2. Demonstrate procedures for cyber-crime scene analysis, including search, seizure, and handling of electronic evidence in compliance with legal requirements.
3. Analyze evidence management and presentation techniques and law enforcement procedures for effective investigation.
4. Examine methodologies for conducting computer and network forensic investigations using forensic workstations, open-source tools, and case study evaluation.
5. Evaluate mobile forensic techniques, legal aspects of digital forensics and recent trends to recommend responsible and lawful practices in investigations.

UNIT - I

Digital Forensics Science: Forensics science, computer forensics, and digital forensics.
Computer Crime: Criminalistics as it relates to the investigative process, analysis of cyber criminalistics area, holistic approach to cyber-forensics

UNIT - II

Cyber Crime Scene Analysis:
Discuss the various court orders etc., methods to search and seizure electronic evidence, retrieved and un-retrieved communications, Discuss the importance of understanding what court documents would be required for a criminal investigation.

UNIT - III

Evidence Management & Presentation:
Create and manage shared folders using operating system, importance of the forensic mindset, define the workload of law enforcement, Explain what the normal case would look like, Define who should be notified of a crime, parts of gathering evidence, Define and apply probable cause.

UNIT - IV

Computer Forensics: Prepare a case, Begin an investigation, Understand computer forensics workstations and software, Conduct an investigation, Complete a case, Critique a case,
Network Forensics: open-source security tools for network forensic analysis, requirements for preservation of network data.

UNIT - V

Mobile Forensics: mobile forensics techniques, mobile forensics tools.

Legal Aspects of Digital Forensics: IT Act 2000, amendment of IT Act 2008.

Recent trends in mobile forensic technique and methods to search and seizure electronic evidence

TEXT BOOKS:

1. John Sammons, The Basics of Digital Forensics, Elsevier
2. John Vacca, Computer Forensics: Computer Crime Scene Investigation, Laxmi Publications

REFERENCES:

1. William Oettinger, Learn Computer Forensics: A beginner's guide to searching, analyzing, and securing digital evidence, Packt Publishing; 1st edition (30 April 2020), ISBN 1838648178

Thomas J. Holt, Adam M. Bossler, Kathryn C. Seigfried-Spellar , Cybercrime and Digital Forensics: An Introduction, Routledge.