

**J.B. INSTITUTE OF ENGINEERING AND TECHNOLOGY**

(UGC AUTONOMOUS)

**Bhaskar Nagar, Moinabad Mandal, R.R. District, Hyderabad -500075****DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING****M. Tech - COMPUTER SCIENCE AND ENGINEERING (CSE): R-25****COURSE STRUCTURE (2025-2026)**

<b>I YEAR-I SEMESTER</b>						
<b>S.No</b>	<b>Course Code</b>	<b>Course Title</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>Credits</b>
1.	NM71A	Mathematical Foundations of Computer Science	3	0	0	3
2.	NM71B	Advanced Data Structures	3	0	0	3
3.	NM71C	1. Database Programming with PL/SQL	3	0	0	3
	NM71D	2. Deep Learning				
	NM71E	3. Natural Language Processing				
	NM71F	4. Advanced UNIX Programming				
4.	NM71G	1. Applied Cryptography	3	0	0	3
	NM71H	2. Software Quality Engineering				
	NM71I	3. Mining Massive Datasets				
	NM71J	4. Agile Methodologies				
5.	NM711	Advanced Data Structures Lab	0	0	4	2
6.		Professional Elective - I Lab	0	0	4	2
7.	NME1A	Research Methodology & IPR	2	0	0	2
8.		Audit Course- I	2	0	0	0
		<b>Total</b>	<b>16</b>	<b>0</b>	<b>8</b>	<b>18</b>

<b>I YEAR-II SEMESTER</b>						
<b>S.No.</b>	<b>Course Code</b>	<b>Course Title</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>Credits</b>
1.	NM72A	Advanced Algorithms	3	0	0	3
2.	NM72B	Advanced Computer Architecture	3	0	0	3
3.	NM72C	1. Enterprise Cloud Concepts	3	0	0	3
	NM72D	2. Cyber Security				
	NM72E	3. Parallel computing				

**J.B. INSTITUTE OF ENGINEERING AND TECHNOLOGY**

(UGC AUTONOMOUS)

**Bhaskar Nagar, Moinabad Mandal, R.R. District, Hyderabad -500075****DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING****M. Tech - COMPUTER SCIENCE AND ENGINEERING (CSE): R-25****COURSE STRUCTURE (2025-2026)**

	NM72F	4. Large Language Models				
4.	NM72G	1. Bioinformatics	3	0	0	3
	NM72H	2. Adhoc Sensor Networks				
	NM72I	3. Robotic Process Automation				
	NM72J	4. Generative AI				
5.	NM721	Advanced Algorithms Lab	0	0	4	2
6.		Professional Elective - III Lab	0	0	4	2
7.	NM726	Mini Project with Seminar	0	0	4	2
8.		Audit Course- II	2	0	0	0
		<b>Total</b>	<b>14</b>	<b>0</b>	<b>12</b>	<b>18</b>

**II YEAR-I SEMESTER**

S.No.	Course Code	Course Title	L	T	P	Credits
1.	NM73A	1. Digital Forensics	3	0	0	3
	NM73B	2. Advanced Operating Systems				
	NM73C	3. Quantum Computing				
	NM73D	4. Prompt Engineering				
2.		Open Elective	3	0	0	3
3.	NM731	Dissertation Work Review – II	0	0	18	6
		<b>Total</b>	<b>6</b>	<b>0</b>	<b>18</b>	<b>12</b>

**II YEAR-II SEMESTER**

S.No.	Course Code	Course Title	L	T	P	Credits
1.	NM741	Dissertation Work Review – III	0	0	18	6
2.	NM742	Dissertation Viva-Voce	0	0	42	14
		<b>Total</b>	<b>0</b>	<b>0</b>	<b>60</b>	<b>20</b>



# **J.B. INSTITUTE OF ENGINEERING AND TECHNOLOGY**

(UGC AUTONOMOUS)

**Bhaskar Nagar, Moinabad Mandal, R.R. District, Hyderabad -500075**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**M. Tech - COMPUTER SCIENCE AND ENGINEERING (CSE): R-25**

**COURSE STRUCTURE (2025-2026)**

## **Open Electives for other Departments:**

1. Intellectual Property Rights
2. Generative AI
3. Intrusion Detection Systems
4. Digital Forensics

## **Audit Course I&II:**

1. English for Research Paper Writing
2. Disaster Management
3. Sanskrit for Technical Knowledge
4. Value Education
5. Constitution of India
6. Pedagogy Studies
7. Stress Management by yoga
8. Personality Development Through Life Enlightenment Skills

<b>AY: 2025-26</b> Onwards	<b>J. B. Institute of Engineering and Technology</b> (UGC Autonomous)	<b>M. Tech CSE</b> <b>I Year-I Sem</b>			
<b>Course Code:</b> <b>NM71A</b>	<b>Mathematical Foundations of Computer Science</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>

**Pre-Requisites:** An understanding of Math in general is sufficient

#### **Unit 1:**

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 2:**

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 3:**

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 4:**

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 5:**

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 I. Mott, Abraham Kandel, Theodore P. Baker, Prentis Hall of India, 2nd ed.

### Reference Books

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.

### Course Objectives:

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. Ability to understand and construct precise mathematical proofs.
2. Ability to use logic and set theory to formulate precise statements.
3. Ability to analyze and solve counting problems on finite and discrete structures.
4. Ability to describe and manipulate sequences.
5. Ability to apply graph theory in solving computing problems.

<b>AY: 2025-26</b> Onwards	<b>J. B. Institute of Engineering and Technology</b> (UGC Autonomous)	<b>M. Tech CSE</b> <b>I Year-I Sem</b>			
<b>Course Code:</b> <b>NM71B</b>	<b>ADVANCED DATA STRUCTURES</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>

**Pre-Requisites:** A course on Data Structures.

#### **Unit 1:**

Heap Structures

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

#### **Unit 2:**

Hashing and Collisions

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

#### **Unit 3:**

Search Structures

OBST, AVL trees, Red-Black trees, Splay trees,

Multiway Search Trees

B-trees, 2-3 trees

#### **Unit 4:**

Digital Search Structures

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

#### **Unit 5:**

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

#### **Reference Books**

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.

#### **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. Ability to select the data structures that efficiently model the information in a problem.
2. Ability to understand how the choice of data structures impact the performance of programs.
3. Design programs using a variety of data structures, including hash tables, search structures and digital search structures.

<b>AY: 2025-26</b> Onwards	<b>J. B. Institute of Engineering and Technology</b> (UGC Autonomous)	<b>M. Tech CSE</b> <b>I Year-I Sem</b>			
<b>Course Code:</b> <b>NM71C</b>	<b>DATABASE PROGRAMMING WITH PL/SQL</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>

### Pre-Requisites:

#### Unit 1: 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 2: 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 3: 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 4: 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 5: 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

#### Reference Books

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.



**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. Understand importance of PL/SQL basics
2. Implement functions and procedures using PL/SQL
3. Understand the importance of triggers in database

<b>AY: 2025-26</b> Onwards	<b>J. B. Institute of Engineering and Technology</b> (UGC Autonomous)	<b>M. Tech CSE</b> <b>I Year-I Sem</b>			
<b>Course Code:</b> <b>NM71D</b>	<b>DEEP LEARNING</b> (PROFESSIONAL ELECTIVE - I)	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>

#### **Pre-Requisites:**

1. Strong fundamentals in Linear Algebra
2. Basic knowledge of Probability and Statistics
3. Understanding of Calculus
4. Proficiency in Python programming

#### **Unit 1: 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 2: 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 3: 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 4: 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 5: 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.

#### **Reference Books**

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.

**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. Implement deep learning algorithms, understand neural networks and traverse the layers of data
2. Learn topics such as convolutional neural networks, recurrent neural networks, training deep networks and high-level interfaces
3. Understand applications of Deep Learning to Computer Vision
4. Understand and analyze Applications of Deep Learning to NLP

<b>AY: 2025-26</b> Onwards	<b>J. B. Institute of Engineering and Technology</b> (UGC Autonomous)	<b>M. Tech CSE</b> <b>I Year-I Sem</b>			
<b>Course Code:</b> <b>NM71E</b>	<b>NATURAL LANGUAGE PROCESSING</b> (PROFESSIONAL ELECTIVE - I)	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>

**Pre-Requisites:** Data structures, finite automata and probability theory.

#### **Unit 1:**

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 2:**

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 3:**

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

#### **Unit 4:**

Predicate-Argument Structure, Meaning Representation Systems, Software

#### **Unit 5:**

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.

#### **Reference Books**

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

#### **Course Objectives:**

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

#### **Course outcomes:**

1. Show sensitivity to linguistic phenomena and an ability to model them with formal grammars.
2. Understand and carry out proper experimental methodology for training and evaluating empirical NLP systems

3. Able to manipulate probabilities, construct statistical models over strings and trees, and estimate parameters using supervised and unsupervised training methods.
4. Able to design, implement, and analyze NLP algorithms Able to design different language modeling Techniques.
5. Able to design different language modeling Techniques.

<b>AY: 2025-26</b> Onwards	<b>J. B. Institute of Engineering and Technology</b> (UGC Autonomous)	<b>M. Tech CSE</b> <b>I Year-I Sem</b>			
<b>Course Code:</b> <b>NM71F</b>	<b>ADVANCED UNIX PROGRAMMING</b> (PROFESSIONAL ELECTIVE - I)	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>

### Pre-Requisites:

1. Basic knowledge of UNIX/Linux operating systems
2. Familiarity with C programming

### Unit 1:

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 2:

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 3:

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 4:

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 5:**

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.

#### **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. Work confidently in Linux environment.
2. Work with shell script to automate different tasks as Linux administration.

<b>AY: 2025-26</b> Onwards	<b>J. B. Institute of Engineering and Technology</b> (UGC Autonomous)	<b>M. Tech CSE</b> <b>I Year-I Sem</b>			
<b>Course Code:</b> <b>NM71G</b>	<b>APPLIED CRYPTOGRAPHY</b> (PROFESSIONAL ELECTIVE - II)	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>

**Pre-Requisites:**

1. Basic understanding of Computer Networks
2. Fundamentals of Mathematics
3. Knowledge of Discrete Mathematics

**Unit 1:**

**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 2:**

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 3: 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 4: 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 5: 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).

**Course Objectives:**

1. Knowledge on significance of cryptographic protocols and symmetric and public key algorithms

**Course outcomes:**

1. Understand the various cryptographic protocols
2. Analyze key length and algorithm types and modes
3. Illustrate different public key algorithms in cryptosystems
4. Understand special algorithms for protocols and usage in the real world.

<b>AY: 2025-26</b> Onwards	<b>J. B. Institute of Engineering and Technology</b> (UGC Autonomous)	<b>M. Tech CSE</b> <b>I Year-I Sem</b>			
<b>Course Code:</b> <b>NM71H</b>	<b>SOFTWARE QUALITY ENGINEERING</b> (PROFESSIONAL ELECTIVE - II)	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>

#### **Pre-Requisites:**

1. Basic knowledge of Software Engineering
2. Understanding of Software Development Life Cycle
3. Familiarity with Software Testing fundamentals

#### **Unit 1: 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 2: 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 3: Quality Engineering**

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

#### **Unit 4: 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 5: Coverage 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**

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

#### **Course Objectives:**

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

#### **Course outcomes:**

1. Understand software quality and its perspectives
2. Analyze defect prevention and defect reduction in software quality assurance
3. Illustrate software quality engineering activities and its process

<b>AY: 2025-26</b> Onwards	<b>J. B. Institute of Engineering and Technology</b> (UGC Autonomous)	<b>M. Tech CSE</b> <b>I Year-I Sem</b>			
<b>Course Code:</b> <b>NM711</b>	<b>MINING MASSIVE DATASETS</b> (PROFESSIONAL ELECTIVE - II)	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>

### Pre-Requisites:

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

### Unit 1: 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 2: 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 3: 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 4: 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 5: 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, 3<sup>rd</sup> 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.

**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. Handle massive data using MapReduce.
2. Develop and implement algorithms for massive data sets and methodologies in the context of data mining.
3. Understand the algorithms for extracting models and information from large datasets
4. Develop recommendation systems.
5. Gain experience in matching various algorithms for particular classes of problems.

<b>AY: 2025-26</b> Onwards	<b>J. B. Institute of Engineering and Technology</b> (UGC Autonomous)	<b>M. Tech CSE</b> <b>I Year-I Sem</b>			
<b>Course Code:</b> <b>NM71J</b>	<b>AGILE METHODOLOGIES</b> (PROFESSIONAL ELECTIVE - II)	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>

### Pre-Requisites:

Basic understanding of Software Engineering concepts

### Unit 1: 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.

### Course Objectives:

1. Knowledge on concepts of Agile development, releasing, planning and developing

**Course outcomes:**

1. Understand basic concepts of agile methods and extreme programming
2. Analyze real customer involvement and ubiquitous language
3. Discuss risk management and iteration planning
4. Summarize incremental requirements, refactoring, incremental design and architecture

<b>AY: 2025-26</b> Onwards	<b>J. B. Institute of Engineering and Technology</b> (UGC Autonomous)	<b>M. Tech CSE</b> <b>I Year-I Sem</b>			
<b>Course Code:</b> <b>NM711</b>	<b>ADVANCED DATA STRUCTURES LAB</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>0</b>	<b>0</b>	<b>4</b>	<b>2</b>

**Pre-Requisites:** A course on Computer Programming & Data Structures

#### **Experiment 1:**

Write a program to perform the following Operation

- Insert an element into a binary search tree.
- Delete an element from a binary search tree.
- Search for a key element in a binary search tree.

#### **Experiment 2:**

Write a program for implementing the following sorting methods:

- Merge sort
- Heap sort
- Quick sort

#### **Experiment 3:**

Write a program to perform the following operations:

- Insert an element into a B- tree.
- Delete an element from a B- tree.
- Search for a key element in a B- tree.

#### **Experiment 4:**

Write a program to perform the following operations:

- Insert an element into a Min-Max heap
- Delete an element from a Min-Max heap
- Search for a key element in a Min-Max heap

#### **Experiment 5:**

Write a program to perform the following operations:

- Insert an element into a Leftist tree
- Delete an element from a Leftist tree
- Search for a key element in a Leftist tree

#### **Experiment 6:**

Write a program to perform the following operations:

- Insert an element into a binomial heap
- Delete an element from a binomial heap.
- Search for a key element in a binomial heap

#### **Experiment 7:**

Write a program to perform the following operations:

- Insert an element into a AVL tree.
- Delete an element from a AVL search tree.
- Search for a key element in a AVL search tree.

**Experiment 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

**Experiment 9:**

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

**Experiment 10:**

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

**Experiment 11:**

Write a program for implementing Brute Force pattern matching algorithm

**Experiment 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.

**Reference Books**

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

**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, and B-trees.
4. Introduces sorting and pattern matching algorithms.

**Course outcomes:**

1. Ability to select the data structures that efficiently model the information in a problem.
2. Ability to assess efficiency trade-offs among different data structure implementations or combinations.
3. Implement and know the application of algorithms for sorting and pattern matching.
4. Design programs using a variety of data structures, including hash tables, binary and general tree structures, search trees, tries, heaps, graphs, and B-trees.



<b>AY: 2025-26</b> Onwards	<b>J. B. Institute of Engineering and Technology</b> (UGC Autonomous)	<b>M. Tech CSE</b> <b>I Year-I Sem</b>			
<b>Course Code:</b> <b>NM712</b>	<b>DATABASE PROGRAMMING WITH PL/SQL</b> <b>LAB (LAB - II)</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>0</b>	<b>0</b>	<b>4</b>	<b>2</b>

### Pre-Requisites:

1. Basic understanding of Database Management Systems (DBMS)
2. Knowledge of SQL (DDL, DML, DCL, TCL)

#### Experiment 1:

Write a PL/SQL program using FOR loop to insert ten rows into a database table.

#### Experiment 2:

Given the table EMPLOYEE (EmpNo, Name, Salary, Designation, DeptID), write a cursor to select the five highest paid employees from the table.

#### Experiment 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.

#### Experiment 4:

Given an integer i, write a PL/SQL procedure to insert the tuple (i, 'xxx') into a given relation.

#### Experiment 5:

Write a PL/SQL program to demonstrate Exceptions.

#### Experiment 6:

Write a PL/SQL program to demonstrate Cursors.

#### Experiment 7:

Write a PL/SQL program to demonstrate Functions.

#### Experiment 8:

Write a PL/SQL program to demonstrate Packages.

#### Experiment 9:

Write PL/SQL queries to create Procedures.

#### Experiment 10:

Write PL/SQL queries to create Triggers

#### Text Books

1. Steven Feuerstein, Oracle PL/SQL Programming, O'Reilly Media
2. Michael McLaughlin, Oracle Database 12c PL/SQL Programming, McGraw-Hill Education
3. Oracle Corporation, Oracle PL/SQL Language Reference, Oracle Documentation
- 4.

**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. Understand importance of PL/SQL basics
2. Implement functions and procedures using PL/SQL
3. Understand the importance of triggers in database

<b>AY: 2025-26</b> Onwards	<b>J. B. Institute of Engineering and Technology</b> (UGC Autonomous)	<b>M. Tech CSE</b> <b>I Year-I Sem</b>			
<b>Course Code:</b> <b>NM713</b>	<b>DEEP LEARNING LAB</b> <b>(LAB - II)</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>0</b>	<b>0</b>	<b>4</b>	<b>2</b>

### Pre-Requisites:

1. Python programming
2. Basics of Machine Learning
3. Fundamentals of Neural Networks

### Experiment 1:

Setting up the Spyder IDE Environment and Executing a Python Program

### Experiment 2:

Installing Keras, Tensorflow and Pytorch libraries and making use of them

### Experiment 3:

Applying the Convolution Neural Network on computer vision problems

### Experiment 4:

Image classification on MNIST dataset (CNN model with Fully connected layer)

### Experiment 5:

Applying the Deep Learning Models in the field of Natural Language Processing

### Experiment 6:

Train a sentiment analysis model on IMDB dataset, use RNN layers with LSTM/GRU notes

### Experiment 7:

Applying the Autoencoder algorithms for encoding the real-world data

### Experiment 8:

Applying Generative Adversarial 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..

### Reference Books

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.

#### E- Resources:

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](http://www.cs.toronto.edu/~fritz/absps/imagenet.pdf)
5. <http://neuralnetworksanddeeplearning.com/>

#### 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. Upon the Successful Completion of the Course, the Students would be able to:
2. Learn the Fundamental Principles of Deep Learning.
3. Identify the Deep Learning Algorithms for Various Types of Learning Tasks in various domains.
4. Implement Deep Learning Algorithms and Solve Real-world problems

<b>AY: 2025-26</b> Onwards	<b>J. B. Institute of Engineering and Technology</b> (UGC Autonomous)	<b>M. Tech CSE</b> <b>I Year-I Sem</b>			
<b>Course Code:</b> <b>NM714</b>	<b>NATURAL LANGUAGE PROCESSING LAB</b> (LAB - II)	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>0</b>	<b>0</b>	<b>4</b>	<b>2</b>

**Pre-Requisites:** Data structures, finite automata and probability theory

**Implement the following using Python**

**Experiment 1:**

Tokenization

**Experiment 2:**

Stemming

**Experiment 3:**

Stop word removal (a, the, are,..)

**Experiment 4:**

Word Analysis

**Experiment 5:**

Word Generation

**Experiment 6:**

Pos tagging

**Experiment 7:**

Morphology

**Experiment 8:**

Chunking

**Experiment 9:**

N-Grams

**Experiment 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

**Reference Books**

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

#### Course Objectives:

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

#### Course outcomes:

1. Show sensitivity to linguistic phenomena and an ability to model them with formal grammars.
2. Able to manipulate probabilities, construct statistical models over strings and trees, and estimate parameters using supervised and unsupervised training methods.
3. Able to design, implement, and analyze NLP algorithms

<b>AY: 2025-26</b> Onwards	<b>J. B. Institute of Engineering and Technology</b> (UGC Autonomous)	<b>M. Tech CSE</b> <b>I Year-I Sem</b>			
<b>Course Code:</b> <b>NME1A</b>	<b>RESEARCH METHODOLOGY &amp; IPR</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>2</b>	<b>0</b>	<b>0</b>	<b>2</b>

**Pre-Requisites:** Basic knowledge of Engineering and Technology concepts

#### **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.

#### **Reference Books**

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.

### **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. Understand research problem formulation.
2. Analyze research related information
3. Follow research ethics
4. Understand that today's world is controlled by Computer, Information Technology, but tomorrow world will be ruled by ideas, concept, and creativity.
5. Understanding that when IPR would take such important place in growth of individuals & nation, it is needless to emphasis the need of information about Intellectual Property Right to be promoted among students in general & engineering in particular.
6. Understand that IPR protection provides an incentive to inventors for further research work and investment in R & D, which leads to creation of new and better products, and in turn brings about, economic growth and social benefits.



<b>AY: 2025-26</b> Onwards	<b>J. B. Institute of Engineering and Technology</b> (UGC Autonomous)	<b>M. Tech CSE</b> <b>I Year-II Sem</b>			
<b>Course Code:</b> <b>NM72A</b>	<b>ADVANCED ALGORITHMS</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>

#### **Pre-Requisites:**

1. Data Structures and Algorithms
2. Basic Design and Analysis of Algorithms
3. Programming fundamentals

#### **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.

#### **Text Books**

1. Kleinberg and Tardos."Algorithm Design

### **Reference Books**

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

### **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. Analyze the complexity/performance of different algorithms.
2. Determine the appropriate data structure for solving a particular set of problems.
3. Categorize the different problems in various classes according to their complexity.

<b>AY: 2025-26</b> Onwards	<b>J. B. Institute of Engineering and Technology</b> (UGC Autonomous)	<b>M. Tech CSE</b> <b>I Year-II Sem</b>			
<b>Course Code:</b> <b>NM72B</b>	<b>ADVANCED COMPUTER ARCHITECTURE</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>

**Pre-Requisites:** Computer Organization

#### **Unit 1:**

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 2:**

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 3:**

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 4:**

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

#### **Unit 5:**

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

#### **Text Books**

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

#### **Reference Books**

1. Computer Architecture, J.L. Hennessy and D.A. Patterson, 4<sup>th</sup> 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.

**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. Computational models and Computer Architectures.
2. Concepts of parallel computer models.
3. Scalable Architectures, Pipelining, Superscalar processors

<b>AY: 2025-26</b> Onwards	<b>J. B. Institute of Engineering and Technology</b> (UGC Autonomous)	<b>M. Tech CSE</b> <b>I Year-II Sem</b>			
<b>Course Code:</b> <b>NM72C</b>	<b>ENTERPRISE CLOUD CONCEPTS(PE-III)</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>

**Pre-Requisites:**

1. Basic knowledge of Computer Networks
2. Fundamentals of Operating Systems
3. Understanding of Virtualization concepts

**Unit 1: 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 2: 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 3: 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 4: 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 5: Transitioning to Cloud-Centric Enterprises**

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 Books**

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

#### **Course Objectives:**

1. 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.

<b>AY: 2025-26</b> Onwards	<b>J. B. Institute of Engineering and Technology</b> (UGC Autonomous)	<b>M. Tech CSE</b> <b>I Year-II Sem</b>			
<b>Course Code:</b> <b>NM72D</b>	<b>CYBER SECURITY</b> (PROFESSIONAL ELECTIVE - I)	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>

#### Pre-Requisites:

1. Basic knowledge of Computer Networks
2. Fundamentals of Operating Systems
3. Understanding of Computer Systems

#### Unit 1:

**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 2:

**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 3

**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 4

**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 5

**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.

#### **Reference Books**

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.

#### **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. Analyze and evaluate the cyber security needs of an organization.
2. Understand Cyber Security Regulations and Roles of International Law
3. Design and develop a security architecture for an organization.
4. Understand fundamental concepts of data privacy attacks



<b>AY: 2025-26</b> Onwards	<b>J. B. Institute of Engineering and Technology</b> (UGC Autonomous)	<b>M. Tech CSE</b> <b>I Year-II Sem</b>			
<b>Course Code:</b> <b>NM72E</b>	<b>PARALLEL COMPUTING</b> <b>(PROFESSIONAL ELECTIVE - III)</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>

**Pre-Requisites:**

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

**Unit 1:**

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

**Unit 2:**

Principles of Parallel Algorithm Design - Analytical Modelling of Parallel Programs

**Unit 3:**

Programming using Message Passing Paradigm (MPI) – Programming Shared Address Space Platforms (P-Threads)

**Unit 4:**

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

**Unit 5:**

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

**Text Books**

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

**Reference Books**

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

**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. Ability to understand the concepts of parallel architectures
2. Ability to select the data structures that efficiently model the information in a problem.
3. Ability to develop an efficient parallel algorithm to solve it.
4. Ability to implement an efficient and correct code to solve it, analyse its performance

<b>AY: 2025-26</b> Onwards	<b>J. B. Institute of Engineering and Technology</b> (UGC Autonomous)	<b>M. Tech CSE</b> <b>I Year-II Sem</b>			
<b>Course Code:</b> <b>NM72F</b>	<b>LARGE LANGUAGE MODELS</b> <b>(PROFESSIONAL ELECTIVE - III)</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>

### Pre-Requisites:

1. Python programming
2. Basics of Machine Learning
3. Fundamentals of Deep Learning

### 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.

### **E-Resources**

1. Hugging Face Documentation – <https://huggingface.co/docs/>

### **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:**

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.

<b>AY: 2025-26</b> Onwards	<b>J. B. Institute of Engineering and Technology</b> (UGC Autonomous)	<b>M. Tech CSE</b> <b>I Year-II Sem</b>			
<b>Course Code:</b> <b>NM72G</b>	<b>BIOINFORMATICS</b> (PROFESSIONAL ELECTIVE - IV)	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>

#### **Pre-Requisites:**

1. Basic knowledge of Biology (molecular biology basics)
2. Fundamentals of Computer Science
3. Understanding of Data Structures

#### **Unit 1: 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 2: 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 3: Databases:**

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

#### **Unit 4: 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 5: 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.

**Course Objectives:**

1. Knowledge On concepts of bioinformatics and biological motivations of sequence analysis.

**Course outcomes:**

1. Understand the Central Dogma & XML (Bio XML) for Bioinformatics
2. Analyze Perl (Bioperl) for Bioinformatics
3. Illustrate Databases technology, architecture and its interfaces
4. Understand Sequence Alignment Algorithms, Phylogenetic Analysis

<b>AY: 2025-26</b> Onwards	<b>J. B. Institute of Engineering and Technology</b> (UGC Autonomous)	<b>M. Tech CSE</b> <b>I Year-II Sem</b>			
<b>Course Code:</b> <b>NM72H</b>	<b>AD-HOC AND SENSOR NETWORKS</b> <b>(PROFESSIONAL ELECTIVE - IV)</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>

**Pre-Requisites:** Computer Networks

#### **Unit 1: 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 2: Data Transmission:**

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

#### **Unit 3: 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 4: 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 5: 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)

#### **Reference Books**

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

**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. Understand the concepts of sensor networks and applications
2. Understand and compare the MAC and routing protocols for adhoc networks
3. Understand the transport protocols of sensor networks

<b>AY: 2025-26</b> Onwards	<b>J. B. Institute of Engineering and Technology</b> (UGC Autonomous)	<b>M. Tech CSE</b> <b>I Year-II Sem</b>			
<b>Course Code:</b> <b>NM72I</b>	<b>ROBOTIC PROCESS AUTOMATION</b> <b>(PROFESSIONAL ELECTIVE - IV)</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>

### Pre-Requisites:

1. Basic knowledge of Programming concepts
2. Fundamentals of Database systems

### Unit 1:

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 2:

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 3:

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 4:

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 5:

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.

### Reference Books

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

### Course Objectives:

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

### Course outcomes:

1. Describe RPA, where it can be applied and how it's implemented.
2. Identify and understand Web Control Room and Client Introduction
3. Understand how to handle various devices and the workload
4. Understand Bot creators, Web recorders and task editors



<b>AY: 2025-26</b> Onwards	<b>J. B. Institute of Engineering and Technology</b> (UGC Autonomous)	<b>M. Tech CSE</b> <b>I Year-II Sem</b>			
<b>Course Code:</b> <b>NM72J</b>	<b>GENERATIVE AI</b> <b>(PROFESSIONAL ELECTIVE - IV)</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>

**Pre-Requisites:**

1. Python programming
2. Basics of Machine Learning
3. Fundamentals of Deep Learning

**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 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 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- Reference**

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.

### **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. Demonstrate knowledge of AI foundations, generative models, and advanced neural architectures.
2. Apply generative AI techniques to create solutions for text, image, video, and multimodal tasks.
3. Design, fine-tune, and optimize Large Language Models for specific applications.
4. Evaluate ethical, social, and legal implications of Generative AI deployments and propose mitigation strategies.

<b>AY: 2025-26</b> Onwards	<b>J. B. Institute of Engineering and Technology</b> (UGC Autonomous)	<b>M. Tech CSE</b> <b>I Year-II Sem</b>			
<b>Course Code:</b> <b>NM721</b>	<b>ADVANCED ALGORITHMS LAB</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>0</b>	<b>0</b>	<b>4</b>	<b>2</b>

**Pre-Requisites:** A course on Computer Programming & Data Structures

#### **Experiment 1:**

Implement assignment problem using Brute Force method

#### **Experiment 2:**

Perform multiplication of long integers using divide and conquer method.

#### **Experiment 3:**

Implement a solution for the knapsack problem using the Greedy method.

#### **Experiment 4:**

Implement Gaussian elimination method.

#### **Experiment 5:**

Implement LU decomposition

#### **Experiment 6:**

Implement Warshall algorithm

#### **Experiment 7:**

Implement the Rabin Karp algorithm.

#### **Experiment 8:**

Implement the KMP algorithm.

#### **Experiment 9:**

Implement Harspool algorithm

#### **Experiment 10:**

Implement max-flow problem.

#### **Text Books**

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

#### **Reference Books**

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.

**Course Objectives:**

1. The student can able to attain knowledge in advanced algorithms.

**Course outcomes:**

1. The student can able to analyze the performance of algorithms.

<b>AY: 2025-26</b> Onwards	<b>J. B. Institute of Engineering and Technology</b> (UGC Autonomous)	<b>M. Tech CSE</b> <b>I Year-II Sem</b>			
<b>Course Code:</b> <b>NM722</b>	<b>ENTERPRISE CLOUD CONCEPTS LAB</b> (LAB -IV)	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>0</b>	<b>0</b>	<b>4</b>	<b>2</b>

#### **Experiment 1:**

Install Virtualbox/VMware Workstation with different flavors of linux or windows OS on top of windows7 or 8

#### **Experiment 2:**

Install a C compiler in the virtual machine created using virtual box and execute Simple Programs

#### **Experiment 3:**

Install Google App Engine. Create a hello world app and other simple web applications using python/java.

#### **Experiment 4:**

Find a procedure to transfer the files from one virtual machine to another virtual machine

#### **Experiment 5:**

Find a procedure to launch virtual machine using trystack (Online Openstack Demo Version)

#### **Experiment 6:**

Install Hadoop single node cluster and run simple applications like word count.

#### **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=2ahUKEwjqrNG0za73AhXZt1YBHZ21DWEQFnoECAMQAQ&url=http%3A%2F%2Fwww.cs.columbia.edu%2F~sedwards%2F\\_classes%2F2015%2F1102-fall%2Flinuxvm.pdf&usg=AOvVaw3xZPuF5xVgk-AQnBRsTtHz](https://www.google.com/urlsa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwjqrNG0za73AhXZt1YBHZ21DWEQFnoECAMQAQ&url=http%3A%2F%2Fwww.cs.columbia.edu%2F~sedwards%2F_classes%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>

#### **Course Objectives:**

1. 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.

<b>AY: 2025-26</b> Onwards	<b>J. B. Institute of Engineering and Technology</b> (UGC Autonomous)	<b>M. Tech CSE</b> <b>I Year-II Sem</b>			
<b>Course Code:</b> <b>NM723</b>	<b>CYBER SECURITY LAB</b> (LAB -IV)	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>0</b>	<b>0</b>	<b>4</b>	<b>2</b>

**Pre-Requisites:** A course on "Network Security and Cryptography".

**Experiment 1:**

Perform an Experiment for port scanning with NMAP.

**Experiment 2:**

Setup a honeypot and monitor the honeypot on the network

**Experiment 3:**

Install Jcrpt /Cryptool tool (or any other equivalent) and demonstrate Asymmetric, Symmetric crypto algorithm, Hash and Digital/PKI signatures.

**Experiment 4:**

Generate minimum 10 passwords of length 12 characters using open SSL command

**Experiment 5:**

Perform practical approach to implement Foot Printing-Gathering target information using Dmitry-Dmagic, UAtester.

**Experiment 6:**

Working with sniffers for monitoring network communication (Wireshark).

**Experiment 7:**

Use Snort to perform real time traffic analysis and packet logging.

**Experiment 8:**

Perform email analysis using Autopsy tool.

**Experiment 9:**

Perform Registry analysis and get boot time logging using process monitor tool

**Experiment 10:**

Perform File type detection using Autopsy tool

**Experiment 11:**

Perform Memory capture and analysis using YAMP tool

**Experiment 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

**Reference Books**

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.

**Course Objectives:**

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

**Course outcomes:**

1. Get the skill to identify cyber threats/attacks.
2. Get the knowledge to solve security issues in day-to-day life.

<b>AY: 2025-26</b> Onwards	<b>J. B. Institute of Engineering and Technology</b> (UGC Autonomous)	<b>M. Tech CSE</b> <b>I Year-II Sem</b>			
<b>Course Code:</b> <b>NM724</b>	<b>PARALLEL COMPUTING LAB</b> <b>(LAB -IV)</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>0</b>	<b>0</b>	<b>4</b>	<b>2</b>

#### **Pre-Requisites:**

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

#### **Experiment 1:**

Design a parallel program to implement Matrix-Vector and Matrix-Matrix Multiplication using MPI library.

#### **Experiment 2:**

Design a parallel program to implement Bubble Sort using OpenMP and Pthread Programming Constructs.

#### **Experiment 3:**

Design a parallel program to implement Quick Sort using OpenMP and Pthread Programming Constructs.

#### **Experiment 4:**

Design a parallel program to implement Bucket Sort using OpenMP and Pthread Programming Constructs.

#### **Experiment 5:**

Design a parallel program to implement Prim's Algorithm using OpenMP and Pthread Programming Constructs.

#### **Experiment 6:**

Design a parallel program to implement DFS Algorithm using OpenMP and Pthread Programming Constructs.

#### **Experiment 7:**

Design a parallel program to implement BFS Algorithm using OpenMP and Pthread Programming Constructs.

#### **Experiment 8:**

Design a parallel program to implement Dijkstra's Algorithm using MPI library

#### **Text Books**

1. Thomas Rauber & Gudula Rünger, Parallel Programming: For Multicore and Cluster Systems, Springer
2. Barry Wilkinson & Michael Allen, Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers, Pearson



3. Ananth Grama, George Karypis, Vipin Kumar, Anshul Gupta, Introduction to Parallel Computing, 2nd Edition, Addison-Wesley
4. Peter Pacheco, An Introduction to Parallel Programming, Morgan Kaufmann

#### **Reference Books**

1. Michael J. Quinn, Parallel Programming in C with MPI and OpenMP, McGraw-Hill
2. P. S. Pacheco, Parallel Programming with MPI, Morgan Kaufmann

#### **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. Ability to understand the concepts of parallel architectures
2. Ability to select the data structures that efficiently model the information in a problem.
3. Ability to develop an efficient parallel algorithm to solve it.
4. Ability to implement an efficient and correct code to solve it, analyze its performance

<b>AY: 2025-26</b> Onwards	<b>J. B. Institute of Engineering and Technology</b> (UGC Autonomous)	<b>M. Tech CSE</b> <b>II Year-I Sem</b>			
<b>Course Code:</b> <b>NM73A</b>	<b>DIGITAL FORENSICS</b> <b>(OPEN ELECTIVE)</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>

**Pre-Requisites:**

1. Cybercrime and Information Warfare, Computer Networks

**Unit 1: 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 2: 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 3: 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 4: 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 5: 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.

**Reference Books**

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.

**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. Understand relevant legislation and codes of ethics.
2. Computer forensics and digital detective and various processes, policies and procedures.
3. E-discovery, guidelines and standards, E-evidence, tools and environment.
4. Email and web forensics and network forensics.

<b>AY: 2025-26</b> Onwards	<b>J. B. Institute of Engineering and Technology</b> (UGC Autonomous)	<b>M. Tech CSE</b> <b>II Year-I Sem</b>			
<b>Course Code:</b> <b>NM73B</b>	<b>ADVANCED OPERATING SYSTEMS</b> <b>(PROFESSIONAL ELECTIVE - V)</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>

### Pre-Requisites:

#### Unit 1:

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 2:

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 Heuristic Algorithm, Raymond's Heuristic Algorithm.

#### Unit 3:

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 4:

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 5:

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 Books

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

#### Reference Books

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

**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

<b>AY: 2025-26</b> Onwards	<b>J. B. Institute of Engineering and Technology</b> (UGC Autonomous)	<b>M. Tech CSE</b> <b>II Year-I Sem</b>			
<b>Course Code:</b> <b>NM73C</b>	<b>QUANTUM COMPUTING</b> <b>(PROFESSIONAL ELECTIVE - V)</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>

### Unit 1: 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 Matrices, Transcendental Numbers

### Unit 2: 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 3: 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 4: 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 5: 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.

### Reference Books

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.

**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

<b>AY: 2025-26</b> Onwards	<b>J. B. Institute of Engineering and Technology</b> (UGC Autonomous)	<b>M. Tech CSE</b> <b>II Year-I Sem</b>			
<b>Course Code:</b> <b>NM73D</b>	<b>PROMPT ENGINEERING</b> <b>(PROFESSIONAL ELECTIVE - V)</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>

#### **Pre-Requisites:**

1. Python programming
2. Basics of Machine Learning and Deep Learning
3. Familiarity with Large Language Models

#### **Unit 1: Fundamentals and Principles of Prompting**

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

#### **Unit 2: 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 3: 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 4: 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 5: 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.

#### **Text Books**

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

#### **Reference Books**

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.



**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.

<b>AY: 2025-26</b> Onwards	<b>J. B. Institute of Engineering and Technology</b> (UGC Autonomous)	<b>M. Tech CSE</b> <b>II Year-I Sem</b>			
<b>Course Code:</b> <b>NME02</b>	<b>INTELLECTUAL PROPERTY RIGHTS</b> <b>(OPEN ELECTIVE)</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>

**Pre-Requisites:**

1. Basic understanding of Engineering and Technology concepts
2. Familiarity with Research and Innovation processes
3. Awareness of Ethical and legal principles

**Unit 1: Introduction to Intellectual property:**

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

**Unit 2: Trade Marks:**

Purpose and function of trademarks, acquisition of trademark rights, protectable matter, selecting, and evaluating trademark, trademark registration processes.

**Unit 3: Law of copyrights:**

Fundamental 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 4: 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 5: New development of intellectual property:**

New developments in trademark law; copyright law, patent law, intellectual property audits. International overview on intellectual property, international – trademark law, copyright law, international patent law, and international development in trade secrets law.

**Text Books**

1. Intellectual property right, Deborah. E. Bouchoux, Cengage learning.

**Reference Books**

1. Intellectual property right – Unleashing the knowledge economy, prabuddha ganguli, Tata McGraw Hill Publishing company ltd.

**Course Objectives:**

1. Introducing the types of intellectual property and their importance
2. Understand IPR laws, patent filing, and protection mechanisms
3. Develop the knowledge of copyrights, trademarks, and industrial designs
4. To explore IPR management, commercialization, and global perspectives

**Course outcomes:**

1. Distinguish and explain various forms of IPRs.
2. Identify criteria to fit one's own intellectual work in particular form of IPRs.
3. Apply statutory provisions to protect particular form of IPRs

<b>AY: 2025-26</b> Onwards	<b>J. B. Institute of Engineering and Technology</b> (UGC Autonomous)	<b>M. Tech CSE</b> <b>II Year-II Sem</b>			
<b>Course Code:</b> <b>NM70A</b>	<b>GENERATIVE AI</b> <b>(OPEN ELECTIVE)</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>

### 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 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 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- Reference**

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.

### **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. Demonstrate knowledge of AI foundations, generative models, and advanced neural architectures.
2. Apply generative AI techniques to create solutions for text, image, video, and multimodal tasks.
3. Design, fine-tune, and optimize Large Language Models for specific applications.
4. Evaluate ethical, social, and legal implications of Generative AI deployments and propose mitigation strategies.

<b>AY: 2025-26</b> Onwards	<b>J. B. Institute of Engineering and Technology</b> (UGC Autonomous)	<b>M. Tech CSE</b> <b>II Year-I Sem</b>			
<b>Course Code:</b> <b>NM70B</b>	<b>INTRUSION DETECTION SYSTEMS</b> <b>(OPEN ELECTIVE)</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>

### Pre-Requisites:

1. Computer Networks, Computer Programming.

### Unit 1:

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 2:

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 3:

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

### Unit 4:

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 5:

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.

### E-Resources

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

**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. Understand fundamental knowledge of intrusion detection and prevention
2. Understand different types of attacks in network layer and code injection human layer
3. Analyze different anomaly detection algorithms

<b>AY: 2025-26</b> Onwards	<b>J. B. Institute of Engineering and Technology</b> (UGC Autonomous)	<b>M. Tech CSE</b> <b>II Year-I Sem</b>			
<b>Course Code:</b> <b>NM73A</b>	<b>DIGITAL FORENSICS</b> <b>(OPEN ELECTIVE)</b>	<b>L</b>	<b>T</b>	<b>P</b>	<b>C</b>
		<b>3</b>	<b>0</b>	<b>0</b>	<b>3</b>

**Pre-Requisites:**

1. Cybercrime and Information Warfare, Computer Networks

**Unit 1: 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 2: 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 3: 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 4: 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 5: 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.

**Reference Books**

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.



**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. Understand relevant legislation and codes of ethics.
2. Computer forensics and digital detective and various processes, policies and procedures.
3. E-discovery, guidelines and standards, E-evidence, tools and environment.
4. Email and web forensics and network forensics.