INFO INSTITUTE OF ENGINEERING
COIMBATORE

(Autonomous, Affiliated to Anna University, Chennai)

DEPARTMENT OF
COMPUTER SCIENCE AND ENGINEERING
M.E. CSE

CURRICULUM AND SYLLABI
R-2026

M.E./M.Tech.Programme (Full-Time)

Applicable to the students admitted from the Academic year 2026– 2027

INFO INSTITUTE OF ENGINEERING, COIMBATORE-641107.

(An Autonomous Institution, Affiliated to Anna University, Chennai)

M.E. - COMPUTER SCIENCE AND ENGINEERING
R-2026, CHOICE BASED CREDIT SYSTEM

Abbreviations:

BS	Basic Science (Mathematics, Physics, Chemistry)	**L**	Laboratory Course
ES	Engineering Science (Programme Core (PC), Programme Elective (PE))	**T**	Theory
SD	Skill Development	**LIT**	Laboratory Integrated Theory
SL	Self Learning	**PW**	Project Work
		TCP	Total Contact Period(s)

DEPARTMENT VISION

- To create a Centre of Excellence in Computer Science and Engineering that nurtures skilled, innovative, and value-driven engineers, capable of responding to the dynamic needs of industry, society, and the research community.

DEPARTMENT MISSION

- To impart a strong academic foundation in Computer Engineering through an outcome-based curriculum, industry-aligned coursework, and extensive hands-on training.
- To foster teamwork, effective communication, ethical conduct, and social responsibility among students.
- To cultivate a spirit of lifelong learning with modern computing tools for problem solving, entrepreneurship, and adaptability to enable students to grow into successful professionals.

PROGRAM OUTCOMES (POs)

PO1 - An ability to independently carry out research /investigation and development work to solve practical problems

PO2 - An ability to write and present a substantial technical report/document.

PO3 - Students should be able to demonstrate a degree of mastery over the area as per the specialization of the program. The mastery should be at a level higher than the requirements in the appropriate bachelor program

INFO INSTITUTE OF ENGINEERING • COIMBATORE-641107 •

PROGRAM SPECIFIC OUTCOMES (PSOs)

PSO1: Advanced Computing and Problem Solving: Analyze, design, and implement advanced algorithms, architectures, and computational models to develop sustainable and scalable solutions, aligning with industry and societal needs to solve complex problems in diverse domains.

PSO2: Research and Innovation Competence: Undertake independent research and apply advanced tools and methodologies to propose innovative solutions for real-world and interdisciplinary computing challenges, demonstrating research aptitude.

Curriculum for I and II Semester

Semester I							
S.No.	**Course Code**	**Course Title**	**Type**	**L-T-P**	**TCP**	**Credits**	**Category**
1	26PC101T	Advanced Mathematical Methods	T	3-1-0	4	4	BS
2	26PC102T	Advanced Database Technologies	T	3-0-0	3	3	ES (PC)
3	26PC103T	Advanced Operating Systems	T	3-0-0	3	3	ES (PC)
4	26PC104T	Advanced Compiler Design	T	3-0-0	3	3	ES (PC)
5	26PC105I	Advanced Data Structures and Algorithms	LIT	3-0-4	7	5	ES (PC)
6	26PC106S	Technical Seminar	-	0-0-2	2	1	SD
				Total Credits	**22**	**19**	

Semester II							
S.No.	**Course Code**	**Course Title**	**Type**	**L-T-P**	**TCP**	**Credits**	**Category**
1	26PC201T	Artificial Intelligence and Machine Learning	T	3	3	3	ES (PC)
2	26PC202T	Cloud and Big Data Analytics	T	3	3	3	ES (PC)
3	26PC203T	Quantum Computing	T	2	2	2	ES (PC)
4		Programme Elective I	T	3	3	3	ES (PE)
5	26PC204I	Multicore Architectures	LIT	3	5	4	ES (PC)
6	26PC2N1T	Industry Oriented Course I	T	1	1	1	SD



Semester II							
S.No.	**Course Code**	**Course Title**	**Type**	**L-T-P**	**TCP**	**Credits**	**Category**
7	26PC205N	Industrial Training	-	-	-	2	SD
8	-	Self-Learning Course	-	-	-	-	SL
				Total Credits	**17**	**18**	

Semester III							
S.No.	**Course Code**	**Course Title**	**Type**	**L-T-P**	**TCP**	**Credits**	**Category**
1		Programme Elective II	T	3-0-0	3	3	ES (PE)
2		Programme Elective III	T	3-0-0	3	3	ES (PE)
3		Programme Elective IV	T	3-0-0	3	3	ES (PE)
4		Programme Elective V	T	3-0-0	3	3	ES (PE)
5	26PC3N2T	Industry Oriented Course II	T	1-0-0	1	1	SD
6	26PC301P	Project Work I	PW	0-0-12	12	6	SD
				Total Credits	**25**	**19**	

Semester IV							
S.No.	**Course Code**	**Course Title**	**Type**	**L-T-P**	**TCP**	**Credits**	**Category**
1	26PC401P	Project Work II	PW	0-0-24	24	12	SD
Total Credits					**24**	**12**	



PROGRAMME ELECTIVE COURSES (PE)

Vertical -1 (Software Engineering & Secure Computing)

S.No.	Course Code	Course Title	Type	L-T-P	TCP	Credits	Category
1.	26PC$A1T	Advanced Software Testing and Quality Assurance	T	3-0-0	3	3	ES (PE)
2.	26PC$A2T	Agile Methodologies	T	3-0-0	3	3	ES (PE)
3.	26PC$A3T	Quantum Cryptography	T	3-0-0	3	3	ES (PE)
4.	26PC$A4T	Web 3.0	T	3-0-0	3	3	ES (PE)
5.	26PC$A5T	Blockchains Architecture and Design	T	3-0-0	3	3	ES (PE)
6.	26PC$A6T	Secure Software Engineering	T	3-0-0	3	3	ES (PE)

Vertical -2 (Artificial Intelligence & Intelligent Systems)

S.No.	Course Code	Course Title	Type	L-T-P	TCP	Credits	Category
1.	26PC$B1T	Advanced Deep Learning and Neural Networks	T	3-0-0	3	3	ES (PE)
2.	26PC$B2T	Text and Speech Processing	T	3-0-0	3	3	ES (PE)
3.	26PC$B3T	Advanced Large Language Models	T	3-0-0	3	3	ES (PE)
4.	26PC$B4T	Agentic AI	T	3-0-0	3	3	ES (PE)
5.	26PC$B5T	Cognitive Computing	T	3-0-0	3	3	ES (PE)
6.	26PC$B6T	Human-Centered AI	T	3-0-0	3	3	ES (PE)
7.	26PC$B7T	Federated Learning	T	3-0-0	3	3	ES (PE)
8.	26PC$B8T	Deep Learning for Computer Vision	T	3-0-0	3	3	ES (PE)



Vertical -3 (IoT, Edge & Emerging Computing)

S.No.	Course Code	Course Title	Type	L-T-P	TCP	Credits	Category
1.	26PC$C1T	Web of Things	T	3-0-0	3	3	ES (PE)
2.	26PC$C2T	AI in IoT	T	3-0-0	3	3	ES (PE)
3.	26PC$C3T	Edge and Fog Computing	T	3-0-0	3	3	ES (PE)
4.	26PC$C4T	Green Computing and Sustainability	T	3-0-0	3	3	ES (PE)
5.	26PC$C5T	Mixed Reality	T	3-0-0	3	3	ES (PE)
6.	26PC$C6T	Quantum Machine Learning	T	3-0-0	3	3	ES (PE)
7.	26PC$C7T	Web 3.0	T	3-0-0	3	3	ES (PE)
8.	26PC$C8T	Vibe Coding	T	3-0-0	3	3	ES (PE)

CREDITS SUMMARY

Name of the Programme: M.E. Computer Science and Engineering						
CATEGORY	Credits per Semester				Total Credits	%
	I	II	III	IV		
BS	4	-	-	-	4	6
ES(PC)	14	12	-	-	26	38
ES(PE)	-	3	12	-	12	18
SD	1	3	7	12	23	34
SL	-	✓	-	-	-	-
TOTAL	19	18	19	12	68	100

INFO INSTITUTE OF ENGINEERING • COIMBATORE - 641 107 •

SEMESTER-I



26PC101T	ADVANCED MATHEMATICAL METHODS	L	T	P	C
		3	1	0	4

Course Objectives:

- Develop an in-depth understanding of advanced concepts in linear algebra, multivariate analysis, and number theory for computer science applications.
- Apply mathematical tools such as eigenvalue decomposition, SVD, and multivariate statistical methods to real-world computing and data-driven problems.
- Analyze and implement number-theoretic techniques for cryptography, security, and algorithmic problem-solving in computer science.

Unit I	LINEAR ALGEBRA	15 Hours

Vector spaces - Norms - Inner Products - Eigenvalues using QR transformations - QR factorization - Generalized eigenvectors - Canonical forms - Singular value decomposition and applications - Pseudo inverse - Least square approximations.
Activities: Compute QR factorization, Singular value decomposition, and Least-square approximations using open-source software.

Unit II	MULTIVARIATE ANALYSIS	15 Hours

Random vectors and matrices - Mean vectors and covariance matrices - Multivariate normal density and its properties - Principal components - Population principal components - Principal components from standardized variables.
Activities: Use open-source software to compute mean vectors, covariance matrices, and perform principal component analysis for multivariate data analysis.

Unit III	ELEMENTARY NUMBER THEORY	15 Hours

The division algorithm - Divisibility and the Euclidean algorithm - The fundamental theorem of arithmetic - Modular arithmetic and basic properties of congruences - Principles of mathematical induction and well ordering principle - Primality Testing algorithms - Chinese Remainder Theorem - Quadratic Congruence.
Activities: Analyze number theory problems involving divisibility, congruences, primality testing, and the Chinese Remainder Theorem using open-source software.

Unit IV	ADVANCED NUMBER THEORY	15 Hours

Advanced Number Theory - Primality Testing algorithms - Chinese Remainder Theorem - Quadratic Congruence - Discrete Logarithm - Factorization Methods - Side Channel Attacks - Shannon Theory - Perfect Secrecy - Semantic Security.
Activities: Analyze cryptographic algorithms and security concepts, including primality testing, discrete logarithms, factorization methods, and information-theoretic security using open-source software.

Total Hours	60

Weightage: Continuous Assessment: 40%, End Semester Examinations: 60%

Continuous Assessment: Assignments (15%), Quiz (10%), Virtual Demo (20%), Flipped Class Room (10%), Review of GATE and IES Questions (25%), Project (20%)

Course outcomes:

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

CO1: Apply concepts of vector spaces, eigenvalues and singular value decomposition to solve computing problems. CO2: Analyze random vectors and covariance structures using principal component analysis for data-driven applications.

CO3: Apply elementary number-theoretic techniques such as the Euclidean algorithm, modular arithmetic and primality testing to algorithmic problems.

CO4: Analyze advanced number-theoretic and cryptographic techniques including discrete logarithm and factorization methods for security applications.

Reference:

1. Gilbert Strang, Linear Algebra and Its Applications, Cengage Learning.
2. Richard A. Johnson & Dean W. Wichern, Applied Multivariate Statistical Analysis, Pearson.
3. Neal Koblitz, A Course in Number Theory and Cryptography, Springer.
4. Victor Shoup, A Computational Introduction to Number Theory and Algebra, Cambridge University Press.

E-resources:

1. Linear algebra: https://ocw.mit.edu/courses/18-06-linear-algebra-spring-2010/video_galleries/video-lectures/
2. Linear algebra: https://home.iitm.ac.in/mtnair/linear.pdf
3. Probability and Statistics: https://nptel.ac.in/courses/111105041
4. Number Theory: https://crypto.stanford.edu/pbc/notes/numbertheory/

	CO-PO Mapping			CO-PSO Mapping	
PO & PSO / CO	**PO1**	**PO2**	**PO3**	**PSO1**	**PSO2**
CO1	3	-	-	2	-
CO2	3	-	-	2	-
CO3	2	-	2	-	2
CO4	2	-	2	-	3
AVERAGE	3	-	2	2	3

1-Low, 2-Medium, 3-High

26PC102T	**ADVANCED DATABASE TECHNOLOGIES**	**L**	**T**	**P**	**C**
		3	0	0	3

Course Objectives:

- To strengthen the understanding of enhanced ER models and their transformation into relational models with indexing and file structures.
- To understand object-oriented and object-relational database concepts and querying using OQL.
- To explore techniques in query processing, execution, and optimization strategies.

Unit I	**ENTITY RELATIONSHIP MODEL**	**9 Hours**

Entity Relationship Model Revised-Subclasses, Superclasses and Inheritance -Specialization and Generalization-Union Types-Aggregation.

Activity: Design ER Model for a specific use case.

Unit II	**ENHANCED ENTITY RELATIONAL MODEL**	**9 Hours**

Relational Model Revised, Converting ER and EER Model to Relational Model-SQL and Advanced Features, File Structures, Hashing, and Indexing.

Activity: Demonstration of SQL Implementation.

Unit III	**OBJECT RELATIONAL DATABASES**	**9 Hours**

Object Database Concepts-Object Database Extensions to SQL, The ODMG Object Model and ODL, Object Database Conceptual Design-Object Query Language OQL-Language Binding in the ODMG Standard.

Activity: Demonstration of Object Query Language.

Unit IV	**QUERY PROCESSING AND OPTIMIZATION & DISTRIBUTED DATABASES**	**9 Hours**

Query Processing, Query Trees and Heuristics, Query Execution Plans, Cost Based Optimization. Distributed Database Concepts, Data Fragmentation, Replication and Allocation, Distributed Database Design Techniques, Distributed Database Architectures.

Activity: Design of Query Evaluation Plans; Demonstration of Concurrency and Transactions.

Unit V	**ADVANCED DATABASE MODELS, SYSTEMS AND APPLICATIONS**	**9 Hours**

Introduction to NOSQL Systems-The CAP Theorem, Document based NOSQL Systems, Key-value Stores, Column-Based or Wide Column NOSQL Systems, NOSQL Graph Databases and Neo4j. Active Database Concepts and Triggers, Temporal Database Concepts, Spatial Database Concepts, Multimedia Database Concepts, Deductive Database Concepts, Introduction to Information Retrieval and Web Search.

Activity: Design application with MongoDB; Demonstration of Hadoop infrastructure for Big Data Analytics.

Total Hours	**45**

Weightage: Continuous Assessment: 40%, End Semester Examinations: 60%

Continuous Assessment: Assignments (15%), Quiz (10%), Virtual Demo (20%), Flipped Class Room (10%), Review of GATE and IES Questions (25%), Project (20%)

Course outcomes:

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

CO1: Elaborate different database models for effective database design. CO2: Implement advanced database features for optimized data retrieval.

CO3: Evaluate query processing and optimization strategies to improve system performance.

CO4: Design solutions using advanced database models to address complex data-intensive applications.

Reference:

1. Elmasri, R., & Navathe, S. B. (2016). Fundamentals of database systems. Pearson Education.
2. Silberschatz, A., Korth, H. F., & Sudarshan, S. (2020). Database system concepts, McGraw Hill Education.
3. Ceri, S., & Pelagatti, G. Distributed databases: Principles and systems. McGraw Hill.
4. Ramakrishnan, R., & Gehrke, J. (2004). Database management systems. McGraw Hill.

E-resources:

1. https://www.edx.org/learn/sql/stanford-university-databases-advanced-topics-in-sql
2. https://www.coursera.org/courses?query=sql&productDifficultyLevel=Advanced

	CO-PO Mapping			**CO-PSO Mapping**	
PO & PSO / CO	**PO1**	**PO2**	**PO3**	**PSO1**	**PSO2**
CO1	-	-	-	-	-
CO2	3	-	-	3	-
CO3	-	-	2	-	2
CO4	-	1	-	3	-
AVERAGE	3	1	2	3	2

1-Low, 2-Medium, 3-High

26PC103T	ADVANCED OPERATING SYSTEMS	L	T	P	C
		3	0	0	3

Course Objectives:

- To analyze the architectures and design issues of advanced operating systems.
- To develop the model for process synchronization and recovery in complex environments.
- To evaluate algorithms for distributed coordination, resource management, fault tolerance, and security.

Unit I	ADVANCED PROCESS AND THREAD MANAGEMENT	9 Hours

Multithreading models, thread pools, context switching, Synchronization issues and solutions: semaphores, monitors, lock-free data structures, CPU scheduling in multi-core systems.

Activity: CPU scheduler simulation for multicore systems.

Unit II	MEMORY AND RESOURCE MANAGEMENT IN MODERN OS	9 Hours

Virtual memory, demand paging, page replacement policies-Huge pages, NUMA-aware memory management-Resource allocation in cloud-native environments.

Activity: Simulate demand paging and page replacement algorithms.

Unit III	VIRTUALIZATION AND CONTAINERIZATION	9 Hours

Hypervisors (Type I & II), KVM, QEMU, Xen-Containers: Docker, LXC, systemd-nspawn-OS-level virtualization and namespaces.

Activity: Deploy and configure Docker containers with various images.

Unit IV	DISTRIBUTED OPERATING SYSTEMS, FILE SYSTEMS AND SECURITY	9 Hours

Distributed scheduling, communication, and synchronization-Distributed file systems: NFS, GFS, HDFS-Transparency issues and fault tolerance. Access control models: DAC, MAC, RBAC-OS hardening techniques, sandboxing, SELinux, AppArmor-Secure boot, rootkit detection, trusted execution environments.

Activity: Simulate distributed process synchronization; Implement RBAC using Linux user and group permissions.

Unit V	REAL-TIME, EMBEDDED, EDGE AND CLOUD OS	9 Hours

Real-time scheduling algorithms (EDF, RM)-POSIX RT extensions, RTOS architecture-TinyOS, FreeRTOS case studies. Serverless OS, unikernels, lightweight OS for edge computing-Mobile OS internals (Android, iOS)-OS for quantum and neuromorphic computing (intro).

Activity: Analyze FreeRTOS task scheduling behavior; Analyze Android's system architecture using emulator tools.

Total Hours	45

Weightage: Continuous Assessment: 40%, End Semester Examinations: 60%

Continuous Assessment: Assignments (15%), Quiz (10%), Virtual Demo (20%), Flipped Class Room (10%), Review of GATE and IES Questions (25%), Project (20%)

Course outcomes:

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

CO1: Describe operating system concepts for memory and resource management.

CO2: Analyse virtualization and distributed OS mechanisms for scalability and performance. CO3: Evaluate OS security and resource handling strategies in diverse environments.

CO4: Design innovative OS solutions using modern tools and techniques.

Reference:

1. Tanenbaum, A. S., & Bos, H. (2023). Modern operating systems. Pearson.
2. Buyya, R., et al. (2022). Content delivery networks and emerging operating systems. Springer.
3. Silberschatz, A., Galvin, P. B., & Gagne, G. (2022). Operating system concepts. Wiley.
4. Anderson, T., & Dahlin, M. (2021). Operating systems: Principles and practice. Recursive Books.
5. Arpaci-Dusseau, R. H., & Arpaci-Dusseau, A. C. (2020). Operating systems: Three easy pieces.

E-resources:

1. Prof. Smruti Ranjan Sarangi, "Advanced Distributed Systems", IIT Delhi, NPTEL, https://onlinecourses.nptel.ac.in/noc22_cs80/preview
2. Prof. Rajiv Misra, "Cloud Computing and Distributed Systems", IIT Patna, NPTEL, https://nptel.ac.in/courses/106104182

	CO-PO Mapping			CO-PSO Mapping	
PO & PSO / CO	**PO1**	**PO2**	**PO3**	**PSO1**	**PSO2**
CO1	-	-	-	-	-
CO2	3	-	-	3	-
CO3	-	-	2	-	2
CO4	-	1	-	3	-
AVERAGE	3	1	2	3	2

1-Low, 2-Medium, 3-High

26PC104T	**ADVANCED COMPILER DESIGN**	**L**	**T**	**P**	**C**
		3	0	0	3

Course Objectives:

- To analyze the theory and principles of modern compiler design and advanced optimization techniques.
- To design and implement efficient front-end and back-end compiler components for programming languages.
- To evaluate code optimization strategies and runtime environment management in contemporary architectures.

Unit I	**INTERMEDIATE REPRESENTATIONS AND CONTROL FLOW ANALYSIS**	**9 Hours**

Static single assignment (SSA) form- Context-Free Grammer (CFG) construction-dominance relations-Intermediate Representation (IR) design for functional and imperative languages-Static single assignment and def-use chains.

Activity: Convert source code to SSA form using LLVM IR; Visualize control flow graphs from SSA using LLVM tools.

Unit II	**PROGRAM ANALYSIS AND TRANSFORMATIONS**	**9 Hours**

Data flow analysis- live variable analysis-reaching definitions-Alias analysis and dependence analysis-Loop optimizations and transformations.

Activity: Perform loop unrolling and strength reduction; Conduct live variable analysis and visualize data flow graphs.

Unit III	**ADVANCED OPTIMIZATIONS, POLYHEDRAL COMPILATION AND JIT COMPILATION**	**9 Hours**

Polyhedral model for loop nests-Tiling, skewing, fusion, and vectorization-Profile-guided and feedback-directed optimizations. JIT compilation models: tracing, method-based-GraalVM architecture, Java HotSpot internals-LLVM JIT and dynamic language support.

Activity: Implement loop tiling and skewing on a matrix multiplication program; Develop a basic JIT-enabled interpreter with LLVM or GraalVM.

Unit IV	MACHINE LEARNING AND DOMAIN-SPECIFIC LANGUAGES IN COMPILER DESIGN	9 Hours

ML for phase ordering, auto-tuning, and IR prediction-Reinforcement learning for optimization passes-Dataset creation and benchmarking for compiler ML. Designing DSLs for AI/ML, DSP, graphics-Code generation for custom accelerators-Integration with TensorFlow XLA and Halide.

Activity: Train an ML model to predict optimization passes; Design and test a simple DSL grammar using ANTLR.

Unit V	SECURITY, VERIFICATION AND FUTURE TRENDS	9 Hours

Secure compilation and type-safe intermediate representations-Compiler fuzzing and formal verification (e.g., CompCert)-Quantum compilers, multi-target compilers, and neuromorphic systems.

Activity: Use CompCert to verify compilation of simple programs; Apply compiler fuzzing using tools like libFuzzer.

Total Hours	**45**

Weightage: Continuous Assessment: 40%, End Semester Examinations: 60%

Continuous Assessment: Assignments (15%), Quiz (10%), Virtual Demo (20%), Flipped Class Room (10%), Review of GATE and IES Questions (25%), Project (20%)

Course outcomes:

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

CO1: Explain intermediate control flow techniques in compiler design.

CO2: Apply program analysis techniques and advanced optimizations for design of compilers. CO3: Develop compiler features and machine learning techniques for optimization.

CO4: Evaluate secure compilation strategies for quantum and multi-target compilation.

Reference:

1. Cooper, K. D., & Torczon, L. (2023). Engineering a compiler. Morgan Kaufmann.
2. Grune, D., Bal, H. E., Jacobs, C. J. H., & Langendoen, K. G. (2012). Modern compiler design (2nd ed.). Springer.
3. Aho, A. V., Lam, M. S., Sethi, R., & Ullman, J. D. (2006). Compilers: Principles, techniques, and tools (2nd ed.). Pearson.
4. Volter, M. (2013). DSL engineering: Designing, implementing and using domain-specific languages. dslbook.org.
5. Sarda, S., & Pandey, M. (2015). LLVM essentials. Packt Publishing.

E-resources:

1. Prof. Amey Karkare, IIT Kanpur, "Advanced Compiler Optimizations" - https://www.cse.iitk.ac.in/users/karkare/Courses/cs738/
2. Prof. Santanu Chattopadhyay, "Compiler Design", IIT Kharagpur - https://onlinecourses.nptel.ac.in/noc21_cs07/preview

	CO-PO Mapping			CO-PSO Mapping	
PO & PSO / CO	**PO1**	**PO2**	**PO3**	**PSO1**	**PSO2**
CO1	-	-	-	-	-
CO2	3	-	-	3	-
CO3	-	-	2	-	2
CO4	-	1	-	3	-
AVERAGE	3	1	2	3	2

1-Low, 2-Medium, 3-High

Approved by 1st Academic Council Meeting held on 25th August 2026



26PC105I	ADVANCED DATA STRUCTURES AND ALGORITHMS	L	T	P	C
		3	0	4	5

Course Objectives:

- To explore advanced linear, tree, and graph data structures and their applications.
- To design efficient algorithms using appropriate algorithmic paradigms.
- To evaluate computational complexity and identify tractable vs. intractable problems.

Unit I	LINEAR DATA STRUCTURES AND MEMORY OPTIMIZATION	8 Hours

Advanced arrays: Sparse arrays, dynamic arrays, cache-aware structures, Linked lists: Skip lists, unrolled linked lists, XOR linked lists, Stacks and Queues: Priority queues, double-ended queues, circular buffers, Hashing: Perfect hashing, cuckoo hashing, extendible hashing.

Activity: Implement skip lists and measure performance compared with balanced BST; Experiment with cache-aware data structures and analyze memory utilization.

Unit II	ADVANCED TREE DATA STRUCTURES	8 Hours

Balanced Trees: AVL, Red-Black Trees, Splay Trees, Treaps, Multi-way Trees: B-Trees, B+ Trees, R-Trees, Segment Trees, Fenwick Trees, Suffix Trees and Tries for string processing, Applications in indexing, text retrieval, computational geometry.

Activity: Implement B+ tree for database indexing use-case; Design a suffix tree-based algorithm for DNA sequence matching.

Unit III	GRAPH DATA STRUCTURES AND ALGORITHMS	8 Hours

Representation: Adjacency list/matrix, incidence matrix, compressed storage, Traversals: DFS, BFS with applications, Shortest Path Algorithms: Dijkstra, Bellman-Ford, Floyd-Warshall, Johnson's algorithm, Minimum Spanning Trees: Prim's, Kruskal's, Borůvka's algorithm, Network Flow Algorithms: Ford-Fulkerson, Edmonds-Karp, Push-Relabel.

Activity: Implement Johnson's algorithm for sparse graph shortest paths; Demonstration of Maximum flow in traffic or network routing simulation.

Unit IV	ALGORITHM DESIGN AND PARADIGMS	7 Hours

Divide and Conquer: Karatsuba's multiplication, Strassen's algorithm, Greedy Methods: Huffman coding, interval scheduling, set cover approximation, Dynamic Programming: Matrix chain multiplication, Floyd-Warshall, knapsack variants, Backtracking and Branch-and-Bound, Randomized Algorithms and Probabilistic Analysis.

Activity: Implement Strassen's algorithm and compare with naive matrix multiplication; Develop a randomized algorithm for primality testing (Miller-Rabin).

Unit V	COMPUTATIONAL COMPLEXITY AND ADVANCED TOPICS	8 Hours

Complexity Classes: P, NP, NP-Complete, NP-Hard, Reductions: Polynomial-time reductions, Cook-Levin theorem (overview), Approximation Algorithms: Vertex cover, set cover, TSP, k-center problem, Heuristic Algorithms: Local search, simulated annealing, genetic algorithms. Randomized Algorithms - Monte Carlo Algorithms, Parallel and Distributed Algorithms - PRAM Model, Streaming Algorithms, Advanced String Matching - Suffix Trees, Suffix Arrays.

Activity: Implement approximation algorithm for vertex cover; Design of parallel and distributed algorithms.

Total Hours	**45**
Practical Hours	**60**

Weightage: Continuous Assessment: 50%, End Semester Examinations: 50%

Continuous Assessment: Assignments (15%), Quiz (10%), Virtual Demo (20%), Flipped Class Room (10%), Review of GATE and IES Questions (25%), Project (20%)

Course outcomes:

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

CO1: Describe data structures and implement algorithmic solutions for complex computational problems. CO2: Analyze the time complexity and efficiency of algorithms for various computing problems.

CO3: Evaluate algorithmic techniques and data structures to determine their suitability for different applications. CO4: Design optimized solutions for real-world problems using appropriate algorithms and data structures.

Reference:

1. Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). Introduction to algorithms. MIT Press.
2. La Rocca, M. (2021). Advanced algorithms and data structures. Manning Publications.
3. Goodrich, M. T., Tamassia, R., & Mount, D. M. (2011). Data structures and algorithms in C++. John Wiley & Sons, Inc.
4. Weiss, M. A. (2014). Data structures and algorithm analysis in C++. Pearson Education.
5. Drozdek, A. (2013). Data structures and algorithms in C++. Cengage Publications.

E-resources:

1. https://www.theiotacademy.co/blog/data-structures-and-algorithms-in-c/
2. https://github.com/afrid18/Data_structures_and_algorithms_in_cpp

	CO-PO Mapping			CO-PSO Mapping	
PO & PSO / CO	**PO1**	**PO2**	**PO3**	**PSO1**	**PSO2**
CO1	-	-	-	-	-
CO2	3	-	-	3	-
CO3	-	-	2	-	2
CO4	-	1	-	3	-
AVERAGE	3	1	2	3	2

1-Low, 2-Medium, 3-High

26PC106S	TECHNICAL SEMINAR	L	T	P	C
		0	0	2	1

Course Objectives:

- To enable students to gain confidence in collecting relevant technical information on a chosen topic, organizing it and presenting it before an audience.
- To develop presentation skills and technical report writing ability.

Weightage: Continuous Assessment: 100% (Based on relevance of the topic, presentation skill, quality of the report and viva-voce)

Course outcomes:

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

CO1: Survey recent literature on an advanced topic in Computer Science and Engineering. CO2: Present technical content effectively using appropriate presentation aids.

CO3: Prepare a well-organized technical seminar report.

	CO-PO Mapping			CO-PSO Mapping	
PO & PSO / CO	**PO1**	**PO2**	**PO3**	**PSO1**	**PSO2**
CO1	3	-	-	-	-
CO2	-	3	-	-	-
CO3	-	3	-	-	-
AVERAGE	3	3	-	-	-

1-Low, 2-Medium, 3-High



SEMESTER-II



26PC201T	ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING	L	T	P	C
		3	0	0	3

Course Objectives:

- To understand the fundamentals of Artificial Intelligence and its core problem-solving techniques using intelligent agents, search strategies, and logic-based reasoning.
- To comprehend the theoretical foundations and algorithmic frameworks of machine learning, with an emphasis on supervised and unsupervised learning methodologies.

Unit I	PROBLEM SOLVING AND KNOWLEDGE REPRESENTATION	9 Hours

Solving problems by searching - adversarial search - constraint satisfaction problems - Logical agents - Propositional logic - First order logic - Forward chaining - Backward chaining - Ontological Representations and Reasoning Systems.

Activity: Develop a simple Intelligent Agent that navigates a grid world using search algorithms.

Unit II	UNCERTAIN KNOWLEDGE AND REASONING	9 Hours

Overview of uncertainty and basic probability - Baye's rule - Bayesian networks - Hidden Markov models - Kalman filters - Utility Functions - Decision Networks - Sequential decision problems - Game theory.

Activity: Develop a simple Intelligent Agent that makes decisions in adversarial conditions.

Unit III	MACHINE LEARNING INTRODUCTION	9 Hours

Types of learning - Hypothesis space - Inductive bias - Evaluation - Cross validation - Bias variance - Bias Variance Trade off.

Activity: Develop a simple Intelligent Agent that uses logical reasoning to infer hidden facts or constraints.

Unit IV	SUPERVISED LEARNING	9 Hours

Linear Regression - Logistic Regression - Decision trees: Classification and regression trees - Neural networks, multilayer perceptron - Support vector machines: linear and non-linear kernel functions - K nearest neighbours - Ensemble learning: bagging and boosting.

Activity: Behaviour Modelling, Disease Diagnosis and other similar applications suitable for supervised learning.

Unit V	UNSUPERVISED AND PROBABILISTIC LEARNING	9 Hours

Clustering: partition, hierarchical and density based clustering - Self organizing maps - Expectation maximization - Gaussian mixture models - Principal component analysis - Bayesian learning: Bayes optimal classifier, Naive Bayes classifier, Bayesian belief networks.

Activity: Customer Segmentation, Image Processing and other similar applications suitable for unsupervised and probabilistic learning.

Total Hours	45

Weightage: Continuous Assessment: 40%, End Semester Theory Examination: 60%

Continuous Assessment: Assignments (30%), Quiz (10%), Virtual Demonstration (25%), Flipped Classroom (10%), Review of GATE & IES Questions (25%)

Course outcomes:

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

CO1: Describe the fundamental concepts, scope, and applications of artificial intelligence and machine learning in solving real-world problems.

CO2: Analyze various artificial intelligence and machine learning models to understand their behavior, performance, and suitability for different problem domains.

CO3: Evaluate machine learning algorithms and artificial intelligence techniques based on accuracy, efficiency, and applicability to specific use cases.

CO4: Design intelligent systems by selecting and integrating appropriate machine learning models and AI techniques to address complex real-world applications.

Reference:

1. Stuart Russell and Peter Norvig, Artificial Intelligence - A Modern Approach, Prentice Hall, 3rd edition, 2011.
2. Elaine Rich, Kevin Knight and Shiv Shankar B. Nair, Artificial Intelligence, 3rd edition, Tata McGraw Hill, 2009.
3. Tom Mitchell, "Machine Learning", McGraw Hill, 3rd Edition, 1997.
4. Ethem Alpaydin, "Introduction to Machine Learning", MIT Press, Prentice Hall of India, Third Edition 2014.
5. Mehryar Mohri, Afshin Rostamizadeh, Ameet Talwalkar "Foundations of Machine Learning", MIT Press, 2012.

E-resources:

1. NPTEL Course, An Introduction to Artificial Intelligence by Prof Mausam
2. NPTEL Course, Machine Learning, by Prof. Carl Gustaf Jansson

	CO-PO Mapping			CO-PSO Mapping	
PO & PSO / CO	**PO1**	**PO2**	**PO3**	**PSO1**	**PSO2**
CO1	-	-	-	-	-
CO2	3	-	-	3	-
CO3	-	-	2	-	2
CO4	-	1	-	3	-
AVERAGE	3	1	2	3	2

1-Low, 2-Medium, 3-High

26PC202T	**CLOUD AND BIG DATA ANALYTICS**	**L**	**T**	**P**	**C**
		3	0	0	3

Course Objectives:

- To understand cloud computing paradigms and service models.
- To study big data characteristics, technologies and storage.
- To explore Hadoop ecosystem and processing tools.
- To analyze, store, and visualize large-scale data.
- To develop and deploy cloud-based big data applications.

Unit I	**CLOUD COMPUTING**	**9 Hours**

Introduction to Cloud Computing - Service Models: IaaS, PaaS, SaaS - Deployment Models: Public, Private, Hybrid, Community - Virtualization: Hypervisors, VM provisioning - Cloud providers: AWS, Azure, Google Cloud overview.

Activity: Create a free-tier AWS account and explore EC2, S3.

Unit II	**BIG DATA ECOSYSTEM & HADOOP**	**9 Hours**

Big Data characteristics (Volume, Velocity, Variety, Veracity, Value) - Challenges in Big Data - Hadoop Architecture and HDFS - MapReduce Programming Model - YARN.

Activity: Watch Hadoop tutorial and submit a summary report.

Unit III	**DATA STORAGE & PROCESSING TECHNIQUES**	**9 Hours**

NoSQL Databases: MongoDB, Cassandra, HBase - Data Ingestion: Sqoop, Flume - Hive, Pig - Architecture and Querying - Data Cleaning and Preparation.

Activity: HiveQL assignments on datasets from Kaggle.

INSTITUTE OF ENGINEERING & TECHNOLOGY • COIMBATORE - 641107

Unit IV	ADVANCED BIG DATA ANALYTICS	9 Hours
Introduction to Apache Spark - Spark vs. MapReduce - Spark SQL, Data Frames, RDDs - Machine Learning using Spark MLlib - Real-time processing: Kafka and Spark Streaming. **Activity:** Mini-project using PySpark on COVID or Traffic dataset.		
Unit V	**CLOUD INTEGRATION AND SECURITY**	**9 Hours**
Data on Cloud - Storage and Compute Services - Data Security and Privacy in Cloud - Cloud-based Data Analytics Use Cases - Cloud-Native vs Traditional Applications - Case Study: Smart Cities / Healthcare Analytics. **Activity:** Create a cloud deployment report using Azure/AWS.		
	Total Hours	**45**
Weightage: Continuous Assessment: 40%, End Semester Theory Examination: 60%		
Continuous Assessment: Assignments (30%), Quiz (10%), Virtual Demonstration (25%), Flipped Classroom (10%), Review of GATE & IES Questions (25%)		

Course outcomes:

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

CO1: Describe the fundamental concepts, architectures, and service models of cloud computing and big data analytics used in modern data-driven environments.

CO2: Analyze cloud-based big data processing frameworks and analytics techniques to understand their performance, scalability, and resource utilization.

CO3: Evaluate big data analytics tools, cloud platforms, and data processing models for their effectiveness in handling large-scale and complex datasets.

CO4: Design scalable cloud-based big data analytics solutions by selecting appropriate architectures, platforms, and analytical methods for real-world applications.

Reference:

1. Rajkumar Buyya, James Broberg, Andrzej Goscinski, Cloud Computing: Principles and Paradigms, Wiley.
2. Tom White, Hadoop: The Definitive Guide, O'Reilly Media.
3. Vignesh Prajapati, Big Data Analytics with R and Hadoop, Packt Publishing.
4. Kleppmann, Martin, Designing Data-Intensive Applications, O'Reilly.
5. Alan Gates, Programming Pig, O'Reilly.
6. Jure Leskovec et al., Mining of Massive Datasets, Cambridge University Press.

E-resources:

1. https://www.nasa.gov/smallsat-institute/sst-soa/thermal-control/

	CO-PO Mapping			CO-PSO Mapping	
PO & PSO / CO	**PO1**	**PO2**	**PO3**	**PSO1**	**PSO2**
CO1	-	-	-	-	-
CO2	3	-	-	3	-
CO3	-	-	2	-	2
CO4	-	1	-	3	-
AVERAGE	3	1	2	3	2

1-Low, 2-Medium, 3-High

26PC203T	QUANTUM COMPUTING	L	T	P	C
		2	0	0	2

Course Objectives:

- To provide a mathematical foundation for Quantum Computing and provide the basics of working.
- To interpret the various aspects and applications of quantum computing.
- To examine the factors that affect Quantum computation.

Text Books:

1. Chuck Easttom, "Quantum Computing Fundamentals", 1st edition, Addison-Wesley Professional (June 1st 2021).
2. Qiskit TextBook - https://qiskit.org/textbook/preface.html (2022)

Unit I	PHYSICAL PROPERTIES AND MATHEMATICAL FOUNDATIONS	6 Hours

Double Slit Experiment; Light: Particle Vs Wave; Heisenberg Uncertainty Principle. Vector spaces - basis; Inner product; Outer product; Tensor product; Linear operators.

Activity: Simulate the double-slit experiment using an interactive virtual lab; Construct simple operators and visualize action on vectors.

Unit II	QUANTUM COMPUTING POSTULATES AND GATES	6 Hours

Review of postulates, Bloch sphere, Single qubit states and gates, superposition; Two Qubit States and Gates - Bell States, Entanglement, CNOT gate, Phase oracles, Pauli Gates.

Activity: Group quiz to match postulates to physical implications; Visualization with Bloch sphere simulators.

Unit III	QUANTUM COMPUTING CIRCUITS	6 Hours

Dirac's notation for quantum computing, Computational Basis, Orthonormality, Hadamard and Phase Gates - building quantum circuits.

Activity: Use IBM Q Composer to build and simulate custom circuits; Apply X, H, Z gates and observe results on simulators.

Unit IV	FUNDAMENTAL QUANTUM ALGORITHMS	6 Hours

Deutsch-Josza Algorithm, Grover search algorithm: Problem definition, Amplitude amplification, Grover oracle, diffuser, multiple solutions in the search space.

Activity: Construct DJ circuit for a 3-bit input function; Simulation of Grover's algorithm with multiple marked elements.

Unit V	PROGRAMMING ON A REAL QUANTUM COMPUTER	6 Hours

Coding a real time quantum computer via IBMQ to carry out basic quantum measurement and state analysis.

Activity: Connect Qiskit with IBMQ using personal API token; Create 1- and 2-qubit circuits and compare real and simulated results.

	Total Hours	30

Weightage: Continuous Assessment: 40%, End Semester Theory Examination: 60%

Continuous Assessment: Assignments (30%), Quiz (10%), Virtual Demonstration (25%), Flipped Classroom (10%), Review of GATE & IES Questions (25%)

Course outcomes:

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

CO1: Describe the fundamental principles, postulates, and computational models of quantum computing and their significance in next-generation computing systems.

CO2: Analyze quantum algorithms and quantum circuit models to understand their computational advantages, limitations, and performance characteristics.

CO3: Evaluate quantum computing paradigms, error correction techniques, and hardware technologies for their suitability in solving complex computational problems.

CO4: Design quantum circuits and algorithmic solutions by selecting appropriate quantum gates, qubit architectures, and computational models for real-world problem scenarios.

Reference:

1. Kasirajan, Venkateswaran. Fundamentals of quantum computing. Springer International Publishing, 2021.
2. Chris Bernhardt, Quantum Computing for Everyone, The MIT Press, Cambridge, 2020.
3. Nielsen, Michael A., and Isaac L. Chuang, "Quantum Computation and Quantum Information" Cambridge University Press (5 April 2013).

	CO-PO Mapping			CO-PSO Mapping	
PO & PSO / CO	**PO1**	**PO2**	**PO3**	**PSO1**	**PSO2**
CO1	-	-	-	-	-
CO2	3	-	-	3	-
CO3	-	-	2	-	2
CO4	-	1	-	3	-
AVERAGE	3	1	2	3	2

1-Low, 2-Medium, 3-High

26PC204I	MULTICORE ARCHITECTURES	L	T	P	C
		3	0	2	4

Course Objectives:

- This course enables students to learn parallel programming models, tools, and techniques using shared and distributed memory systems.
- The course emphasizes the implementation and performance evaluation of parallel applications using OpenMP and MPI.

Unit I	MULTICORE SYSTEMS AND ARCHITECTURES	7 Hours

Evolution from Single-core to Multicore Systems - SIMD vs MIMD Architectures - Shared and Distributed Memory Systems - Cache Coherence and Memory Consistency.

Activity: Write a program to create and execute multiple threads using Pthreads; demonstrate mutex locks using std::mutex.

Unit II	THREAD MANAGEMENT AND SYNCHRONIZATION TECHNIQUES	7 Hours
Synchronization primitives: Mutex - Locks - Semaphores - Data races and deadlocks - Inter-thread communication: Condition variables - Pipes. **Activity:** Compute the Fibonacci series using OpenMP tasks; write a Rust program to spawn threads and use channels for communication.		
Unit III	**PARALLEL PROGRAMMING USING OPENMP FOR SHARED MEMORY**	**7 Hours**
OpenMP architecture and memory model - Directives and work-sharing constructs - Parallelizing Loops and Task Management - Performance optimization. **Activity:** Write a program using MPI to calculate the average of an array using collective communication.		
Unit IV	**MESSAGE PASSING PROGRAMMING WITH MPI**	**7 Hours**
MPI programming concepts - Point-to-point and collective communication - Derived data types and synchronization - Performance tuning. **Activity:** Demonstrate data race and synchronization using the threading module; perform matrix multiplication using OpenMP.		
Unit V	**EVALUATING PARALLEL PERFORMANCE, SCALABILITY AND CASE STUDIES**	**7 Hours**
Measuring and analyzing performance - Scalability metrics - Use of Profiling Tools for Performance Bottlenecks. Tree search, N-body simulation, graph algorithms - Comparative Analysis of OpenMP and MPI Implementations - Trends in multicore and manycore processors. **Activity:** Parallelize a for loop using the rayon crate in Rust; profile performance using gprof or perf; mini project implementing parallel merge sort/BFS using OpenMP and MPI.		
	Total Hours	**35**
	Practical Hours	**30**
Weightage: Continuous Assessment: 60%, End Semester Theory Examinations: 40%		
Continuous Assessment: Assignments (30%), Quiz (10%), Virtual Demo (25%), Flipped Class Room (10%), Review of GATE and IES Questions (25%)		

Course outcomes:

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

CO1: Describe the fundamental concepts, evolution, and components of multicore processor architectures and their role in modern computing systems.

CO2: Analyze the performance, parallelism, and scalability issues of multicore architectures using appropriate architectural and programming models.

CO3: Evaluate different multicore design techniques, cache coherence protocols, and interconnection networks for their effectiveness in specific computing scenarios.

CO4: Design efficient multicore-based solutions by selecting suitable architectures, synchronization mechanisms, and parallel execution strategies for real-world applications.

Reference:

1. Michael J. Quinn, Parallel Programming in C with MPI and OpenMP, McGraw-Hill, 2004.
2. Yan Solihin, Fundamentals of Parallel Multicore Architecture, CRC Press, 2015.
3. Victor Alessandrini, Shared Memory Application Programming, Morgan Kaufmann, 2015.
4. Rohit Chandra et al., Parallel Programming in OpenMP, Morgan Kaufmann, 2000.
5. Gerassimos Barlas, Multicore and GPU Programming, Morgan Kaufmann, 2014.

E-resources:

1. NPTEL Course: https://nptel.ac.in/courses/106104189
2. OpenMP Official Documentation: https://www.openmp.org
3. MPI Forum: https://www.mpi-forum.org
4. Coursera: https://www.coursera.org/learn/parallel-programming-in-java
5. Book PDF: Multicore Application Programming by Darryl Gove (Intel site)

	CO-PO Mapping			CO-PSO Mapping	
PO & PSO / CO	**PO1**	**PO2**	**PO3**	**PSO1**	**PSO2**
CO1	-	-	-	-	-
CO2	3	-	-	3	-
CO3	-	-	2	-	2
CO4	-	1	-	3	-
AVERAGE	3	1	2	3	2

1-Low, 2-Medium, 3-High