

NANDHA COLLEGE OF TECHNOLOGY

(An Autonomous Institution Affiliated to Anna University Chennai)

Erode-52

www.nandhatech.org



REGULATIONS – 2025

CURRICULUM AND SYLLABUS

M.E - COMPUTER SCIENCE AND ENGINEERING

Chairman-Board of Studies

Computer Science and Engineering

Nandha College of Technology (Autonomous)

Erode - 638 052.

VISION

- To emerge as a Centre of Excellence in Computer Science by achieving global standards in academics and research to meet the needs of society and industry.

MISSION

- To deliver value-based education in Computer Science and develop innovative, skilled, and competent Computer Engineers capable of fulfilling the increasing needs of society and industry.
- To empower students to enhance their technical skills and engage in research activities through lifelong learning.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

Graduates will be able to:

PEO 1: Pursue higher education, research, and advanced learning in the field of Computer Science and Engineering with strong analytical and problem-solving abilities.

PEO 2: Apply advanced computing knowledge, modern tools, and computational models to design sustainable and scalable solutions for complex real-world problems in industry and society.

PEO 3: Develop innovative and interdisciplinary solutions through research aptitude, technical competence, and entrepreneurial skills in emerging areas of Computer Science and Engineering.

PEO 4: Adapt to evolving technologies through lifelong learning, professional ethics, and continuous skill enhancement to achieve successful careers in academia, research, and industry.

PROGRAMME OUTCOMES

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.


Chairman-Board of Studies
Computer Science and Engineering
Nandha College of Technology (Autonomous)
Erode - 638 052.

PROGRAMME SPECIFIC OUTCOMES (PSOs)

PSO1: Advanced Computing and Problem Solving: Apply advanced algorithms, architectures, and computational models to analyze, design, and implement sustainable and scalable solutions that address complex problems across various domains, meeting the needs of industry and society.

PSO2: Research and Innovation Competence: Conduct independent research and utilize advanced tools and methodologies to develop innovative solutions for real-world and interdisciplinary computing challenges, while exhibiting strong research aptitude.


Chairman-Board of Studies
Computer Science and Engineering
Nandha College of Technology (Autonomous)
Erode - 638 052.

REGULATIONS 2025
CURRICULUM AND SYLLABUS

SEMESTER – I

S No.	Course Code	Course Name	Course Type	Periods / Week		Credits	Category
				L-T- P	TCP		
1.	MA25C07	Advanced Mathematical Methods (CSIE)	T	3-1-0	4	4	BS
2.	CP25C01	Advanced Data Structures and Algorithms	LIT	3-0-4	7	5	ES (PC)
3.	CP25C02	Advanced Database Technologies	T	3-0-0	3	3	ES (PC)
4.	CP25C03	Advanced Operating Systems	T	3-0-0	3	3	ES (PC)
5.	CP25C04	Advanced Compiler Design	T	3-0-0	3	3	ES (PC)
6.	CP25101	Technical Seminar	-	0-0-2	2	1	SD
TOTAL				22	19	---	

SEMESTER – II

S No.	Course Code	Course Name	Course Type	Periods / Week		Credits	Category
				L-T- P	TCP		
1.	CP25201	Multicore Architectures	LIT	3-0-2	5	4	ES (PC)
2.	CP25C05	Artificial Intelligence and Machine Learning	T	3-0-0	3	3	ES (PC)
3.	CP25C06	Cloud and Big Data Analytics	T	3-0-0	3	3	ES (PC)
4.	CP25C07	Quantum Computing	T	2-0-0	2	2	ES (PC)
5.		Programme Elective I	T	3-0-0	3	3	ES (PE)
6.		Industry Oriented Course I	-	1-0-0	1	1	SD
7.	CP25202	Industrial Training	-	-	-	2	SD
8.		Self-Learning Course	-	-	-	1	-
TOTAL				17	19	---	

SEMESTER – III

S No.	Course Code	Course Name	Course Type	Periods / Week		Credits	Category
				L-T- P	TCP		
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.		Industry-Oriented Course II	-	1-0-0	1	1	SD
6.	CP25301	Project Work I	-	0-0-12	12	6	SD
TOTAL					25	19	---

SEMESTER – IV

S No.	Course Code	Course Name	Course Type	Periods / Week		Credits	Category
				L-T- P	TCP		
1.	CP25401	Project Work II	-	0-0-24	24	12	SD
TOTAL					24	12	---

PROGRAMME ELECTIVE COURSES (PE)

S No.	Course Code	Course Name	Course Type	Periods / Week		Credits
				L-T- P	TCP	
1.	CP25C08	Advanced Software Testing and Quality Assurance	T	3-0-0	3	3
2.	CP25C09	Agile Methodologies	T	3-0-0	3	3
3.	CP25C10	Web of Things	T	3-0-0	3	3
4.	CP25001	Text and Speech Processing	T	3-0-0	3	3
5.	CP25C11	Advanced Deep Learning and Neural Networks	T	3-0-0	3	3
6.	CP25C12	Quantum Cryptography	T	3-0-0	3	3
7.	CP25C13	Quantum Machine Learning	T	3-0-0	3	3
8.	CP25C14	AI in IoT	T	3-0-0	3	3
9.	CP25C15	Web 3.0	T	3-0-0	3	3
10.	CP25C16	Advanced Large Language Models	T	3-0-0	3	3
11.	CP25C17	Edge and Fog Computing	T	3-0-0	3	3
12.	CP25C18	Green Computing and Sustainability	T	3-0-0	3	3
13.	CP25C19	Cognitive Computing	T	3-0-0	3	3
14.	CP25C20	Agentic AI	T	3-0-0	3	3
15.	CP25C21	Mixed Reality	T	3-0-0	3	3
16.	CP25C22	Blockchains Architecture and Design	T	3-0-0	3	3
17.	CP25C23	Human-Centered AI	T	3-0-0	3	3
18.	CP25C24	Vibe Coding	T	3-0-0	3	3
19.	CP25C25	Federated Learning	T	3-0-0	3	3
20.	CP25002	Deep Learning for Computer Vision	T	3-0-0	3	3


 Chairman-Board of Studies
 Computer Science and Engineering
 Nandha College of Technology (Autonomous)
 Erode - 638 052.

MA25C07	Advanced Mathematical Methods (CSIE)	L	T	P	C
		3	1	0	4
Course Objectives: <ul style="list-style-type: none">• 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.					
Linear Algebra: 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.					
Multivariate Analysis: 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.					
Elementary Number Theory: 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.					
Advanced Number Theory: Advanced Number Theory, Primality Testing algorithms, Chinese Remainder Theorem, Quadratic Congruence, Discrete Logarithm, Factorization Methods, Side Channel Attacks, Shannon Theory, Perfect Secrecy, Semantic Security.					
Weightage: Continuous Assessment: 40%, End Semester Examinations: 60%.					
Assessment Methodology: Assignments (15), Quiz (10), Virtual Demo (20), Flipped Class Room (10), Review of Gate and IES Questions (25), Project (20).					
References: <ol style="list-style-type: none">1. Gilbert Strang, <i>Linear Algebra and Its Applications</i>, Cengage Learning.2. Richard A. Johnson & Dean W. Wichern, <i>Applied Multivariate Statistical Analysis</i>, Pearson.3. Neal Koblitz, <i>A Course in Number Theory and Cryptography</i>, Springer.4. Victor Shoup, <i>A Computational Introduction to Number Theory and Algebra</i>, Cambridge University Press.					

E-resources:

1. <https://ocw.mit.edu/courses/18-06-linear-algebra>
2. <https://nptel.ac.in/courses/111105041>
3. <https://crypto.stanford.edu/pbc/notes/numbertheory>

CP25C01	Advanced Data Structures and Algorithms	L	T	P	C
		3	0	4	5

Course Objectives:

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

Linear Data Structures and Memory Optimization: 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.

Practical:

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

Advanced Tree Data Structures: 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.

Practical:

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

Graph Data Structures and Algorithms: 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.

Practical:

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

Algorithm Design and Paradigms: 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.

Practical:

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

Computational Complexity and Approximation Algorithms: 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.

Practical:

- Implement approximation algorithm for vertex cover.
- Complexity analysis of a chosen NP-hard problem and implement a heuristic.

Advanced Topics and Emerging Trends: Randomized Algorithms – Monte Carlo Algorithms, Parallel and Distributed Algorithms – PRAM Model, Divide and Conquer in Parallel, Load Balancing, Streaming Algorithms – Data Stream Models, Sketching and Sampling, Frequency Moments, Advanced String Matching – Suffix Trees, Suffix Arrays, Pattern Matching in Linear Time.

Practical:

- Implement randomized and streaming algorithms on real-world datasets.
- Design of parallel and distributed algorithms.

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

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

References:

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/afriid18/Data_structures_and_algorithms_in_cpp
3. <https://www.udemy.com/course/introduction-to-algorithms-and-data-structures-in-c/?srsltid=AfmBOorEZIkqV7QzaEh6lqzAaKLjC-lpFU1NGqWFOHMLhOos-uDVKjCK>

	Description of CO	PO	PSO
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.	PO1(3)	PSO1(3)
CO3	Evaluate algorithmic techniques and data structures to determine their suitability for different applications.	PO3(2)	PSO2(2)
CO4	Design optimized solutions for real-world problems using appropriate algorithms and data structures.	PO2(1)	PSO1(3)

CP25C02	Advanced Database Technologies	L	T	P	C
		3	0	0	3

<p>Course Objectives:</p> <ul style="list-style-type: none"> • 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.
<p>Entity Relationship Model: Entity Relationship Model Revised-Subclasses, Superclasses and Inheritance -Specialization and Generalization-Union Types-Aggregation.</p> <p>Activity: Design ER Model for a specific use case.</p>
<p>Enhanced Entity Relational Model: Relational Model Revised, Converting ER and EER Model to Relational Model-SQL and Advanced Features, File Structures, Hashing, and Indexing.</p> <p>Activity: Demonstration of SQL Implementation.</p>
<p>Object Relational Databases: 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.</p> <p>Activity: Demonstration of Object Query Language.</p>
<p>Query Processing and Optimization: Query Processing, Query Trees and Heuristics, Query Execution Plans, Cost Based Optimization.</p> <p>Activity: Design of Query Evaluation Plans.</p>
<p>Distributed Databases: Real-Time Bidding, E-mail Marketing, Affiliate Marketing, Social Marketing Mobile Marketing, Distributed Database Concepts, Data Fragmentation, Replication and Allocation, Distributed Database Design Techniques, Distributed Database Design Techniques, Distributed Database Architectures.</p> <p>Activity: Demonstration of Concurrency and Transactions.</p>
<p>NOSQL Systems and Bigdata: 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.</p> <p>Activity: Design application with MongoDB.</p>

Advanced Database Models, Systems and Applications: 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: Demonstration of Hadoop infrastructure for Big Data Analytics.

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

Assessment Methodology: Assignments (15), Quiz (10), Virtual Demo (20), Flipped Class Room (10), Review of Gate and IES Questions (25), Project (20).

References:

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>

	Description of CO	PO	PSO
CO1	Elaborate different database models for effective database design.	--	--
CO2	Implement advanced database features for optimized data retrieval.	PO1(3)	PSO1(3)
CO3	Evaluate query processing and optimization strategies to improve system performance.	PO3(2)	PSO2(2)
CO4	Design solutions using advanced database models to address complex data-intensive applications.	PO2(1)	PSO1(3)

CP25C03	Advanced Operating Systems	L	T	P	C
		3	0	0	3
Course Objectives: <ul style="list-style-type: none">• 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.					
Advanced Process and Thread Management: 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.					
Memory and Resource Management in Modern OS: 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.					
Virtualization and Containerization: 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.					
Distributed Operating Systems and File Systems: Distributed scheduling, communication, and synchronization-Distributed file systems: NFS, GFS, HDFS-Transparency issues and fault tolerance Activity: Simulate distributed process synchronization.					
Security and Trust in Operating Systems: Access control models: DAC, MAC, RBAC-OS hardening techniques, sandboxing, SELinux, AppArmor-Secure boot, rootkit detection, trusted execution environments Activity: Implement Role-Based Access Control (RBAC) using Linux user and group permissions.					
Real-Time and Embedded Operating Systems: Real-time scheduling algorithms (EDF, RM)-POSIX RT extensions, RTOS architecture-TinyOS, FreeRTOS case studies Activity: Analyze FreeRTOS task scheduling behavior.					
Edge and Cloud OS: Future Paradigms: Serverless OS, unikernels, lightweight OS for edge computing-Mobile OS internals (Android, iOS)-OS for quantum and neuromorphic computing (intro) Activity: Analyze Android's system architecture using emulator tools.					

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

Assessment Methodology: Assignments (15), Quiz (10), Virtual Demo (20), Flipped Class Room (10), Review of Gate and IES Questions (25), Project (20).

References:

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>

	Description of CO	PO	PSO
CO1	Describe operating system concepts for memory and resource management.	--	--
CO2	Analyse virtualization and distributed OS mechanisms for scalability and performance.	PO1(3)	PSO1(3)
CO3	Evaluate OS security and resource handling strategies in diverse environments.	PO3(2)	PSO2(2)
CO4	Design innovative OS solutions using modern tools and techniques.	PO2(1)	PSO1(3)

CP25C04	Advanced Compiler Design	L	T	P	C
		3	0	0	3
Course Objective: <ul style="list-style-type: none">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.					
Intermediate Representations and Control Flow Analysis: Static single assignment (SSA) form- Context-Free Grammar (CFG) construction-dominance relations-Intermediate Representation (IR) design for functional and imperative languages-Static single assignment and def-use chains					
Activities: <ol style="list-style-type: none">Convert source code to SSA form using LLVM IR.Visualize control flow graphs from SSA using LLVM tools.					
Program Analysis and Transformations: Data flow analysis- live variable analysis-reaching definitions-Alias analysis and dependence analysis-Loop optimizations and transformations					
Activities: <ol style="list-style-type: none">Perform loop unrolling and strength reduction.Conduct live variable analysis and visualize data flow graphs.					
Advanced Optimizations and Polyhedral Compilation: Polyhedral model for loop nests-Tiling, skewing, fusion, and vectorization-Profile-guided and feedback-directed optimizations					
Activities: <ol style="list-style-type: none">Implement loop tiling and loop skewing on a matrix multiplication program.Analyze the effect on loop-intensive code with LLVM optimization flags.					
Just-in-Time (JIT) and Runtime Compilation: JIT compilation models: tracing, method-based-GraalVM architecture, Java HotSpot internals-LLVM JIT and dynamic language support					
Activities: <ol style="list-style-type: none">Develop a basic JIT-enabled interpreter with LLVM or GraalVM.Implement dynamic dispatch using LLVM JIT API.					

Machine Learning in Compiler Design: ML for phase ordering, auto-tuning, and IR prediction-Reinforcement learning for optimization passes-Dataset creation and benchmarking for compiler ML

Activities:

1. Train an ML model to predict optimization passes.
2. Use reinforcement learning for pass selection in toy compiler.

Domain-Specific Languages (DSLs) and Compiler Extensions: Designing DSLs for AI/ML, DSP, graphics-Code generation for custom accelerators-Integration with TensorFlow XLA and Halide

Activities:

1. Design and test a simple DSL grammar using ANTLR.
2. Integrate a DSL with TensorFlow XLA or Halide.

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

Activities:

1. Use CompCert to verify compilation of simple programs.
2. Apply compiler fuzzing using tools like libFuzzer.

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

Assessment Methodology: Assignments (15), Quiz (10), Virtual Demo (20), Flipped Class Room (10), Review of Gate and IES Questions (25), Project (20).

References:

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. Völter, 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. AmeyKarkare, IIT Kanpur, "Advanced Compiler Optimizations"
Link: <https://www.cse.iitk.ac.in/users/karkare/Courses/cs738/>
2. Prof. Santanu Chattopadhyay, "Compiler Design", IIT Kharagpur
Link: https://onlinecourses.nptel.ac.in/noc21_cs07/preview

	Description of CO	PO	PSO
CO1	Explain intermediate control flow techniques in compiler design.	--	--
CO2	Apply program analysis techniques and advanced optimizations for design of compilers.	PO1(3)	PSO1(3)
CO3	Develop compiler features and machine learning techniques for optimization.	PO3(2)	PSO2(2)
CO4	Evaluate secure compilation strategies for quantum and multi-target compilation.	PO2(1)	PSO1(3)

CP25201- MULTICORE ARCHITECTURE		L	T	P	C
		3	0	2	4
Course Objective	<ul style="list-style-type: none"> To understand parallel programming models, tools, and techniques for shared and distributed memory systems. To implement and evaluate the performance of parallel applications using OpenMP and MPI. 				
Course Outcomes: The Student will be able to					Cognitive Level
CO1	Apply multicore architectures and memory models to design parallel solutions.	Ap			
CO2	Analyze thread management and synchronization techniques to handle concurrency issues.	An			
CO3	Develop parallel programs using OpenMP and MPI for shared and distributed systems.	Ap			
CO4	Evaluate performance and scalability of parallel programs using appropriate metrics and tools.	Ev			
CO5	Design advanced parallel algorithms and compare different parallel programming models.	C			

UNIT I – MULTICORE SYSTEMS AND ARCHITECTURES	9
Multicore Systems and Architectures: Evolution from single-core to multicore systems – SIMD vs MIMD architectures – Shared and distributed memory systems – Cache coherence – Memory consistency models.	
UNIT II – THREAD MANAGEMENT AND SYNCHRONIZATION TECHNIQUES	9
Thread management – Synchronization primitives: mutex, locks, semaphores – Data races and deadlocks – Inter-thread communication: condition variables, pipes.	
UNIT III – PARALLEL PROGRAMMING USING OPENMP AND MPI	9
Parallel Programming using OpenMP: OpenMP architecture and memory model – Directives and work-sharing constructs – Parallelizing loops – Task management – Performance optimization. Message Passing Programming with MPI: MPI programming concepts – Point-to-point and collective communication – Derived data types – Synchronization – Performance tuning.	

UNIT IV – PERFORMANCE EVALUATION AND SCALABILITY	9
Evaluating Parallel Performance and Scalability: Measuring and analyzing performance – Scalability metrics – Speedup and efficiency – Use of profiling tools for performance bottlenecks.	
UNIT V – ADVANCED PARALLEL ALGORITHMS AND CASE STUDIES	9
Advanced concepts: Tree search, N-body simulation, graph algorithms – Comparative analysis of OpenMP and MPI implementations – Trends in multicore and many core processors.	
TOTAL PERIODS	45
List of Experiments:	
<ol style="list-style-type: none"> 1. Write a program to create and execute multiple threads using Pthreads. 2. Write a program to demonstrate mutex locks using std::mutex. 3. Write a program to compute the Fibonacci series using OpenMP tasks. 4. Write a Rust program to spawn threads and use channels for communication. 5. Write a program using MPI to calculate the average of an array using collective communication. 6. Write a program to perform matrix multiplication using OpenMP. 7. Write a program using threading module to demonstrate data race and synchronization. 8. Write a program and profile its performance using gprof or perf. 9. Write a Rust program to parallelize a for loop using the rayon crate. 10. Mini Project: Implement parallel merge sort or breadth-first search using OpenMP and MPI and compare complexity. 	
HARDWARE / SOFTWARE REQUIRED FOR A BATCH OF 30 STUDENTS:	
HARDWARE REQUIRED:	
<ul style="list-style-type: none"> • Computer Systems – 30 Nos. (1 per student) • Processor – Intel Core i5 / AMD Ryzen 5 or higher (Multicore preferred) • RAM – Minimum 8 GB (16 GB recommended for parallel workloads) • Storage – 500 GB HDD / 256 GB SSD • Network – High-speed LAN connectivity (for MPI experiments) • Server / Cluster Node (Optional) – 1 No. for distributed computing • GPU-enabled System (Optional) – For advanced parallel computing experiments 	

SOFTWARE REQUIRED:

- Operating System – Ubuntu (preferred) / Windows 10 or higher
- Programming Languages – C, C++, Rust, Python
- Parallel Programming Tools –
 - OpenMP
 - MPI (e.g., MPICH / OpenMPI)
 - Pthreads
- Development Tools – Visual Studio Code / GCC
- Libraries / Packages –
 - pthread library
 - OpenMP library
 - MPI libraries
 - Rust crates (rayon, std::thread)
- Performance & Profiling Tools –
 - gprof
 - perf

TOTAL PERIODS**30****References:**

1. Michael J. Quinn, "Parallel Programming in C with MPI and OpenMP", McGrawHill, 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)

Mapping of COs with POs / PSOs					
COs	POs			PSOs	
	1	2	3	1	2
1	3	2	1	2	1
2	2	2	3	3	1
3	2	2	3	3	2
4	2	2	3	3	2
5	3	3	3	3	3
CO (W.A)	2.6	2.2	2.6	2.8	1.8


 Chairman-Board of Studies
 Computer Science and Engineering
 Nandha College of Technology (Autonomous)
 Erode - 638 052.

CP25C05 – ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING		L	T	P	C
		3	0	0	3
Course Objective	<ul style="list-style-type: none"> 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, focusing on supervised and unsupervised learning methodologies. 				
Course Outcomes: The Student will be able to					Cognitive Level
CO1	Explain intelligent agents and apply search techniques to solve AI problems.				Ap
CO2	Apply knowledge representation and logical reasoning methods to infer solutions.				Ap
CO3	Analyze decision-making under uncertainty using probabilistic models and game theory.				An
CO4	Apply and evaluate supervised machine learning algorithms for prediction tasks.				Ap
CO5	Analyze unsupervised and probabilistic learning techniques for data analysis and clustering.				An

UNIT I – PROBLEM SOLVING AND KNOWLEDGE REPRESENTATION	9
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.	
UNIT II – UNCERTAIN KNOWLEDGE AND REASONING	9
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.	
UNIT III – MACHINE LEARNING INTRODUCTION	9
Types of learning – Hypothesis space – Inductive bias – Evaluation – Cross validation – Bias variance – Bias Variance Trade off.	
UNIT IV – SUPERVISED LEARNING	9
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.	

UNIT V – UNSUPERVISED AND PROBABILISTIC LEARNING	9
Clustering techniques: Clustering: partition, hierarchical and density based clustering – Self organizing maps – Expectation maximization – Gaussian mixture models – Principal component analysis – Bayesian learning: Bayes optimal classifier, Naïve Bayes classifier, Bayesian belief networks.	
TOTAL PERIODS	45
REFERENCES:	
1. Stuart Russell and Peter Norvig “Artificial Intelligence - A Modern Approach”, Prentice Hall, 3 rd Edition, 2011. 2. Elaine Rich, Kevin Knight and Shiv Shankar B. Nair, “Artificial Intelligence”, 3 rd Edition, Tata McGraw Hill, 2009. 3. Tom Mitchell, “Machine Learning”, McGraw Hill, 3 rd 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.	

Mapping of COs with POs / PSOs					
COs	POs			PSOs	
	1	2	3	1	2
1	2	1	2	2	1
2	2	2	2	2	2
3	3	2	3	3	2
4	3	2	3	3	2
5	3	3	3	3	3
CO (W.A)	2.6	2.0	2.6	2.6	2.0

CP25C06 - CLOUD AND BIG DATA ANALYTICS		L	T	P	C
		3	0	0	3
Course Objective	<ul style="list-style-type: none"> To explore Cloud and big data Terminologies, Hadoop ecosystem and processing tools. To analyze, store, and visualize large-scale data. To develop and deploy cloud-based big data applications. 				
Course Outcomes: The Student will be able to					Cognitive Level
CO1	Describe the fundamental concepts, architectures, and service models of cloud computing and big data analytics used in modern data-driven environments.				U
CO2	Analyze cloud-based big data processing frameworks and analytics techniques to understand their performance, scalability, and resource utilization				An
CO3	Evaluate big data analytics tools, cloud platforms, and data processing models for their effectiveness in handling large-scale and complex datasets				Ap
CO4	Design scalable cloud-based big data analytics solutions by selecting appropriate architectures, platforms, and analytical methods for real-world applications.				Ap
CO5	Explain cloud-based data storage, computing services, security mechanisms, and analytics applications, and analyze the role of cloud-native architectures in real-world domains such as smart cities and healthcare.				An

UNIT I - CLOUD COMPUTING	9
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.	
UNIT II - BIG DATA ECOSYSTEM & HADOOP	9
Big Data characteristics (Volume, Velocity, Variety, Veracity, Value) - Challenges in Big Data - Hadoop Architecture and HDFS - MapReduce Programming Model - YARN.	
UNIT III - DATA STORAGE & PROCESSING TECHNIQUES	9
NoSQL Databases: MongoDB, Cassandra, HBase - Data Ingestion: Sqoop, Flume - Hive, Pig - Architecture and Querying - Data Cleaning and Preparation.	
UNIT IV - ADVANCED BIG DATA ANALYTICS	9
Introduction to Apache Spark - Spark vs. MapReduce - Spark SQL, Data Frames, RDDs- Machine Learning using Spark MLlib - Real-time processing: Kafka and Spark Streaming.	

UNIT V - CLOUD INTEGRATION AND SECURITY	9
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.	
TOTAL PERIODS	45

REFERENCES:

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.

Mapping of COs with POs / PSOs					
COs	POs			PSOs	
	1	2	3	1	2
1	-	-	-	-	-
2	3	-	-	3	-
3	-	-	2	-	2
4	-	1	-	3	-
5	3	3	2	3	2
CO (W.A)	1.2	0.6	0.8	1.8	0.8

CP25C07 - QUANTUM COMPUTING		L	T	P	C
		2	0	0	2
Course Objective:	<ul style="list-style-type: none"> To provide a mathematical foundation for quantum computing and the basics of its working. To interpret the various aspects and applications of quantum computing and to examine the factors that affect quantum computation. 				
Course Outcomes: The Student will be able to					Cognitive Level
CO1	Analyze the fundamental physical principles and mathematical foundations of quantum computing, including wave-particle duality, uncertainty principle, and vector space operations.	An			
CO2	Analyze the postulates of quantum mechanics and the behavior of single- and two-qubit systems using quantum gates, Bloch sphere representation, and entanglement concepts.	An			
CO3	Apply Dirac notation, computational basis, and quantum gate operations to construct and analyze basic quantum circuits.	Ap			
CO4	Evaluate the working principles, efficiency, and performance of fundamental quantum algorithms such as Deutsch-Jozsa and Grover's search algorithm.	E			
CO5	Create and implement simple quantum programs on IBM Quantum hardware (IBMQ) for measurement, state preparation, and quantum state analysis.	C			

UNIT 1 PHYSICAL PROPERTIES AND MATHEMATICAL FOUNDATIONS	(9)
Double Slit Experiment; Light: Particle Vs Wave; Heisenberg Uncertainty Principle. Vector spaces – basis; Inner product; Outer product; Tensor product; Linear operators	
UNIT II QUANTUM COMPUTING POSTULATES AND GATES	(9)
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.	
UNIT III QUANTUM COMPUTING CIRCUITS	(9)
Dirac's notation for quantum computing, Computational Basis, Orthonormality, Hadamard and Phase Gates- building quantum circuits	
UNIT IV FUNDAMENTAL QUANTUM ALGORITHMS	(9)
Deutsch-Josza Algorithm, Grover search algorithm: Problem definition, Amplitude amplification, Grover oracle, diffuser, multiple solutions in the search space	
UNIT V PROGRAMMING ON A REAL QUANTUM COMPUTER	(9)
Coding areal time quantum computer via IBMQ to carryout basic quantum measurement and state analysis.	
TOTAL PERIODS	45

REFERENCES:

1. Michael A. Nielsen, Isaac L. Chuang, "Quantum Computation and Quantum Information", Cambridge University Press, 10th Anniversary Edition, 2010.
2. Noson S. Yanofsky, Mirco A. Mannucci, "Quantum Computing for Computer Scientists", Cambridge University Press, 1st Edition, 2008.
3. Phillip Kaye, Raymond Laflamme, Michele Mosca, "An Introduction to Quantum Computing", Oxford University Press, 1st Edition, 2007.
4. Venkateswaran Kasirajan, "Fundamentals of Quantum Computing: Theory and Practice", Springer International Publishing, 1st Edition, 2021.

Mapping of COs with POs / PSOs					
COs	POs			PSOs	
	1	2	3	1	2
1	-	-	2	2	-
2	-	-	2	2	-
3	3	-	-	3	-
4	3	-	-	-	2
5	-	2	-	-	3
CO (W.A)	3.0	2.0	2.0	2.3	2.5

CP25C08 – ADVANCED SOFTWARE TESTING AND QUALITY ASSURANCE		L	T	P	C
		3	0	0	3
Course Objective	<ul style="list-style-type: none"> To understand advanced software testing methodologies, quality assurance principles, and modern testing practices used in Agile and DevOps environments. To apply automated testing frameworks, quality models, and test process improvement techniques for developing reliable and high-quality software systems. 				
Course Outcomes: The Student will be able to					Cognitive Level
CO1	Apply software testing fundamentals, STLC phases, and testing strategies in Agile and DevOps environments for software quality assurance.				Ap
CO2	Apply different testing techniques and risk-based testing approaches for software quality evaluation.				Ap
CO3	Analyze advanced test design techniques and develop optimized test cases for real-world applications.				An
CO4	Apply automated testing frameworks and continuous testing tools for software validation.				Ap
CO5	Analyze software quality assurance frameworks, metrics, and test process improvement models for enterprise applications.				An

UNIT I – FOUNDATIONS OF SOFTWARE TESTING AND TEST LIFECYCLE	9
Introduction to software testing – Evolution of software testing – Role of testing in Agile and DevOps – Software Test Life Cycle (STLC) phases – Requirement analysis, test planning, test design, test execution, defect tracking, test closure – Quality attributes: reliability, maintainability, usability, testability – Testing principles and classifications – Verification and validation techniques.	
UNIT II – TESTING TYPES AND RISK-BASED TESTING	9
Types of testing: Unit testing, integration testing, system testing, acceptance testing – Regression, smoke, sanity, alpha and beta testing – Functional and non-functional testing – Risk-based testing concepts – Test prioritization and risk assessment – Defect severity and priority – Test strategy formulation – Testing in CI/CD environments.	
UNIT III – ADVANCED TEST DESIGN TECHNIQUES	9
Black-box testing techniques: Equivalence Class Partitioning (ECP), Boundary Value Analysis (BVA), decision table testing – White-box testing: control flow testing, data flow testing, path testing – State transition testing – Use-case testing – Model-based testing – Cause-effect graphing – Mutation testing – Fault injection and fuzz testing – Test case minimization and prioritization.	
UNIT IV – TEST AUTOMATION AND CONTINUOUS TESTING	9
Automation testing frameworks – Reusable and maintainable automation architecture – Selenium WebDriver – TestNG and JUnit – Robot Framework and Cypress – Behavior Driven Development (BDD) and Test Driven Development (TDD) using Cucumber – Performance testing using JMeter – Introduction to LoadRunner concepts – Continuous testing with Jenkins and GitHub Actions – Test reporting and automation metrics.	

UNIT V – SOFTWARE QUALITY ASSURANCE AND TEST PROCESS IMPROVEMENT	9
Software Quality Assurance (SQA): quality planning, quality control, quality assurance and quality improvement – Reviews, inspections and audits – Quality models: McCall, Boehm, ISO 9126 and ISO 25010 – Testing metrics: defect density, test coverage, MTBF, defect leakage – Cost of quality – Test maturity models: TMMi and CMMI-Dev – Benchmarking and process audits – AI/ML-based testing approaches – Shift-left and shift-right testing trends.	
TOTAL PERIODS	45
References: <ol style="list-style-type: none"> 1. Glenford J. Myers, Corey Sandler and Tom Badgett, “The Art of Software Testing”, Wiley, 3rd Edition, 2011. 2. Paul C. Jorgensen, “Software Testing: A Craftsman’s Approach”, CRC Press, 4th Edition, 2013. 3. Ron Patton, “Software Testing”, Sams Publishing, 2nd Edition, 2007. 4. Aditya P. Mathur, “Foundations of Software Testing”, Pearson Education, 2nd Edition, 2013. 5. Roger S. Pressman and Bruce R. Maxim, “Software Engineering: A Practitioner’s Approach”, McGraw-Hill, 8th Edition, 2019. 	
E-Resources: <ol style="list-style-type: none"> 1. MIT OpenCourseWare (Software Testing) 2. NPTEL (Software Testing by Prof. Nandini Prasad, IIT Kharagpur) 	
TOTAL PERIODS	45

Mapping of POs and COs					
CO	PO			PSO	
	1	2	3	1	2
1	2	2	1	2	1
2	2	3	2	2	2
3	3	3	2	3	2
4	3	2	3	3	3
5	3	3	3	3	3
CO (W.A)	2.6	2.6	2.2	2.6	2.2

CP25C09 – AGILE METHODOLOGIES		L	T	P	C
		3	0	0	3
Course Objective:	<ul style="list-style-type: none"> To understand the principles, frameworks, and cultural practices of Agile software development and their role in modern software engineering. To apply Agile project management, engineering practices, and scaling techniques for effective software delivery in diverse organizational environments. 				
Course Outcomes: The Student will be able to					Cognitive Level
CO1	Apply Agile principles, values, frameworks, and cultural transformation practices in modern software development environments.				Ap
CO2	Apply Agile methodologies and project management techniques for iterative planning, collaboration, and software delivery.				Ap
CO3	Analyze Agile engineering practices, metrics, and team processes to improve software quality and productivity.				An
CO4	Evaluate Agile frameworks and scaling models to determine their suitability for different organizational and project environments.				E
CO5	Design Agile-based workflows and DevOps-enabled solutions for developing high-quality software systems.				C

UNIT I – AGILE FOUNDATIONS AND MINDSET	9
Evolution of Software Development Methodologies – Challenges in traditional software development models – Agile Manifesto – Agile Values and Principles – Agile mindset in individuals and teams – Characteristics of Agile teams and leaders – Lean Thinking in Software Development – Cultural shift from traditional to Agile thinking – Agile vs Traditional methodologies.	
UNIT II – AGILE PROCESS MODELS AND FRAMEWORKS	9
Scrum framework: Roles, artifacts, and events – Sprint planning, daily stand-ups, sprint review, retrospectives – Extreme Programming (XP): Pair programming, test-first development, refactoring – Kanban: Flow, WIP limits, pull systems – Feature Driven Development (FDD) – Crystal methodology – SAFe (Scaled Agile Framework) – Disciplined Agile Delivery (DAD) – LeSS and Nexus frameworks.	
UNIT III – AGILE PROJECT MANAGEMENT AND PLANNING	9
User stories and story mapping – Product backlog management and grooming – Sprint planning and release planning – Estimation techniques: Planning poker and T-shirt sizing – Agile metrics: Velocity, burndown charts, cumulative flow diagrams (CFD) – Risk management in Agile projects – Agile collaboration and communication practices – Agile tools: Jira, Trello, Azure DevOps.	
UNIT IV – AGILE ENGINEERING PRACTICES	9
Test-Driven Development (TDD) – Behaviour Driven Development (BDD) – Continuous Integration and Continuous Delivery (CI/CD) – DevOps principles and Agile integration – Refactoring techniques – Code smells and clean code practices – Design patterns in Agile development – Automated testing strategies and quality assurance in Agile projects.	

UNIT V – AGILE AT SCALE AND INDUSTRY APPLICATIONS	9
Agile scaling frameworks and enterprise adoption challenges – Distributed Agile teams and remote collaboration – Agile in non-software domains such as education, HR, and manufacturing – Agile governance and contracting – Case studies from Google, Spotify, IBM, and startups – End-to-end Agile project implementation using Scrum/Kanban and CI/CD pipeline.	
TOTAL PERIODS	45
References: <ol style="list-style-type: none"> 1. Ken Schwaber and Jeff Sutherland, Software in 30 Days: How Agile Managers Beat the Odds, Delight Their Customers, and Leave Competitors in the Dust, Wiley, 1st Edition, 2012. 2. Mike Cohn, User Stories Applied: For Agile Software Development, Addison-Wesley Professional, 1st Edition, 2004. 3. Craig Larman and Bas Vodde, Scaling Lean & Agile Development: Thinking and Organizational Tools for Large-Scale Scrum, Addison-Wesley Professional, 1st Edition, 2008. 4. Robert C. Martin, Clean Code: A Handbook of Agile Software Craftsmanship, Pearson Education (Prentice Hall), 1st Edition, 2008. 	
E-Resources: <ol style="list-style-type: none"> 1. NPTEL: Agile Software Development – Prof. Rajib Mall (IIT Kharagpur) 2. MIT Open Course Ware Agile content 	

Mapping of POs and COs					
COs	POs			PSOs	
	1	2	3	1	2
1	2	1	1	2	1
2	3	2	2	3	2
3	3	2	2	3	3
4	3	3	2	2	3
5	3	3	3	3	3
CO (W.A)	2.8	2.2	2.0	2.6	2.4

CP25C10 – WEB OF THINGS		L	T	P	C
		3	0	0	3
Course Objective	<ul style="list-style-type: none"> To understand the concepts, architectures, standards, and communication protocols of the Web of Things (WoT) for integrating smart devices with web technologies. To design interoperable, secure, and scalable WoT-based applications using Thing Descriptions, WoT scripting interfaces, and embedded system integration. 				
Course Outcomes: The Student will be able to					Cognitive Level
CO1	Apply the concepts, architectures, and enabling technologies of the Web of Things for connecting smart devices to web-based platforms.				Ap
CO2	Analyze Web of Things frameworks, communication protocols, and interoperability mechanisms in distributed environments.				An
CO3	Evaluate Web of Things platforms, security mechanisms, and data integration techniques for various smart application domains.				E
CO4	Design scalable and secure Web of Things-based solutions using appropriate architectures, protocols, and services.				C
CO5	Develop embedded device integration solutions using WoT frameworks and scripting interfaces for smart applications.				Ap

UNIT I – INTRODUCTION TO WEB OF THINGS	9
Evolution from Internet of Things (IoT) to Web of Things (WoT) – Architecture of WoT – Thing Description, Binding Templates, Servients – Protocol bindings – Overview of web communication protocols: HTTP, Web Sockets, MQTT, CoAP – WoT scripting and interaction models: Properties, Actions, and Events.	
UNIT II – WOT STANDARDS AND THING DESCRIPTION	9
W3C WoT architecture and building blocks – Thing Description (TD): Syntax and vocabulary – JSON-LD format – TD schema validation and interpretation – Semantic annotation in TD – Metadata modeling and semantic interoperability – WoT discovery mechanisms.	
UNIT III – IMPLEMENTATION OF WOT SERVIENTS	9
WoT Servient structure: Exposed Things and Consumed Things – Node-WoT framework – WoT Scripting API – Protocol bindings using HTTP, MQTT, and CoAP – Event-driven communication models – Integration of sensors and actuators with WoT applications.	
UNIT IV – INTEROPERABILITY, SECURITY, AND PRIVACY	9
Interoperability challenges in heterogeneous environments – Security mechanisms: HTTPS, OAuth2.0, and access control – Authentication and authorization – Privacy preservation and anonymization – Trust and identity management in WoT ecosystems – Secure communication practices in distributed smart environments.	

UNIT V – APPLICATIONS AND INTEGRATION TO WOT	9
Applications of WoT in Smart Cities, Smart Homes, e-Health, and Industrial IoT (IIoT) – Cloud and Edge integration in WoT – Data analytics and AI integration in WoT – Future trends: Digital Twins and Metaverse – Integration of Arduino, BeagleBone, Intel Edison, and other embedded systems with WoT frameworks.	
References: <ol style="list-style-type: none"> 1. Michael Kuniavsky, Smart Things: Ubiquitous Computing User Experience Design, Morgan Kaufmann (Elsevier), 1st Edition, 2010. 2. Dom Guinard and Vlad Trifa, Building the Web of Things, Manning Publications, 1st Edition, 2016. 3. Hakima Chaouchi, The Internet of Things: Connecting Objects to the Web, Wiley, 1st Edition, 2010. 4. Peter Waher, Mastering Internet of Things, Packt Publishing, 1st Edition, 2018. 5. Alasdair Gilchrist, Industry 4.0: The Industrial Internet of Things, Apress, 2nd Edition, 2021. 	
E-Resources: <ol style="list-style-type: none"> 1. W3C WoT Working Group Documents: https://www.w3.org/WoT/H 	
TOTAL PERIODS	45

Mapping of POs and COs					
COs	POs			PSOs	
	1	2	3	1	2
1	2	1	1	2	1
2	3	2	2	3	2
3	3	2	3	2	3
4	3	3	3	3	3
5	2	3	2	3	2
CO (W.A)	2.6	2.2	2.2	2.6	2.2


 Chairman-Board of Studies
 Computer Science and Engineering
 Nandha College of Technology (Autonomous)
 Erode - 638 052.

CP25001 - TEXT AND SPEECH PROCESSING		L	T	P	C
		3	0	0	3
Course Objective	<ul style="list-style-type: none"> To apply the fundamental concepts and techniques of text and speech processing, including text normalization, language modelling, speech signal analysis, speech recognition, speech synthesis and conversational systems for natural language and audio-based applications. To design and develop intelligent language and speech technologies using NLP methods, multilingual processing, modern deep learning architectures and large language models while addressing practical and ethical considerations. 				
Course Outcomes: The Student will be able to					Cognitive Level
CO1	Apply the fundamental concepts and techniques involved in text and speech processing for natural language and audio-based applications.				Ap
CO2	Analyze text processing techniques such as part-of-speech tagging, parsing, named entity recognition and language modeling.				An
CO3	Evaluate speech signal processing and speech recognition models based on their performance and effectiveness.				E
CO4	Design speech synthesis and speech-enabled applications using appropriate models, databases, and evaluation metrics.				C
CO5	Develop intelligent NLP and speech-based applications using modern language models and conversational systems.				C

UNIT I – INTRODUCTION TO TEXT AND SPEECH PROCESSING	9
NLP pipeline overview – Human speech production and acoustic phonetics – Text normalization: tokenization, stemming, lemmatization – Basics of speech signal: sampling, frequency, noise.	
UNIT II – TEXT PROCESSING TECHNIQUES	9
Part-of-speech tagging – Parsing techniques: CFG, Dependency Parsing – Named Entity Recognition (NER) – Chunking – Language modeling: N-gram models.	
UNIT III – SPEECH SIGNAL PROCESSING	9
Feature extraction: MFCC, PLP, LPC – Acoustic modeling: HMM, GMM – Basics of ASR (Automatic Speech Recognition) – Hidden Markov Models for speech.	
UNIT IV – SPEECH SYNTHESIS AND APPLICATIONS	9
Text-to-speech (TTS) systems – Speech databases and corpora – Evaluation metrics: WER, CER – Applications: Voice Assistants, Dictation Systems.	


 Chairman-Board of Studies
 Computer Science and Engineering
 Nandha College of Technology (Autonomous)
 Erode - 638 032

UNIT V – NLP AND SPEECH IN PRACTICE	9
Sentiment analysis – Chatbots and dialog systems – Multilingual and low-resource NLP – Ethics in speech and text technology – Models for NLP: RNNs and LSTM – The Transformer – Large Language Models – Masked Language Models.	
TOTAL PERIODS	45

REFERENCES:

1. Daniel Jurafsky and James H. Martin, "Speech and Language Processing", Pearson Education, 3rd Edition, 2025.
2. Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, "Introduction to Information Retrieval and Natural Language Processing", Cambridge University Press, 2024.
3. Lewis Tunstall, Leandro von Werra and Thomas Wolf, "Natural Language Processing with Transformers", O'Reilly Media, 2022.
4. Jacob Eisenstein, "Introduction to Natural Language Processing", MIT Press, 2019.

Mapping of COs with POs / PSOs					
COs	POs			PSOs	
	1	2	3	1	2
1	1	-	-	1	-
2	3	-	1	3	-
3	2	-	2	-	2
4	-	3	2	3	1
5	2	3	3	3	2
CO (W.A)	1.6	1.2	1.6	2.0	1.0


 Chairman-Board of Studies
 Computer Science and Engineering
 Nandha College of Technology (Autonomous)
 Erode - 638 052.

CP25C11 – ADVANCED DEEP LEARNING AND NEURAL NETWORKS		L	T	P	C
		3	0	0	3
Course Objective	<ul style="list-style-type: none"> To learn the fundamentals of neural networks and techniques for training deep models. To study advanced deep learning architectures and their applications in vision, NLP, and time series. 				
Course Outcomes: The Student will be able to					Cognitive Level
CO1	Explain the basics of neural networks and regularization techniques.				U
CO2	Build CNN and RNN models for image and sequence data.				Ap
CO3	Examine optimization and tuning methods for deep networks.				An
CO4	Construct generative and graph-based deep learning models.				Ap
CO5	Implement deep learning solutions for vision, NLP, and time series tasks.				Ap

UNIT I - FOUNDATIONS OF NEURAL NETWORKS	9
Neural Networks: Introduction – Perceptron – Multilayer Feed forward networks – Back propagation – Activation functions – Loss Function – Regularization: Data Augmentation – Noise Robustness – Early Stopping – Bagging – Dropout – Batch Normalization.	
UNIT II - CONVOLUTIONAL AND RECURRENT NEURAL NETWORKS	9
CNN and RNN: Convolutional Neural Networks – Convolution Operation – Architecture Overview – Input layers – Convolutional layers – Pooling layers – Fully connected layers. Recurrent Neural networks – LSTM – Bidirectional RNNs – RNN Language model – Word Level RNN – Deep Recurrent Networks – Recursive Neural Networks.	
UNIT III - OPTIMIZATION AND DEEP NETWORK TUNING	9
Optimization and Deep Networks: Optimization in deep learning – Non-convex optimization for deep networks – Stochastic Optimization – Generalization in neural networks – Spatial Transformer Networks. Tuning Deep Networks: Basic Concepts – Matching input data and network architecture – Relating model goal and output layers – Working layer count, parameter count and Memory – Weight initialization strategies – Using activation functions – Applying loss function – Dealing with overfitting.	
UNIT IV - ENCODER-DECODER AND GENERATIVE MODELS	9
Encoder / Decoder Models and Regressive Models: Encoder Decoder Models – Attention Mechanism – Attention over images – Hierarchical Attention – Variational Auto Encoders – Autoregressive models – NADE – MADE – PixelRNN – Generative Adversarial Networks (GANs) – Graph Convolution Network – Deep Belief Network.	

UNIT V - ADVANCED APPLICATIONS OF NEURAL NETWORKS AND DEEP LEARNING	9
Advanced Applications of Neural Networks: Object Detection: RCNN, Faster RCNN, and YOLO – MobileNet – DarkNet – Object Tracking – Audio WaveNet – Natural Language Processing: Word2Vec – Joint Detection – Face Recognition – Scene Captioning – Language Transformer Models. Advanced Applications of Deep Learning: Deep learning for Computer Vision – Deep learning for Time Series – Deep learning for Text – Generative Deep Learning.	
TOTAL PERIODS	45

REFERENCES:

1. François Chollet, "Deep Learning with Python", Manning Publications Co, Second Edition, 2021.
2. Josh Patterson, Adam Gibson, "Deep Learning: A Practitioner's Approach", O'Reilly Media, Inc, First Edition, 2017.
3. Bengio Yoshua, Ian J. Goodfellow, Aaron Courville, "Deep Learning", MIT Press, First Edition, 2016.
4. Dr. Adrian Rosebrock, "Deep Learning for Computer Vision with Python: Starter Bundle", PyImageSearch, First Edition, 2017.
5. Michael Nielsen, "Neural Networks and Deep Learning", Determination Press, First Edition, 2015.

Mapping of COs with POs / PSOs					
COs	POs			PSOs	
	1	2	3	1	2
1	3	1	1	2	1
2	3	3	2	3	2
3	2	3	3	2	3
4	2	3	2	3	2
5	2	3	3	3	3
CO (W.A)	2.4	2.6	2.2	2.6	2.2

CP25C12 - QUANTUM CRYPTOGRAPHY		L	T	P	C
		3	0	0	3
Course Objective	<ul style="list-style-type: none"> To introduce the principles of quantum information and computation along with key quantum algorithms and their applications. To understand post-quantum and advanced quantum cryptographic techniques for secure communication in the quantum era. 				
Course Outcomes: The Student will be able to					Cognitive Level
CO1	Apply the principles of quantum information and quantum computation to model quantum systems.				Ap
CO2	Analyze and implement quantum algorithms such as Deutsch-Jozsa, Grover's, and Shor's algorithms.				An
CO3	Evaluate quantum cryptanalysis techniques and their impact on classical cryptographic systems.				Ev
CO4	Apply post-quantum cryptographic techniques such as lattice-based methods for secure communication.				Ap
CO5	Design and simulate advanced quantum cryptographic protocols including quantum key distribution.				C

UNIT I – QUANTUM INFORMATION AND COMPUTATION	9
Quantum Information: Qubit – Single and multiple qubits – Mathematical model for quantum mechanics – Quantum measurements – Quantum computation and no-cloning theorem. Quantum operations: Phase shift – Bit flips – Hadamard transform – Arbitrary transforms – Entanglement. Quantum gates and circuits: CNOT gate – CCNOT gate – Reversible gates – Universal quantum gates – Quantum circuits – Quantum parallelism.	
UNIT II – QUANTUM ALGORITHMS	9
Deutsch's algorithm – Phase kickback – Deutsch-Jozsa algorithm – Simon's algorithm (analysis). Quantum Fourier Transform – Grover's search algorithm – Shor's factoring algorithm.	
UNIT III – QUANTUM CRYPTANALYSIS AND CLASSICAL CRYPTO SYSTEMS	9
Quantum cryptanalysis: Order finding – Factoring – Discrete logarithms – Hidden subgroup problem (HSP). Key exchange: Diffie-Hellman – Computational DH – Decisional DH. Security models: IND-CPA – Applications in RSA and ElGamal encryption.	
UNIT IV – POST-QUANTUM CRYPTOGRAPHY	9
Introduction to post-quantum cryptography – Lattice-based cryptography – Code-based cryptography – Isogeny-based cryptography. Lattice problems: Learning With Errors (LWE), Short Integer Solution (SIS). Public key encryption from LWE – Fully Homomorphic Encryption (FHE).	

UNIT V – ADVANCED QUANTUM CRYPTOGRAPHY	9
Quantum key distribution (QKD) – Bit commitment – Random oracles – Quantum one-time pad – Quantum public key encryption – Quantum FHE – Quantum indistinguishability – Quantum money. Advanced QKD: Imperfect devices – Beyond point-to-point QKD – Device-independent quantum cryptography.	
TOTAL PERIODS	45

REFERENCES:

1. M. A. Nielsen and I. Chuang, "Quantum Computation and Quantum Information", Cambridge University Press, 2012.
2. Riley Tipton Perry, "Quantum Computing from the Ground Up", World Scientific Publishing Ltd., 2012.
3. Richard J. Lipton Kenneth W. Regan, "Quantum algorithms via linear Algebra Primer", The MIT Press Cambridge, 2014.
4. Jack D. Hidary, "Quantum Computing: An Applied Approach", 1st Edition, Springer, 2019.

E-RESOURCES:

1. Quantum Algorithms and Cryptography (Video) – NPTEL
2. Introduction to Quantum Computing: Quantum Algorithms and Qiskit (Video) – NPTEL
3. Practical Quantum Computing with IBM Qiskit for Beginners (Video) – Coursera

Mapping of COs with POs / PSOs					
COs	POs			PSOs	
	1	2	3	1	2
1	3	2	1	2	1
2	2	2	3	3	1
3	2	2	3	3	2
4	2	2	3	3	2
5	2	3	3	3	3
CO (W.A)	2.3	2.2	2.6	2.8	1.8

CP25C13 – QUANTUM MACHINE LEARNING		L	T	P	C
		3	0	0	3
Course Objective	<ul style="list-style-type: none"> To learn the fundamentals of quantum computing, quantum information theory, and quantum algorithms for machine learning. To study quantum machine learning applications, programming tools, and advanced quantum machine learning techniques. 				
Course Outcomes: The Student will be able to					Cognitive Level
CO1	Explain the basics of quantum computing and quantum machine learning.				U
CO2	Apply quantum information theory and quantum algorithms for processing tasks.				Ap
CO3	Build quantum machine learning models and apply quantum optimization algorithms.				Ap
CO4	Examine quantum machine learning applications and programming tools.				An
CO5	Implement advanced quantum machine learning techniques for real-world tasks.				Ap

UNIT I - INTRODUCTION TO QUANTUM COMPUTING AND QUANTUM MACHINE LEARNING	9
Basics of Quantum Computing: Qubits, Superposition, and Entanglement – Quantum Gates, Circuits, and Measurement – Quantum Algorithms: Grover's and Shor's Algorithm – Introduction to Quantum Machine Learning (QML) – Classical vs Quantum ML – Applications of Quantum ML.	
UNIT II - QUANTUM INFORMATION THEORY AND QUANTUM ALGORITHMS	9
Quantum Entropy and Quantum Information – Quantum Teleportation and Superdense Coding – No-Cloning Theorem and Quantum Cryptography – Quantum Fourier Transform (QFT) – Quantum Phase Estimation (QPE).	
UNIT III - QUANTUM ALGORITHMS AND OPTIMIZATION FOR MACHINE LEARNING	9
Quantum Algorithms for Machine Learning: Quantum Support Vector Machines (QSVM) – Quantum Principal Component Analysis (QPCA) – Quantum k-Nearest Neighbors (QkNN) – Quantum Neural Networks (QNN). Quantum Optimization and Hybrid Quantum-Classical Models: Variational Quantum Eigensolver (VQE) – Quantum Approximate Optimization Algorithm (QAOA) – Hybrid Quantum-Classical Algorithms – Applications of Quantum Optimization in Machine Learning.	
UNIT IV - APPLICATIONS, TOOLS AND FUTURE OF QUANTUM MACHINE LEARNING	9
Applications of Quantum Machine Learning in Finance, Healthcare, NLP, and Drug Discovery – Quantum Programming Tools: Qiskit, TensorFlow Quantum – The Future of Quantum ML: Challenges and Opportunities.	

UNIT V - ADVANCED QUANTUM MACHINE LEARNING	9
Advanced Quantum Machine Learning – I: Quantum Mechanics and Data-Driven Physics – Kernelizing Quantum Mechanics – Qubit Maps – One-Qubit Transverse-Field Ising Model and Variational Quantum Algorithms. Advanced Quantum Machine Learning – II: Two-Qubit Transverse-Field Ising Model and Entanglement – Variational Algorithms, Quantum Approximate Optimization Algorithm, and Neural Network Quantum States with Two Qubits – Quantum Reservoir Computing.	
TOTAL PERIODS	45

REFERENCES:

1. Claudio Conti, "Quantum Machine Learning: Thinking and Exploration in Neural Network Models for Quantum Science and Quantum Computing", Springer Nature, First Edition, 2024.
2. Michael A. Nielsen, Isaac L. Chuang, "Quantum Computation and Quantum Information", Cambridge University Press, Second Edition, 2010.
3. Peter Wittek, "Quantum Machine Learning: What Quantum Computing Means to Data Mining", Springer, First Edition, 2014.
4. Chris Bernhardt, "Quantum Computing for Everyone", MIT Press, First Edition, 2019.
5. Eleanor G. Rieffel, Wolfgang H. Polak, "Quantum Computing: A Gentle Introduction", MIT Press, First Edition, 2014.
6. Nathan Wiebe, Ashish Kapoor, Krysta Svore, "Quantum Machine Learning", Springer, First Edition, 2021.
7. Brendon S. H. Tan, "Quantum Computing and Quantum Information Science", CRC Press, First Edition, 2021.

Mapping of COs with POs / PSOs					
COs	POs			PSOs	
	1	2	3	1	2
1	3	1	1	2	1
2	3	3	2	3	2
3	3	3	2	3	2
4	2	3	3	2	3
5	2	3	3	3	3
CO (W.A)	2.6	2.6	2.2	2.6	2.2

CP25C14 – AI IN IOT		L	T	P	C
		3	0	0	3
Course Objective	<ul style="list-style-type: none"> To learn the fundamentals of AI and IoT, data acquisition, and AI models for IoT data processing and decision-making. To study edge AI deployment, security, industry trends, and future directions of AI-enabled IoT systems. 				
Course Outcomes: The Student will be able to					Cognitive Level
CO1	Explain the basics of AI and IoT and their integration in intelligent systems.				U
CO2	Apply data acquisition and pre-processing techniques for AI-enabled IoT data.				Ap
CO3	Build AI models for IoT data processing and decision-making.				Ap
CO4	Examine edge AI deployment and security strategies for IoT systems.				An
CO5	Implement AI in IoT solutions for industry applications and future trends.				Ap

UNIT I - INTRODUCTION TO AI IN IoT	9
Introduction to AI in IoT: Fundamentals of AI and IoT – IoT Architecture and AI Integration – AI Models for IoT (Machine Learning, Deep Learning) – AI-Driven IoT Applications (Smart Cities, Healthcare, Industrial IoT) – Ethical Considerations in AI-Enabled IoT – Case Study: Smart Agriculture with AI-Enabled IoT Sensors.	
UNIT II - DATA ACQUISITION AND PRE-PROCESSING IN AI-ENABLED IoT	9
Data Acquisition & Pre-processing in AI-Enabled IoT: Sensor Data Collection & Processing – AI-Based Data Normalization, Filtering & Feature Engineering – Challenges in Data Handling & Storage for AI Models – Ethical Considerations in AI-Driven Data Processing – Case Study: Predictive Maintenance in Smart Factories.	
UNIT III - AI MODELS FOR IoT DATA PROCESSING AND DECISION-MAKING	9
AI Models for IoT Data Processing & Decision-Making: Supervised & Unsupervised AI Models in IoT – Classification, Clustering & Anomaly Detection – Deep Learning Models for IoT: CNNs, RNNs – Performance Metrics & Model Optimization – Case Study: AI-Driven Traffic Monitoring System.	
UNIT IV - EDGE AI AND SECURITY FOR IoT SYSTEMS	9
Edge AI & Security for IoT Systems: Edge AI Deployment Strategies – Edge AI for Real-Time Decision-Making – Deploying AI Models in Resource-Constrained IoT Devices – AI-Powered Security & Intrusion Detection in IoT Networks – Ethical Implications & Privacy Concerns – Case Study: AI-Based Intrusion Detection in Healthcare IoT Networks.	

UNIT V - AI IN IoT: INDUSTRY TRENDS, ETHICS AND FUTURE	9
AI in IoT – Industry Trends: Industry Applications of AI in IoT – Smart Cities, Healthcare, & Industrial IoT – Scaling AI for Large-Scale IoT Deployments – Sustainability & Scalability Challenges in AI-Enabled IoT. AI in IoT – Ethics and Future Trends: Monetization Strategies & AI Business Models in IoT – Green AI and IoT Regulations and Standardization – Edge AI advancements for real-time IoT decision-making – Quantum AI & IoT interoperability for future applications – Case Study: AI-Driven Smart Cities for Sustainable Urban Development.	
TOTAL PERIODS	45

REFERENCES:

1. Fadi Al-Turjman, "Artificial Intelligence in IoT", Springer, First Edition, 2019.
2. Ambika Nagaraj, "The Role of AI in Enhancing IoT-Cloud Applications", Bentham Books, First Edition, 2023.
3. Rajkumar Buyya, Amir Vahid Dastjerdi, "Internet of Things: Principles and Paradigms", Elsevier, First Edition, 2016.
4. Jason Ioannou, "Data Analytics for IoT and Smart Systems", Springer, First Edition, 2020.
5. Ian Goodfellow, Yoshua Bengio, Aaron Courville, "Deep Learning", MIT Press, First Edition, 2016.
6. Weisong Shi, "Edge Computing: Vision and Challenges", Springer, First Edition, 2020.
7. Mauro Conti, "Security and Privacy in the Age of IoT", Springer, First Edition, 2020.
8. François Chollet, "Deep Learning with Python", Manning Publications, Second Edition, 2021.
9. Stuart Russell, Peter Norvig, "Artificial Intelligence: A Modern Approach", Pearson, Fourth Edition, 2020.
10. Arshdeep Bahga, Vijay Madisetti, "Internet of Things: A Hands-On Approach", Universities Press, Second Edition, 2016.

Mapping of COs with POs / PSOs					
COs	POs			PSOs	
	1	2	3	1	2
1	3	1	1	2	1
2	3	3	2	3	2
3	3	3	2	3	2
4	2	3	3	2	3
5	2	3	3	3	3
CO (W.A)	2.6	2.6	2.2	2.6	2.2

CP25C15 – WEB 3.0		L	T	P	C
		3	0	0	3
Course Objective	<ul style="list-style-type: none"> To understand Web 3.0 concepts, blockchain, smart contracts, dApps, NFTs, and digital identity. To develop secure and interoperable decentralized applications using AI, IoT, and Semantic Web technologies. 				
Course Outcomes: The Student will be able to					Cognitive Level
CO1	Describe the evolution, principles, and architecture of Web 3.0.				U
CO2	Apply blockchain, consensus mechanisms, and distributed ledger technologies.				Ap
CO3	Develop smart contracts, dApps, and NFT-based solutions using open standards.				Ap
CO4	Implement decentralized identity and secure peer-to-peer networking using W3C standards.				Ap
CO5	Evaluate the integration of AI, IoT, and Semantic Web technologies in decentralized applications and governance.				E

UNIT I - INTRODUCTION	9
Evolution of the Internet – Transition from Web 1.0 (static/read-only) to Web 2.0 (interactive/centralized) and on to Web 3.0 (decentralized) – Core Concepts – Principles – Decentralisation – Semantic Interoperability - Exploration of Web 3.0's layered structure: Blockchain, Decentralized storage, P2P networks, and Semantic Web protocols.	
UNIT II - BLOCKCHAIN TECHNOLOGY & DISTRIBUTED LEDGER SYSTEMS	9
Blockchain Architecture – Consensus Mechanism – Proof-of-Work (PoW), Proof-of-Stake (PoS), and hybrid models – Pros and Cons - Directed Acyclic Graphs (DAGs) and emphasis on RESTful API design following open standards.	
UNIT III - SMART CONTRACTS, DIGITAL ASSET TOKENIZATION (NFTS)	9
Fundamentals of design, deployment, and management of smart contracts on Blockchain platforms - Decentralized Applications (dApps) - Integration of Blockchain back - ends with front-end systems - HTML5, semantic HTML, and ARIA/WCAG. Digital Asset Tokenization & NFTs - ERC-721/1155 standards - Metadata structuring with JSON-LD for interoperability - Oracle Services – Dynamic and real-time data - Decentralized Storage Systems – Storage Protocols – IPFS – Filecoin.	
UNIT IV - DIGITAL IDENTITY AND PEER-TO-PEER (P2P) NETWORKING	9
Security – Scalability – Interoperability with traditional systems – Digital Identity – Decentralized Identity (DIDs) self-sovereign identity principles, and W3C DID and verifiable credentials standards.	

UNIT V - DECENTRALIZED GOVERNANCE & TECH INTEGRATION	9
Decentralized Governance & DAOs – Token – based governance – Structure of DAOs – voting mechanism - Regulatory & Economic Frameworks – Technology integration - Semantic Web protocols (RDF, OWL, SPARQL), AI for data curation, and IoT for real-time data collection - open standard compliance.	
TOTAL PERIODS	45

REFERENCES:

1. Prabhat Kumar Srivastav, Prateek Singhal, Basudeo Singh, Nitin Sharma, "Web 3.0 | The Next Generation's Internet and Understanding the Concept", CRC Press, 2024.
2. Imran Bashir, "Mastering Blockchain: Unlocking the World of Cryptocurrencies", Packt Publishing, 2023.
3. Andreas M. Antonopoulos and Gavin Wood, "Mastering Ethereum: Building Smart Contracts and DApps", 2nd Edition, O'Reilly Media, 2018.
4. Alex Tapscott, "Web3: Charting the Internet's Next Economic and Cultural Frontier, Portfolio", 2023.

Mapping of COs with POs / PSOs					
COs	POs			PSOs	
	1	2	3	1	2
1	3	1	1	2	1
2	3	3	2	3	2
3	3	2	2	3	2
4	2	3	3	2	3
5	2	2	3	3	3
CO (W.A)	2.6	2.0	2.2	2.6	2.2


 Chairman-Board of Studies
 Computer Science and Engineering
 Nandha College of Technology (Autonomous)
 Erode - 638 052.

CP25C16 - ADVANCED LARGE LANGUAGE MODELS		L	T	P	C
		3	0	0	3
Course Objective	<ul style="list-style-type: none"> To understand the fundamentals, capabilities, and limitations of Large Language Models (LLMs). To apply LLMs for content creation, code generation, sentiment analysis, and other real-world applications. 				
Course Outcomes: The Student will be able to					Cognitive Level
CO1	Understand the fundamental concepts, architecture, and components of Large Language Models.				U
CO2	Describe pretraining methods, language modeling techniques, and decoding strategies used in LLMs.				U
CO3	Apply parameter-efficient tuning methods and fine-tuning approaches for LLM adaptation.				Ap
CO4	Analyze prompt engineering techniques and in-context learning approaches for various applications.				An
CO5	Analyze evaluation methods and metrics used to assess the performance of Large Language Models.				An

UNIT I – BUILDING BLOCKS OF MODERN LLM'S	9
Large language model basics, Large Language model architecture, Transformer Architecture, Attention all you need – Encoder – Decoder. Pre-training and Language Modeling, Auto-Regressive Language Modelling and Auto-Encoder Language Modelling, Early experiments with decoder and auto-encoder pretraining -Transformer Decoding Pretraining by Denoising Task.	
UNIT II – LLM PRETRAINING METHODS	9
Masked Language Modeling, BERT Pretraining with Masked LM, Autoregressive LM and Masked LM, LLM pretraining data, preprocessing clean text, Scaling Up pretraining data using the web, Language Model (LM) - Decoding Strategies.	
UNIT III – PARAMETER EFFICIENT TUNING METHODS	9
Categories of PETM – Prefix tuning, Prompt tuning- Adapters, Compacters- Layer freezing-Bias terms, Fine-tuning- diff pruning, Low Rank Adaptation (LoRA) – Advantages and Pitfalls, Explainability and LLMs, ethical considerations.	
UNIT IV – IN-CONTEXT LEARNING, PROMPT ENGINEERING AND ALIGNMENT	9
In-Context Learning, Finetuning - Zero-shot learning- Few-shot learning, Basics of prompting – Instruction prompting, Chain of thought prompting –Prompt selection, Automatic prompt design, Case Study: Visual Questions and Answering Systems, Sentiment training with Multi language dataset – CLIP.	

UNIT V – GENERATION BASED AUTOMATIC EVALUATION METHODS	9
What is Evaluation – Human Evaluation- Intrinsic Vs Extrinsic Evaluation, Why automatic evaluation- Ranking, Multiple metrics, GLUE-GEM- Beyond metrics, Human Evaluation metric – Quantitative and Qualitative evaluation, Human annotation- Reporting- Challenges in Evaluation, Evaluation metrics.	
TOTAL PERIODS	45

REFERENCES:

1. Sebastian Raschka, "Build a Large Language Model (From Scratch)", September 2024, Manning Publications.
2. Tanmoy Chakraborty, "Introduction to Large Language Models", Wiley Publications.
3. Jay Alammar, Maarten Grootendorst, "Hands on Large Language Models, Language Understanding and Generation", Shroff Publishers.
4. Amaratunga, Thimira, "Understanding Large Language Models: Learning Their Underlying Concepts and Technologies", Springer Verlag, 2023.
5. Raj Arun R, "Mastering Large Language Models with Python", AVA publications 2024.

Mapping of COs with POs / PSOs					
COs	POs			PSOs	
	1	2	3	1	2
1	1	-	-	1	-
2	3	-	1	3	-
3	2	-	2	-	2
4	-	3	2	3	1
5	2	3	3	3	2
CO (W.A)	1.6	1.2	1.6	2.0	1.0

CP25C17 - EDGE AND FOG COMPUTING		L	T	P	C
		3	0	0	3
Course Objective	<ul style="list-style-type: none"> To provide students with a comprehensive understanding of designing computing systems that process data closer to where it's created, rather than sending everything to distant cloud servers. To build smart applications that work across different computing layers - from powerful cloud servers to small edge devices like sensors and smart phones. 				
Course Outcomes: The Student will be able to					Cognitive Level
CO1	Explain the evolution of cloud-to-edge paradigms and describe the role of edge and fog computing in real-time IoT applications				U
CO2	Design and optimize network slicing strategies for integrated cloud-fog-edge systems using mathematical modeling and optimization techniques.				An
CO3	Develop middleware architectures and container-based solutions for distributed edge computing environments.				Ap
CO4	Implement intelligent data management systems and machine learning solutions at the fog layer for real-time analytics				Ap
CO5	Evaluate emerging technologies and research directions in edge and fog computing for future system design.				An

UNIT I - CLOUD TO EDGE COMPUTING PARADIGMS	9
Evolution from Cloud to Fog and Edge Computing - IoT Architecture Requirements and Latency Limitations - Fog vs Edge vs Cloud: Differences in Computation and Storage - Multi-access Edge Computing (MEC) Concepts - Multi-tier Architectural Models for Distributed Systems	
UNIT II - EDGE OFFLOADING AND SYSTEM OPTIMIZATION	9
Edge Offloading Strategies and Placement Issues - Challenges in Latency, Energy Efficiency, QoS, and Privacy - Queuing Theory and System Performance Metrics - Optimization Techniques using Machine Learning for Slice Orchestration Cloud-to-Fog-to-Things (C2F2T) Integration - Network Slicing for 5G/6G: Slice Types and Lifecycle - Resource Virtualization: SDN, NFV, and Container-Based Federation - Modeling and Simulation for Resource Management	
UNIT III - MIDDLEWARE AND ANALYTICS AT THE EDGE	9
Middleware Design Goals: Heterogeneity, Discovery, Scalability – Service Orchestration at the Edge - Lightweight Containers: Docker and Kubernetes on Constrained Devices - Edge Cluster Formation and Node Coordination - Communication Protocols: MQTT, CoAP, and Microservices - Distributed Storage, Synchronization, and Fault Recovery Data Management at the Fog Layer: Caching, Placement, Lifecycle - Federated Learning and Distributed ML for Edge Analytics - Stream Processing and Real-Time Analytics Frameworks	


Chairman-Board of Studies

UNIT IV - ADVANCED TECHNOLOGIES FOR EDGE SYSTEMS	9
Block chain for Fog Resource -Software-defined Edge Networks and - Green and Sustainable Edge Computing Security and Privacy Challenges in Edge Environments.	
UNIT V - EMERGING TRENDS FOR EDGE COMPUTING	9
Edge AI and TinyML: Neural Compression, Quantization, Model Deployment - 6G Vision: Ultra-Low Latency, Terahertz Communications - Advanced Paradigms: Digital Twins, Serverless Edge, Cognitive Networks.	
TOTAL PERIODS	45

REFERENCES:

1. Rajiv Pandey, Sunil Kumar Khatri, Neeraj Kumar Singh, Parul Verma, "Artificial Intelligence and Machine Learning for EDGE Computing", Academic Press, 1st Edition, 2022.
2. S. Goundar, Ed., "Edge Computing - Technology, Management and Integration", IntechOpen, 2023.
3. Ahmed Banafa, "Secure And Smart Internet of Things (IoT) using Blockchain and Artificial Intelligence (AI)", River publications, 2018.
4. Javid Taheri, Schahram Dustdar, Albert Zomaya, Shuiguang Deng, "Edge Intelligence From Theory to Practice", Springer, 2023.
5. Deepak Gupta, Aditya Khamparia, "Fog, Edge, and Pervasive Computing in Intelligent IoT Driven Applications", Wiley-IEEE Press, 2021.

Mapping of COs with POs / PSOs					
COs	POs			PSOs	
	1	2	3	1	2
1	-	3	3	-	2
2	3	-	2	2	2
3	3	-	2	2	-
4	2	-	3	2	3
5	2	2	3	-	3
CO (W.A)	2.0	1.0	2.6	1.2	2.0

CP25C18 – GREEN COMPUTING AND SUSTAINABILITY		L	T	P	C
		3	0	0	3
Course Objective	<ul style="list-style-type: none"> To understand the principles of green computing, environmental impacts of computing technologies, and the need for sustainable computing practices. To explore sustainable IT practices and energy-efficient hardware systems for achieving sustainability and green growth. 				
Course Outcomes: The Student will be able to					Cognitive Level
CO1	Explain the concepts, environmental impact, frameworks, and energy efficiency techniques involved in Green Computing and sustainable IT practices.				U
CO2	Apply green hardware design, manufacturing, recycling, and energy-saving software techniques for sustainable computing systems.				A
CO3	Analyze sustainable software development methodologies, energy-aware software engineering practices, and green algorithms using appropriate metrics.				An
CO4	Evaluate energy-efficient infrastructures, storage systems, server power management, and green data center technologies for sustainable computing environments.				E
CO5	Examine Green IT strategies, standards, laws, sustainability reporting, and auditing practices for implementing eco-friendly IT initiatives.				An

UNIT I - INTRODUCTION TO GREEN COMPUTING	9
Introduction to Green Computing: Scope - Environmental Impact of Computing - Need for Sustainable Computing – Power-Aware Resource Management - Green IT - OCED Green IT Framework – Greening IT - Benefits and Challenges - Quantifying IT Energy Efficiency.	
UNIT II - GREEN HARDWARE	9
Life cycle of hardware - Reuse, Recycle and Dispose Green Design and Manufacturing of Computers- Green Software - Energy-Saving Software Techniques.	
UNIT III - SUSTAINABLE SOFTWARE DEVELOPMENT	9
Attributes – metrics – Methodology - Energyaware Software Engineering - Green Algorithms Activities: Case study: Project's Carbon Footprint	
UNIT IV -GREEN DATA CENTER AND STORAGE SYSTEMS	9
Infrastructure for Energy Efficiency – server power management - Practices for Improving the Energy Efficiency of Data Storage - Optical Interconnects for Green Computers and Data Centers Case study: Energy-efficient cloud computing	

UNIT V - MANAGING GREEN IT AND NONREGULATORY GOVERNMENT INITIATIVES	9
Strategizing Green Initiatives – Implementation - Green IT Laws, Standards and Protocols- Regulatory Environment and IT Manufacturers Green Building Standards - Sustainability Reporting and Green Audits Activities: Demo: Green IT Laws in India. Debate: Role of IT professional in Green Computing. And Perform Energy audit in computer lab using software tools.	
TOTAL PERIODS	45

REFERENCES:

1. San Murugesan, Harnessing "Green IT: Principles and Practices", Wiley, 2012.
2. Suyel Namasudra(Editor), "Green and Sustainable Computing", Elsevier, 1st Edition, Volume 87, 2012.
3. Toby Velte, Anthony Velte, Robert Elsenpeter, "Green IT: Reduce Your Information System's Environmental Impact", McGraw Hill, 2008.
4. Bhuvan Unhelkar, "Green IT Strategies and Applications: Using Environmental Intelligence", CRC Press, 2011.
5. Balamurugan Balusamy, Naveen Chilamkurti, SeifedineKadry, "Green Computing in Smart Cities: Simulation and Techniques", Springer Singapore, 2020.
6. Mike Halsey, "The Green IT Guide - How to Make Your IT Systems and Business Sustainable and Carbon Neutral", Apress, 2025

E-RESOURCES:

1. Prof. Deepu Philip, Prof. Amandeep Singh | IIT Kanpur Link
http://onlinecourses.nptel.ac.in/noc21_mg85/preview
2. Prof. Sayak Banerjee | IIT Hyderabad Link "https://onlinecourses.nptel.ac.in/noc23_me138/preview"

Mapping of COs with POs / PSOs					
COs	POs			PSOs	
	1	2	3	1	2
1	2	1	1	2	1
2	3	2	2	3	1
3	2	1	3	2	2
4	2	2	3	2	3
5	3	2	3	3	3
CO (W.A)	2.4	1.6	2.4	2.4	2.0

CP25C19 - COGNITIVE COMPUTING		L	T	P	C
		3	0	0	3
Course Objective	<ul style="list-style-type: none"> To understand the core principles of cognitive computing and how they emulate human thought processes. To apply cognitive computing to solve complex, unstructured, and real-world problems. 				
Course Outcomes: The Student will be able to					Cognitive Level
CO1	Describe the fundamental concepts, architectures, and paradigms of cognitive computing and their role in intelligent decision-making systems.				U
CO2	Analyze cognitive computing models and learning mechanisms to understand perception, reasoning, and knowledge representation in intelligent systems.				An
CO3	Evaluate cognitive computing techniques, frameworks, and applications to assess their effectiveness in solving complex real-world problems.				Ap
CO4	Design cognitive computing solutions by selecting appropriate models, algorithms, and system architectures for human-centric intelligent applications.				Ap
CO5	Evaluate the various applications and emerging areas in cognitive computing				An

UNIT I - INTRODUCTION TO COGNITIVE COMPUTING	9
Human cognition - Introduction to Cognitive Computing – Elements of a Cognitive System - Design Principles for Cognitive Systems - Artificial Intelligence, Natural Language Processing and Machine Learning in Cognitive Computing.	
UNIT II - COGNITIVE ARCHITECTURES	9
Cognitive architectures: ACT-R, Soar, OpenCog - Knowledge Representation and Reasoning - Machine Learning models in cognitive systems - Human-like Learning: Supervised, Unsupervised, Reinforcement	
UNIT III - NLP IN A COGNITIVE SYSTEM	9
Role of NLP in a Cognitive System - NLP pipeline: POS tagging, NER, parsing - Sentiment Analysis and Question Answering Debate: Sentiment analysis and its challenges in Regional Languages. Semantic Web - Semantic Understanding: Ontologies and Knowledge Graphs	
UNIT IV - COGNITIVE AGENTS AND EXPERT SYSTEMS	9
Cognitive Agents - Rule-based Agents and Expert Systems - Decision Support Systems -Emotion-aware computing and human-computer interaction- Cognitive bias and ethics in decision making	


Chairman-Board of Studies

Computer Science and Engineering
Nandha College of Technology (Autonomous)
Erode - 638 052.

UNIT V - APPLICATION AND EMERGING TRENDS IN COGNITIVE APPLICATION	9
Process of Building a Cognitive Application -Implications of Cognitive Computing Cognitive Computing Applications: Healthcare, Finance, Education, Legal, RoboticsBuilding a Cognitive Healthcare Application - Emerging Cognitive Computing Areas - Cognitive cloud- Cognitive IoT - Cognitive Chatbots - Neuromorphic computing and brain-inspired models.	
TOTAL PERIODS	45

REFERENCES:

1. Velankar, Mahalle & Shinde, "Cognitive Computing for Machine Thinking", CRC Press, 2024.
2. Judith S. Hurwitz, Marcia Kaufman, Adrian Bowles, "Cognitive Computing and Big Data Analytics", Wiley, 2015.
3. Peter Fingar, "Cognitive Computing: A Brief Guide for Game Changers", Meghan Kiffer, 2015.
4. Building Cognitive Applications with IBM Watson Services, Redbooks, 2017
5. Steven Bird, Ewan Klein, Edward Loper, "Natural Language Processing with Python", O'Reilly, 2009.

Mapping of COs with POs / PSOs					
COs	POs			PSOs	
	1	2	3	1	2
1	-	-	-	-	-
2	3	-	-	3	1
3	-	-	2	-	2
4	-	1	-	3	-
5	2	2	3	-	3
CO (W.A)	2.5	1.5	2.5	1.8	1.4


 Chairman-Board of Studies
 Computer Science and Engineering
 Nandha College of Technology (Autonomous)
 Erode - 638 052.

CP25C20 – AGENTIC AI		L	T	P	C
		3	0	0	3
Course Objective	<ul style="list-style-type: none"> Understand the foundations, architectures, and ethical considerations of intelligent agent systems, including autonomous decision-making, coordination in multi-agent and human-AI environments, and real-world agentic AI applications. Design and develop intelligent agents using modern AI frameworks by applying rational decision-making and reinforcement learning techniques for adaptive and autonomous behavior. 				
Course Outcomes: The Student will be able to					Cognitive Level
CO1	Analyze the concepts, architectures, environments, and types of intelligent agents used in Agentic AI systems.	An			
CO2	Apply decision-making and planning techniques for solving problems under deterministic and uncertain environments.	Ap			
CO3	Evaluate reinforcement learning and deep reinforcement learning techniques for intelligent agent behavior and adaptation.	E			
CO4	Analyze multi-agent coordination, communication, negotiation, and human-AI collaborative systems using agent platforms.	An			
CO5	Design LLM-based and embodied intelligent agent applications while considering ethical, social, and real-world implications.	C			

UNIT I - FOUNDATIONS OF INTELLIGENT AGENTS	9
Introduction to Agentic AI: History, Motivation, Applications - Agents and Environments: Sensors, Actuators, Environment Types - Types of Agents: Simple Reflex, Goal-Based, Utility-Based, Learning Agents - Architectures: Reactive, Deliberative, Hybrid, Subsumption.	
UNIT II - AGENT DECISION-MAKING AND PLANNING	9
Rationality and Utility Theory - Task Environment Analysis - AI Planning Techniques: STRIPS, Classical Planning - Decision-Making under Uncertainty: MDPs, Bayesian Models, Game Theory.	
UNIT III - LEARNING IN AGENTIC SYSTEMS	9
Reinforcement Learning: Q-Learning, SARSA - Value and Policy Iteration - Deep RL: DQNs, Policy Gradient Methods, Actor-Critic - Tools: OpenAI Gym, PettingZoo, TensorFlow RL.	
UNIT IV -MULTI-AGENT SYSTEMS AND COLLABORATION	9
Multi-Agent Coordination: Cooperation, Competition - Agent Communication and Negotiation - Human-AI Collaboration: Interactive and Mixed-Initiative Systems - Platforms: NetLogo, RoboCup Simulation Activity: • Create Autonomous agents using NetLogo platform.	

UNIT V - LLM AGENTS AND EMBODIED COGNITION AND ETHICS AND APPLICATIONS OF AGENTIC AI	9
Language-based Agents: Tool Use, Planning, and Reasoning - Memory-Augmented and Situated Agents - Agents in Robotics and Smart Environments - Unity ML-Agents, Hugging Face Tools AI Alignment, Fairness, Explainability - Social and Environmental Implications - Case Studies: Autonomous Vehicles, Conversational Agents - Course Project Evaluation and Presentations.	
TOTAL PERIODS	45

REFERENCES:

1. Russell, Norvig, "Artificial Intelligence: A Modern Approach", Pearson, 4th Edition, 2021.
2. Shoham, Leyton-Brown, "Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations", CUP, 2009.
3. Sutton, Barto, "Reinforcement Learning: An Introduction", MIT Press, 2nd Edition, 2018.

Mapping of COs with POs / PSOs					
COs	POs			PSOs	
	1	2	3	1	2
1	2	1	3	2	1
2	3	1	3	3	2
3	3	1	3	3	3
4	2	2	3	3	2
5	3	2	3	3	3
CO (W.A)	1.8	2.4	2.6	2.4	2.8


 Chairman-Board of Studies
 Computer Science and Engineering
 Nandha College of Technology (Autonomous)
 Erode - 638 052.

CP25C21 - MIXED REALITY		L	T	P	C
		3	0	0	3
Course Objective	<ul style="list-style-type: none"> To define the concepts of Mixed Reality (MR), Virtual Reality (VR), and Augmented Reality (AR) and distinguish their features and applications. To understand holographic technologies and develop immersive 3D applications using Mixed Reality concepts for real-world problem-solving. 				
Course Outcomes: The Student will be able to					Cognitive Level
CO1	Understand the concepts and characteristics of Virtual Reality, Augmented Reality, and Mixed Reality.				U
CO2	Describe holograms, spatial computing, and the components of Mixed Reality systems.				U
CO3	Apply Mixed Reality development tools and techniques to create basic 3D environments.				Ap
CO4	Analyze the design principles and user interactions involved in Mixed Reality applications.				An
CO5	Analyze the use of Mixed Reality technologies in real-world applications across various domains.				An

UNIT I – DEFINING VIRTUAL AND AUGMENTED REALITY	9
Four Key Elements of Virtual Reality Experience Looking at Some Other Types of Virtual and Augmented Reality. Looking at the Available Form Factors, Focusing on Features. Room-scale versus stationary experience. Inside-out tracking. Tracking devices Haptic feedback, Audio, Considering Controllers.	
UNIT II – EXPLORING THE CURRENT STATE OF AUGMENTED REALITY	9
Mobile devices AR headsets. AR glasses. Hand tracking. Motion controllers Hand and Gesture Tracking, Whole Body Tracking.	
UNIT III – SIMPLE VR ENVIRONMENT DEVELOPMENT	9
Interacting with the Virtual World, Interactions: Manipulation, Navigation, and Communication, Creating Basic objects in Sketchup tool, Grouping and Components, Development of Virtual Environment using SketchUp tool, connecting the VR environment to Meta Quest 2 (HMD) using VR Sketch, Navigation in the Virtual Environment.	
UNIT IV – CREATING 2D/3DPROJECTS	9
Getting started with Unity, Installing Unity Hub and Unity Editor, Editor interface, Creating Projects with templates, Asset store, creating a simple 2d game using game objects, Scripting in c#, Visual Scripting, creating a simple 3D game by importing game objects from asset store, Building the projects to desktop OS/Mobile OS.	

UNIT V – SETTING UP FOR AR DEVELOPMENT	9
Installing XR plugins for AR devices, Installing the AR Foundation package, Building the Simple AR scene in project, Importing the Sample assets, Using AR Session and AR Session Origin, AR Camera, Adding AR Ray cast Manager, Building and running the scene. Placing an object on a plane, Writing the Place Object on Plane script, creating a prefab for placing on the scene.	
TOTAL PERIODS	45

REFERENCES:

1. Jonathan Lenôwes, “Augmented Reality with Unity AR Foundation”, Packt Publishing, 2021.
2. William R. Sherman, Alan B. Craig, “Understanding-Virtual-Reality-InterfaceApplication-and-Design”, The Morgan Kaufmann Series in Computer Graphics.
3. Paul Mealy, “Virtual &Augmented Reality for Dummies”, Wiley Publications, 2013.
4. Chen, “Virtual, Augmented and Mixed Reality - Applications and Case Studies”, Springer Publications 2019.
5. Rajiv Chopra, “Virtual and Augmented Reality”, Khanna Publishing House, 2021.

Mapping of COs with POs / PSOs					
COs	POs			PSOs	
	1	2	3	1	2
1	1	-	-	1	-
2	3	-	1	3	-
3	2	-	2	-	2
4	-	3	2	3	1
5	2	3	3	3	2
CO (W.A)	1.6	1.2	1.6	2.0	1.0

CP25C22 - BLOCKCHAINS ARCHITECTURE AND DESIGN		L	T	P	C
		3	0	0	3
Course Objective	<ul style="list-style-type: none"> To apply the fundamental concepts, architectures, and cryptographic principles of blockchain technology, including distributed ledgers, consensus mechanisms, smart contracts and blockchain platforms. To design and develop blockchain-based solutions for decentralized applications by selecting appropriate protocols, platforms and security mechanisms while addressing scalability and privacy challenges. 				
Course Outcomes: The Student will be able to					Cognitive Level
CO1	Apply the fundamental concepts, components and operational principles of blockchain architectures and distributed ledger technologies.				Ap
CO2	Analyze blockchain consensus mechanisms, data structures and network architectures to understand security, performance and scalability issues.				An
CO3	Evaluate blockchain platforms, smart contract frameworks and architectural designs for decentralized applications.				E
CO4	Design blockchain-based systems using appropriate architectures, consensus protocols and smart contract models.				C
CO5	Develop blockchain solutions for real-world applications by addressing scalability, privacy and emerging technological challenges.				C

UNIT I – INTRODUCTION TO BLOCKCHAIN	9
Introduction to Blockchain – Layers of Blockchain – Types of Blockchain – Generic Elements of Blockchain – Basic Cryptographic Primitives used in Blockchain – Hash Functions – Merkle Tree – Structure of Block – Linking Blocks in the Blockchain – Mining the Block – Validating a New Block – Assembling and Selecting Chains of Blocks.	
UNIT II – BITCOIN AND CONSENSUS MECHANISMS	9
Bitcoin Overview – Bitcoin Scripts – Bitcoin Transactions – Bitcoin Network and Payments – Consensus Mechanism – Types of Consensus Mechanism – Proof of Work (PoW) – Consensus Protocols for Permissionless Blockchain Environment – Consensus Protocols for Permissioned Blockchain Environment.	
UNIT III – SMART CONTRACTS AND ETHEREUM	9
Smart Contract – Definition – Ricardian Contracts – Introduction to Ethereum – Ethereum Network – Components of Ethereum – Ethereum Virtual Machine (EVM) – Ethereum Development Environment – Solidity Language.	
UNIT IV – WEB3 AND HYPERLEDGER FABRIC	9
Exploring Web3 with Geth – Contract Deployment – Interacting with Contracts via Frontends – Truffle – Introduction to Hyperledger – Reference Architecture – Hyperledger Fabric – Hyperledger Sawtooth – Corda.	

UNIT V - BLOCKCHAIN APPLICATIONS, CHALLENGES AND FUTURE DIRECTIONS	9
Blockchain for Government: Border Control – Voting – Citizen Identification – Blockchain for Health Industry – Blockchain for Finance: Insurance – Post Trade Settlement – Financial Crime Prevention – Scalability – Privacy – Emerging Trends – Limitations of Current Blockchain Platforms – Open Research Challenges.	
TOTAL PERIODS	45

REFERENCES:
<ol style="list-style-type: none"> 1. Imran Bashir, "Mastering Blockchain: A Deep Dive into Distributed Ledgers, Consensus Protocols, Smart Contracts, DApps, Cryptocurrencies, Ethereum and More", Packt Publishing, 4th Edition, 2023. 2. Andreas M. Antonopoulos and David A. Harding, "Mastering Bitcoin: Programming the Open Blockchain", O'Reilly Media, 3rd Edition, 2023. 3. Ahmed Banafa, "Blockchain Technology and Applications", River Publishers, 2024. 4. Melanie Swan, "Blockchain: Blueprint for a New Economy", O'Reilly Media, 2015.

Mapping of COs with POs / PSOs					
COs	POs			PSOs	
	1	2	3	1	2
1	1	-	-	1	-
2	3	-	1	3	-
3	2	-	2	-	2
4	-	3	2	3	1
5	2	3	3	3	2
CO (W.A)	1.6	1.2	1.6	2.0	1.0


Chairman-Board of Studies
Computer Science and Engineering
Nandha College of Technology (Autonomous)
Erode - 638 052.

CP25C23 - HUMAN - CENTERED AI		L	T	P	C
		3	0	0	3
Course Objective	<ul style="list-style-type: none"> To apply the principles, frameworks, and methodologies of Human-Centered Artificial Intelligence for developing responsible, transparent and user-focused intelligent systems. To design and evaluate trustworthy AI solutions by incorporating fairness, privacy, reliability, explainability, governance and ethical considerations in real-world applications. 				
Course Outcomes: The Student will be able to					Cognitive Level
CO1	Apply the fundamental concepts, principles and design philosophies of Human-Centered AI for developing responsible and user-focused intelligent systems.				Ap
CO2	Analyze human-AI interaction models, human factors and ethical considerations influencing intelligent systems.				An
CO3	Evaluate fairness, bias mitigation techniques, privacy-preserving methods and HCAI frameworks for trustworthy AI deployment.				E
CO4	Design human-centered AI solutions using appropriate interaction techniques, ethical guidelines and evaluation strategies.				C
CO5	Develop AI systems that incorporate governance structures, safety practices, explainability and human capability enhancement principles.				C

UNIT I – INTRODUCTION TO HUMAN-CENTERED AI	9
Overview of AI Development – Defining Human-Centered AI (HCAI) – Difference between Traditional AI and HCAI – Process of Design Thinking – Cognitive Models.	
UNIT II – HUMAN FACTORS AND AI	9
People and Computers: Are They in the Same Category? – HCAI Principles – Impact of AI on Society – Importance of Human Values.	
UNIT III – FAIRNESS, BIAS AND PRIVACY PRESERVATION	9
Sources of Bias – Group Fairness – Exploratory Data Analysis – Limitation of a Dataset – Group Fairness and Individual Fairness – Counterfactual Fairness – Attack Models – Privacy Preserving Learning – Differential Privacy – Federated Learning.	
UNIT IV – HCAI FRAMEWORKS AND DESIGN METAPHORS	9
Levels of Automation – Defining Reliable, Safe and Trustworthy Systems – Two-Dimensional HCAI Framework: Life-Critical Systems – Golden Rules for Design – Application Examples – Science and Innovation Goals – Intelligent Agents – Tele-bots – Assured Autonomy: Design, Control Centers – Social Robots: Active Appliances – Enhancing Human Capabilities.	

UNIT V – HCAI GOVERNANCE STRUCTURES	9
Ethical Implications of Emerging AI Technologies – Software Engineering Best Practices – Establishing Safety Cultures in Organizations – Independent Oversight and Certification – Role of Government Regulation.	
TOTAL PERIODS	45

REFERENCES:
<ol style="list-style-type: none"> 1. Ben Shneiderman, "Human-Centered AI: A New Synthesis", Oxford University Press, 1st Edition, 2022. 2. Virginia Dignum, "Responsible Artificial Intelligence: How to Develop and Use AI in a Responsible Way", Springer Nature, 2019. 3. Christoph Molnar, "Interpretable Machine Learning: A Guide for Making Black Box Models Explainable", Lulu Press, 2nd Edition, 2022. 4. Paula Boddington, "AI Ethics", Springer Nature Switzerland AG, 2023.

Mapping of COs with POs / PSOs					
COs	POs			PSOs	
	1	2	3	1	2
1	1	-	-	1	-
2	3	-	1	3	-
3	2	-	2	-	2
4	-	3	2	3	1
5	2	3	3	3	2
CO (W.A)	1.6	1.2	1.6	2.0	1.0


 Chairman-Board of Studies
 Computer Science and Engineering
 Nandha College of Technology (Autonomous)
 Erode - 635 052.

CP25C24 - VIBE CODING		L	T	P	C
		3	0	0	3
Course Objective	<ul style="list-style-type: none"> To introduce the fundamentals of AI-assisted coding and Vibe Coding, enabling learners to effectively utilize AI tools for software development and problem-solving. To develop practical skills in building AI-powered applications while understanding the ethical, professional, and collaborative aspects of AI-driven development. 				
Course Outcomes: The Student will be able to					Cognitive Level
CO1	Explain the basics of AI-assisted coding and Vibe Coding in software development.				U
CO2	Use AI coding tools to generate, debug, and improve program code.				Ap
CO3	Analyze AI-assisted development workflows and collaboration practices to enhance productivity.				An
CO4	Develop simple AI-powered applications using AI tools and programming concepts.				C
CO5	Apply ethical and responsible practices while using AI-assisted coding tools in software development.				Ap

UNIT I – INTRODUCTION TO VIBE CODING AND AI-ASSISTED DEVELOPMENT	9
Introduction to Vibe Coding – Basic Prompting Techniques for Coding Tasks – Roles and Responsibilities in Human-AI Collaboration – AI Assistant, Prompt Engineer, and Developer – Designing Simple AI-Augmented Workflows – Interaction Between AI-Generated Code and Developer Feedback – Overview of AI-Assisted Coding Tools – Cursor, v0, Bolt, and Similar Platforms – Features and Use Cases of AI Coding Tools.	
UNIT II – AI CODING PLATFORMS AND APPLICATION DEVELOPMENT	9
Setting Up and Navigating AI Coding Platforms – Building Simple Web Applications with AI Integration – Workflow Prototyping Using AI Tools – Vibe Coder’s Toolkit: Prompts, Context, and Flow – Prompt Optimization Techniques – Human Oversight in AI Coding Systems – Best Practices for Effective AI-Assisted Development.	
UNIT III – HUMAN-AI COLLABORATION AND THE FUTURE OF SOFTWARE DEVELOPMENT	9
The Unbundled Programmer and the Future of Coding Roles – Evolving Developer Roles in the AI Era – Shift from Coding to Orchestration and Strategy – AI Integration Across the Software Development Lifecycle – Human-AI Collaboration in Design, Debugging, Testing, and Deployment – Strategic Thinking and Adaptability in AI-Augmented Teams.	
UNIT IV – CODE QUALITY, RELIABILITY, AND ETHICAL AI PRACTICES	9
Long-Term Maintainability of AI-Generated Code – Refactoring AI-Assisted Code for Clarity and Standards – Documentation Strategies for Mixed Human-AI Codebases – Common Security and	

Reliability Issues in AI-Generated Code – Ethical and Responsible AI Usage – Ethical Coding Practices in AI-Augmented Environments – Cultural Shifts, Accountability, and Governance in Vibe Coding.

UNIT V – APPLIED PROJECTS AND REAL-WORLD AI APPLICATIONS

9

Development of AI-Powered Applications – AI-Based Task Manager and To-Do Lists – Resume Builder with Smart AI Suggestions – Note-Taking Application with Auto-Organization – Recipe Generator with Ingredient Substitution Logic – Visual Memory Game Using AI Tools – Non-Diagnostic Health Symptom Checker – Project Design, Development, Testing, and Demonstration.

TOTAL PERIODS

45

REFERENCES:

1. Packt Publishing, GitHub Copilot Essentials: Accelerate Software Development with AI-Assisted Programming, 1st Edition, 2024.
2. Tom Taulli, Artificial Intelligence Basics: A Non-Technical Introduction, Second Edition, Apress, 2023.
3. Gene Kim, Steve Yegge, "Vibe Coding: Building Production-Grade Software with GenAI, Chat, Agents, and Beyond", IT Revolution Press, 1st Edition, 2025.
4. Andrew Ng, "Generative AI for Everyone", DeepLearning.AI, 1st Edition, 2024.

Mapping of COs with POs / PSOs					
COs	POs			PSOs	
	1	2	3	1	2
1	3	2	1	2	1
2	3	2	2	3	1
3	2	3	2	3	2
4	2	2	3	2	3
5	2	3	2	2	2
CO (W.A)	1.8	2.2	2.4	2.4	2.2


 Chairman-Board of Studies
 Computer Science and Engineering
 Nandha College of Technology (Autonomous)
 Erode - 638 052.

CP25C25 - FEDERATED LEARNING		L	T	P	C
		3	0	0	3
Course Objective	<ul style="list-style-type: none"> To learn Federated Learning algorithms, privacy-preserving techniques, and challenges in decentralized environments. To apply Federated Learning frameworks and tools for designing and developing practical distributed AI applications through project-based learning. 				
Course Outcomes: The Student will be able to					Cognitive Level
CO1	Understand the fundamental concepts, architecture, and components of Federated Learning.				U
CO2	Describe Federated Learning workflows, communication protocols, and decentralized learning processes.				U
CO3	Apply Federated Learning algorithms and privacy-preserving techniques in distributed environments.				Ap
CO4	Analyze system challenges such as data heterogeneity, communication efficiency, and security in Federated Learning.				An
CO5	Apply Federated Learning frameworks and tools to develop practical decentralized AI applications.				Ap

UNIT I – INTRODUCTION TO FEDERATED LEARNING	9
Definition and Scope – Motivation: Data Privacy, Decentralization, Compliance (GDPR, HIPAA) – Centralized vs. Federated vs. Distributed Learning – Types of FL: Horizontal, Vertical, and Transfer FL – Real-World Applications: Mobile Keyboard (GBoard), Healthcare, IoT – FL System Architecture – Federated Learning Workflow and Lifecycle – Communication Protocols.	
UNIT II – FEDERATED LEARNING ALGORITHMS	9
Federated Averaging (FedAvg) – Federated SGD (FedSGD) – FedProx for handling system heterogeneity – Personalized Federated Learning (pFedMe, FedPer, FedBN) – FL with Non-IID Data – Aggregation and Weight Update Strategies – Convergence Issues – Performance Metrics.	
UNIT III – PRIVACY AND SECURITY IN FL	9
Security Threats in FL – Differential Privacy (DP): Concept, Application in FL – Secure Multiparty Computation (SMC) – Homomorphic Encryption (Overview) – Secure Aggregation Protocols – Byzantine Robustness – Model Inversion Attacks – Data Poisoning – Mitigation Strategies.	
UNIT IV – SYSTEM & COMMUNICATION CHALLENGES	9
Device Heterogeneity – Intermittent Availability – Resource Constraints (Memory, Battery, Network) – Client Selection Strategies – Communication Efficiency: Quantization, Compression, Sparsification – Dropout and Partial Client Participation – FL over Edge Devices – Energy-aware FL.	

UNIT V – APPLICATIONS AND FRAMEWORKS	9
FL Applications in Healthcare, Finance, Smart City, Recommender Systems – Case Studies – Tools and Frameworks: TensorFlow Federated, PySyft, Flower – FL Simulators. Project Work: Building a Simple Federated Learning Model – Model Deployment and Evaluation – Future Directions: Split Learning, Hybrid FL, Federated Analytics, FL + AI/ML.	
TOTAL PERIODS	45

REFERENCES:

1. Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated Learning", PHI Learning Pvt. Ltd., 2021.
2. M. Mohri, A. Rostamizadeh, and A. Talwalkar, "Foundations of Machine Learning", 2nd Edition, Pearson Education, 2020.
3. K. P. Murphy, "Machine Learning: A Probabilistic Perspective", Pearson Education, 2017.
4. C. M. Bishop, "Pattern Recognition and Machine Learning", Oxford University Press, 2016.
5. T. Mitchell, "Machine Learning", Tata McGraw-Hill Education, 2017.

Mapping of COs with POs / PSOs					
COs	POs			PSOs	
	1	2	3	1	2
1	1	-	-	1	-
2	3	-	1	3	-
3	2	-	2	-	2
4	-	3	2	3	1
5	2	3	3	3	2
CO (W.A)	1.6	1.2	1.6	2.0	1.0


 Chairman-Board of Studies
 Computer Science and Engineering
 Nandha College of Technology (Autonomous)
 Erode - 638 052.

CP25002 - DEEP LEARNING FOR COMPUTER VISION		L	T	P	C
		3	0	0	3
Course Objective	<ul style="list-style-type: none"> To understand the fundamental concepts, techniques, and applications of Computer Vision for automatic analysis and interpretation of images and videos. To apply deep learning techniques to develop Computer Vision solutions for real-world applications such as security, healthcare, entertainment, and mobility. 				
Course Outcomes: The Student will be able to					Cognitive Level
CO1	Explain the fundamental concepts of image processing, feature extraction, and deep learning techniques used in Computer Vision.				U
CO2	Apply convolutional neural network (CNN) architectures for image classification and visual recognition tasks.				Ap
CO3	Analyze computer vision problems using advanced CNN-based techniques for object detection and image segmentation.				An
CO4	Develop vision-based applications using recurrent neural networks, attention mechanisms, and transformer models.				Ap
CO5	Evaluate deep generative models and modern deep learning approaches for real-world computer vision applications.				E

UNIT I – INTRODUCTION TO IMAGE FUNDAMENTALS	9
Introduction to Image Formation – Capture and Representation – Linear Filtering – Correlation – Convolution – Edge – Blobs – Corner Detection – Scale Space and Scale Selection – SIFT – SURF – HoG – LBP. Bag-of-words – VLAD – RANSAC – Hough transform – Pyramid Matching – Optical Flow – Review of Deep Learning – Multi-layer Perceptron – Backpropagation.	
UNIT II – CONVOLUTIONAL NEURAL NETWORKS (CNN)	9
Introduction to CNNs – Evolution of CNN Architectures – AlexNet – ZFNet – VGG – InceptionNets – ResNets – DenseNets – Visualization of Kernels – Backpropagation to image/Deconvolution Methods – Deep Dream – Hallucination – Neural Style Transfer – CAM – Grad – CAM – Grad CAM++ – Recent Methods (IG, Segment-IG, SmoothGrad).	
UNIT III – APPLICATIONS OF CNN	9
CNNs for Recognition and Verification (Siamese Networks, Triplet Loss, Contrastive Loss, Ranking Loss) – CNNs for Detection: Background of Object Detection – R-CNN – Fast R-CNN – Faster R-CNN – YOLO – SSD – RetinaNet – CNNs for Segmentation: FCN – SegNet – U-Net – Mask-RCNN.	
UNIT IV – RECURRENT NEURAL NETWORKS (RNN) AND ATTENTION MODELS	9
Introduction to RNNs – CNN and RNN Models for Video Understanding: Spatiotemporal Models, Action/Activity Recognition – Introduction to Attention Models in Vision – Vision and Language: Image Captioning – Visual QA – Visual Dialog – Spatial Transformers – Transformer Networks.	

UNIT V – DEEP GENERATIVE MODELS	9
Deep Generative Models: GANs – VAEs – Other Generative Models: PixelRNNs, NADE, Normalizing Flows - Applications: Image Editing – Inpainting – Superresolution - 3D Object Generation – Security - Variants: CycleGANs – Progressive GANs – StackGANs - Pix2Pix. Zero-shot – One-shot – Few-shot Learning – Self-supervised Learning – Reinforcement Learning in Vision – Applications.	
TOTAL PERIODS	45

REFERENCES:

1. Ian Goodfellow, Yoshua Bengio, Aaron Courville, “Deep Learning”, MIT Press, 2016.
2. Michael Nielsen, “Neural Networks and Deep Learning”, 2016.
3. Yoshua Bengio, “Learning Deep Architectures for AI”, Now Publishers, 2009.
4. Richard Szeliski, “Computer Vision: Algorithms and Applications”, Springer Publisher, 2022.
5. Simon Prince, “Computer Vision: Models, Learning, and Inference”, Cambridge University Press, 2012.
6. David Forsyth, Jean Ponce, “Computer Vision: A Modern Approach”, Pearson Education, 2002.

Mapping of COs with POs / PSOs					
COs	POs			PSOs	
	1	2	3	1	2
1	3	2	1	2	1
2	3	2	2	3	1
3	2	3	2	3	2
4	2	2	3	2	3
5	2	3	3	2	2
CO (W.A)	1.8	2.2	2.4	2.4	2.4


Chairman-Board of Studies
Computer Science and Engineering
Nandha College of Technology (Autonomous)
Erode - 638 052.