ANNA UNIVERSITY, CHENNAI

NON- AUTONOMOUS COLLEGES

AFFILIATED TO ANNA UNIVERSITY

M.E. COMPUTER SCIENCE AND ENGINEERING

REGULATIONS 2025

PROGRAMME OUTCOMES (POs):

PO	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

PROGRAMME SPECIFIC OUTCOMES:

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

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

ANNA UNIVERSITY, CHENNAI

POSTGRADUATE CURRICULUM (NON-AUTONOMOUS AFFILIATED INSTITUTIONS)

Programme: M.E. Computer Science and Engineering **Regulations:** 2025

Abbreviations:

BS – Basic Science (Mathematics)

ES – Engineering Science (Programme Core (**PC**), Programme Elective (**PE**))

SD – Skill Development

SL – Self Learning

OE – Open Elective

L – Laboratory Course

T – Theory

LIT – Laboratory Integrated Theory

PW – Project Work

TCP – Total Contact Period(s)

Semester I									
S. No.	**Course Code**	**Course Title**	**Type**	**Periods per week**			**TCP**	**Credits**	**Category**
				L	**T**	**P**			
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 Credits							**22**	**19**	

Semester II									
S. No.	**Course Code**	**Course Title**	**Type**	**Periods per week**			**TCP**	**Credits**	**Category**
				L	**T**	**P**			
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 Credits							**17**	**19**	

Semester III									
S. No.	**Course Code**	**Course Title**	**Type**	**Periods per week**			**TCP**	**Credits**	**Category**
				L	**T**	**P**			
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.		Open Elective	-	3	0	0	3	3	-
5.		Industry-Oriented Course II	-	1	0	0	1	1	SD
6.	CP25301	Project Work I	-	0	0	12	12	6	SD
Total Credits							**25**	**19**	

Semester IV									
S. No.	Course Code	Course Title	Type	Periods per week			TCP	Credits	Category
				L	T	P			
1.	CP25401	Project Work II	-	0	0	24	24	12	SD
						Total Credits	24	12	

PROGRAMME ELECTIVE COURSES (PE)

S. No.	Course Code	Course Title	Type	Periods per week			TCP	Credits
				L	T	P		
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

MA25C07	Advanced Mathematical Methods (CSIE)	L	T	P	C
		3	1	0	4

Course Objectives:

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

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:

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

E-resources:

1. 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/afrid18/Data_structures_and_algorithms_in_cpp
3. https://www.udemy.com/course/introduction-to-algorithms-and-data-structures-in-c/?srsltid=AfmBOorEZlkgV7QzaEh6IqzAaKLjC-IpFU1NGgWFoHMLhOos-

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

Course Objectives:

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

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

Activity: Design ER Model for a specific use case.

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

Activity: Demonstration of SQL Implementation.

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.

Activity: Demonstration of Object Query Language.

Query Processing and Optimization: Query Processing, Query Trees and Heuristics, Query Execution Plans, Cost Based Optimization.

Activity: Design of Query Evaluation Plans.

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.

Activity: Demonstration of Concurrency and Transactions.

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.

Activity: Design application with MongoDB.

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:

- 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:

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

Activities:

1. Convert source code to SSA form using LLVM IR.
2. 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:

1. Perform loop unrolling and strength reduction.
2. 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:

1. Implement loop tiling and loop skewing on a matrix multiplication program.
2. 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:

1. Develop a basic JIT-enabled interpreter with LLVM or GraalVM.
2. 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:
This course enables students to learn parallel programming models, tools, and techniques using shared and distributed memory systems. The course emphasizes the implementation and performance evaluation of parallel applications using OpenMP and MPI.

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

Practical:

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

Thread Management and Synchronization Techniques: Synchronization primitives: Mutex - Locks - Semaphores - Data races and deadlocks - Inter-thread communication: Condition variables- Pipes

Practical:

1. Write a program to compute the Fibonacci series using OpenMP tasks.
2. Write a Rust program to spawn threads and use channels for communication.

Parallel Programming using OpenMP for Shared Memory: OpenMP architecture and memory model - Directives and work-sharing constructs - Parallelizing Loops and Task Management - Performance optimization

Practical:

1. Write a program using MPI to calculate the average of an array using collective communication

Message Passing Programming with MPI: MPI programming concepts - Point-to-point and collective communication - Derived data types and synchronization - Performance tuning.

Practical:

1. Write a program using the threading module to demonstrate data race and synchronization.
2. Write a program to perform matrix multiplication using OpenMP.

Evaluating Parallel Performance and Scalability: Measuring and analyzing performance - Scalability metrics - Use of Profiling Tools for Performance Bottlenecks.

Practical:

1. Write a Rust program to parallelize a for loop using the rayon crate.
2. Write a program and profile its performance using `gprof` or `perf`.

Advanced Concepts and Practical Case Studies: Tree search, N-body simulation, graph algorithms - Comparative Analysis of OpenMP and MPI Implementations - Trends in multicore and manycore processors.

Practical:

1. Mini Project: Write a C program to implement parallel merge sort or breadth-first search using both OpenMP and MPI.Compare complexity.

Course Outcomes:

- Explain the architectural features and classifications of multicore processors
- Demonstrate the use of synchronization primitives to ensure correctness in parallel code
- Implement shared memory parallel programs using OpenMP
- Develop scalable MPI programs for distributed memory architectures
- Analyze and compare performance of OpenMP and MPI implementations for real-world problems

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

Assessment Methodology: Assignments (30), Quiz (10), Virtual Demo (25), Flipped Class Room (10), Review of Gate and IES Questions (25)

References:

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

E-Resources:

1. NPTEL Course: https://nptel.ac.in/courses/106104189
2. OpenMP Official Documentation: https://www.openmp.org
3. MPI Forum: https://www.mpi-forum.org

4. Coursera: https://www.coursera.org/learn/parallel-programming-in-java
5. Book PDF: Multicore Application Programming by Darryl Gove (Intel site)

CO-PO-PSO Mapping

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, evolution, and components of multicore processor architectures and their role in modern computing systems.	--	--
CO2	**Analyze** the performance, parallelism, and scalability issues of multicore architectures using appropriate architectural and programming models.	PO1 (3)	PSO1 (3)
CO3	**Evaluate** different multicore design techniques, cache coherence protocols, and interconnection networks for their effectiveness in specific computing scenarios.	PO3 (2)	PSO2 (2)
CO4	**Design** efficient multicore-based solutions by selecting suitable architectures, synchronization mechanisms, and parallel execution strategies for real-world applications.	PO2 (1)	PSO1 (3)

CP25C05	Artificial Intelligence and Machine Learning	L	T	P	C
		3	0	0	3

Course Objectives:

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

Problem Solving and Knowledge Representation: 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.

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

Uncertain Knowledge and Reasoning: 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

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

Machine Learning Introduction: Types of learning – Hypothesis space – Inductive bias – Evaluation – Cross validation – Bias variance – Bias Variance Trade off.

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

Supervised Learning: 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.

Activities: Behaviour Modelling, Disease Diagnosis and any other similar ones suitable for supervised learning

Unsupervised and Probabilistic Learning: 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.

Activities: Customer Segmentation, Image Processing and any other similar ones suitable for unsupervised and probabilistic learning

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

Assessment Methodology: Assignments (30), Quiz (10), Virtual demonstration (25), Flipped Classroom (10), Review of GATE & IES questions (25).

References:

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

E-resources:

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

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, scope, and applications of artificial intelligence and machine learning in solving real-world problems.	--	--
CO2	**Analyze** various artificial intelligence and machine learning models to understand their behavior, performance, and suitability for different problem domains.	PO1 (3)	PSO1 (3)
CO3	**Evaluate** machine learning algorithms and artificial intelligence techniques based on accuracy, efficiency, and applicability to specific use cases.	PO3 (2)	PSO2 (2)
CO4	**Design** intelligent systems by selecting and integrating appropriate machine learning models and AI techniques to address complex real-world applications.	PO2 (1)	PSO1 (3)

CP25C06	Cloud and Big Data Analytics	L	T	P	C
		3	0	0	3

Course Objectives:

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

Cloud Computing: 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

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

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

Activities: Watch Hadoop tutorial and submit a summary report.

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

Activities: HiveQL assignments on datasets from Kaggle.

Advanced Big Data Analytics: Introduction to Apache Spark - Spark vs. MapReduce - Spark SQL, Data Frames, RDDs- Machine Learning using Spark MLlib - Real-time processing: Kafka and Spark Streaming

Activities: Mini-project using PySpark on COVID or Traffic dataset.

Cloud Integration and Security: 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

Activities: Create a cloud deployment report using Azure/AWS.

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

Assessment Methodology: Assignments (30), Quiz (10), Virtual demonstration (25), Flipped Classroom (10), Review of GATE & IES questions (25).

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.

E-resources:

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

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, architectures, and service models of cloud computing and big data analytics used in modern data-driven environments.	--	--
CO2	**Analyze** cloud-based big data processing frameworks and analytics techniques to understand their performance, scalability, and resource utilization.	PO1 (3)	PSO1 (3)
CO3	**Evaluate** big data analytics tools, cloud platforms, and data processing models for their effectiveness in handling large-scale and complex datasets.	PO3 (2)	PSO2 (2)
CO4	**Design** scalable cloud-based big data analytics solutions by selecting appropriate architectures, platforms, and analytical methods for real-world applications.	PO2 (1)	PSO1 (3)

		L	T	P	C
CP25C07	**Quantum Computing**	2	0	0	2

Course Objective:

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

Physical properties and mathematical foundations

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

Activities:

- Simulate the experiment using an interactive virtual lab
- Construct simple operators and visualize action on vectors

Quantum computing postulates and gates

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.

Activities:

- Group quiz to match postulates to physical implications
- Visualization with Bloch sphere simulators

Quantum computing circuits

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

Activities:

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

Fundamental Quantum Algorithms

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

Activities:

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

Programming on a real quantum computer

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

Activities:

- Connect Qiskit with IBMQ using personal API token
- Hands-on: Create 1- and 2-qubit circuits using Hadamard, X, Z, and measurement gates
- Compare real and simulated results and Observe impact of quantum noise

Text Books:

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

References:

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

Evaluation method to be used

Continuous Assessment	Activities	End Semester
50%	10%	40%

CO	Description of CO	PO	PSO
CO1	Describe the fundamental principles, postulates, and computational models of quantum computing and their significance in next-generation computing systems.	--	--
CO2	**Analyze** quantum algorithms and quantum circuit models to understand their computational advantages, limitations, and performance characteristics.	PO1 (3)	PSO1 (3)
CO3	**Evaluate** quantum computing paradigms, error correction techniques, and hardware technologies for their suitability in solving complex computational problems.	PO3 (2)	PSO2 (2)
CO4	**Design** quantum circuits and algorithmic solutions by selecting appropriate quantum gates, qubit architectures, and computational models for real-world problem scenarios.	PO2 (1)	PSO1 (3)

CP25C08	Advanced Software Testing and Quality Assurance	L	T	P	C
		3	0	0	3

Course Objectives:

1. Provide an in-depth understanding of advanced software testing techniques and their role in quality assurance.
2. Explore automated testing frameworks, tools, and methods for test design and test process improvement.
3. Enable students to design test plans, develop test cases, and assess test results critically.

Foundations and Test Lifecycle: Evolution of software testing in agile and DevOps ecosystems -Software Test Life Cycle (STLC) phases: Requirements analysis to test Closure - Quality attributes: reliability, maintainability, testability - Introduction to testing types and classifications.

Activity: Mapping STLC to Agile/DevOps pipelines.

Types of Testing and Risk-Based Approaches: Types of testing: Unit, Integration, System, Acceptance, Regression, Smoke, and Sanity - Risk-based testing and test Prioritization - Aligning testing strategies with product risk and delivery goals

Activity: Test strategy design for risk-based scenarios

Advanced Test Design Techniques: Black-box and white-box test strategies: BVA, ECP, decision tables, control/data flow - State transition and use-case testing - Model-based testing, combinatorial testing, and cause-effect graphing - Mutation testing, fault injection, fuzz testing - Test case minimization and prioritization techniques

Activity: Create test cases for a real-world problem using model-based design

Test Automation and Continuous Testing: Design of reusable and maintainable automation frameworks - Open-source tools: Selenium, TestNG, JUnit, Robot Framework, Cypress - Performance testing: JMeter and basic LoadRunner Concepts - Integration with Jenkins and GitHub for continuous testing - BDD and TDD using Cucumber **Activity:** Implement a hybrid automation framework for a web-based application
Software Quality Assurance Frameworks and Metrics: Quality planning, control, assurance, and improvement - SQA tools and activities: Reviews, audits, inspections - Quality models: McCall, Boehm, ISO 9126 / ISO 25010 -Metrics for testing: test coverage, defect density, MTBF, test effort -Cost of quality and defect leakage **Activity:** Analyze QA strategy in enterprise software (e.g., ERP or banking)
Test Process Improvement and Research Trends: Test maturity models: TMMi, CMMI-Dev (QA perspective) - Test process audits, benchmarking, and reviews - Empirical methods in testing and defect prediction models -AI/ML-based testing approaches: self-healing tests, anomaly detection -Testing trends: shift-left/right testing, visual and mobile testing. **Activity:** Design a test improvement roadmap based on TMMi for a sample company
Weightage: Continuous Assessment: 40%, End Semester Theory Examinations: 60%
Assessment Methodology: Assignments (30), Quiz (10), Virtual Demo (25), Flipped Class Room (10), Review of Gate and IES Questions (25)
References: 1. Glenford J. Myers, The Art of Software Testing, 3rd edition, Wiley 2. Paul C. Jorgensen, Software Testing: A Craftsman's Approach, 4th edition, CRC Press 3. Ron Patton, Software Testing, Second Edition, Sams Publishing, Pearson Education, 2007 4. Aditya Mathur, Foundations of Software Testing, 2nd edition, Pearson, 2013 5. Pressman & Maxim, Software Engineering: A Practitioner’s Approach, McGraw-Hill.
E-Resources: 1. MIT OpenCourseWare (Software Testing) 2. NPTEL (Sftware Testing by Prof. Nandini Prasad, IIT Kharagpur)

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, principles, and processes involved in advanced software testing and quality assurance for reliable software development.	--	--
CO2	**Analyze** software systems and testing strategies to identify defects, assess risks, and ensure compliance with quality standards.	PO1 (3)	PSO1 (3)
CO3	**Evaluate** software testing techniques, quality models, and assurance frameworks to determine their effectiveness in diverse development environments.	PO3 (2)	PSO2 (2)
CO4	**Design** comprehensive testing and quality assurance strategies by selecting appropriate testing tools, automation frameworks, and quality metrics for real-world software projects.	PO2 (1)	PSO1 (3)

CP25C09	**Agile Methodologies**	L	T	P	C
		3	0	0	3

Course Objectives:

- Introduce students to the principles, values, and practices of Agile software development.
- Provide hands-on experience with Agile tools and techniques used in industry.
- Develop the ability to plan, manage, and deliver software in Agile environments.

Evolution of Software Development and Agile Foundations: Evolution of Software Development Methodologies - Challenges in traditional models that led to Agile - The Agile Manifesto - Agile Values and Principles

Activity: Agile vs Waterfall in large-scale government or startup projects (e.g., FBI Sentinel, Spotify)

Agile Mindset and Cultural Transformation: Agile Mindset in Individuals and Teams - Characteristics of Agile teams and leaders - **Lean Thinking in Software Development -** Cultural Shift from Traditional to Agile Thinking - Applying Lean in Agile - Agile vs Traditional Methodologies

Activity: Reflective write-up on Agile mind set adoption challenges in traditional teams

Agile Process Models and Frameworks: Scrum: Roles, artifacts, events - Extreme Programming (XP): Pair programming, test-first, refactoring - Kanban: Flow, work-in-progress (WIP), pull systems - SAFe (Scaled Agile Framework) and Disciplined Agile Delivery (DAD) - Feature-Driven Development (FDD), Crystal, LeSS, Nexus

Activity: Team simulation using Scrum ceremonies (sprint planning, stand-ups, retrospectives)

Agile Project Management and Planning: User stories and story mapping - Estimation techniques: Planning poker, T-shirt sizing - Product backlog grooming and sprint planning - Agile metrics: Velocity, burndown chart, cumulative flow diagram (CFD) - Risk management in Agile

Activity: Tool Demo**:** Jira, Trello, Azure DevOps

Agile Engineering Practices: Test-Driven Development (TDD) and Behaviour Driven Development (BDD) - Continuous Integration and Continuous Delivery (CI/CD) - DevOps principles and its relationship with Agile - Refactoring, code smells, clean code - Design patterns in Agile context

Activity: Lab Component: Automate testing and CI/CD for a sample app

---

Agile at Scale and Industry Case Studies: Agile scaling frameworks and challenges - Distributed Agile teams and remote collaboration - Agile in non-software contexts (e.g., education, HR, manufacturing) - Agile contracting and governance - Agile case studies from Google, Spotify, IBM, and startups

Suggested Activity:

Activity: Project based experiential learning: End-to-end Agile project using Scrum/Kanban and CI/CD pipeline

---

References:

1. Ken Schwaber & Jeff Sutherland, Software in 30 Days: How Agile Managers Beat the Odds, Wiley
2. Mike Cohn, User Stories Applied: For Agile Software Development, Addison-Wesley
3. Craig Larman & Bas Vodde, Scaling Lean & Agile Development, Addison-Wesley
4. Robert C. Martin, Clean Code: A Handbook of Agile Software Craftsmanship, Pearson

---

E-Resources:

1. NPTEL***:*** Agile Software Development – Prof. Rajib Mall (IIT Kharagpur)
2. MIT Open Course Ware Agile *content*

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

Assessment Methodology: Assignments (30), Quiz (10), Virtual Demo (25), Flipped Class Room (10), Review of Gate and IES Questions (25)

CO	Description of CO	PO	PSO
CO1	Describe the fundamental principles, values, and frameworks of agile methodologies and their role in modern software development practices.	--	--
CO2	**Analyze** agile processes and practices to understand their effectiveness in managing requirements, collaboration, and iterative software development.	PO1 (3)	PSO1 (3)
CO3	**Evaluate** agile models, project management techniques, and team practices to assess their suitability for different organizational and project contexts.	PO3 (2)	PSO2 (2)
CO4	**Design** agile-based project workflows by selecting appropriate methodologies, roles, and practices to deliver high-quality software solutions.	PO2 (1)	PSO1 (3)

CP25C10	Web of Things	L	T	P	C
		3	0	0	3

Course Objective:

To equip students with the knowledge and skills to integrate Internet of Things (IoT) devices with web technologies using the Web of Things (WoT) framework. This course enables learners to design interoperable, secure, and scalable smart applications by leveraging standardized WoT architectures, Thing Descriptions, and scripting interfaces.

Introduction to Web of Things: Evolution: From IoT to WoT, Architecture of WoT: Thing Description, Binding, Servient, Protocol Bindings, Web protocols overview (HTTP, Web Sockets, MQTT, CoAP), WoT Scripting and Interaction Models (Properties, Actions, Events)

Activities:

- Mind map comparison of IoT vs WoT
- Analysis of WoT reference architecture

WoT Standards and Thing Description (TD): W3C WoT Architecture and Building Blocks, Thing Description (TD): Syntax, Vocabulary, JSON-LD format, TD Schema validation and interpretation, Semantic Annotation in TD

Activities:

- Write TD for a smart device in JSON-LD
- Validate TD using WoT tools

Implementation of WoT Servients: WoT Servient structure: Exposed vs Consumed Things, Node-WoT framework, WoT Scripting API, Protocol bindings: HTTP, MQTT, CoAP

Activities:

- Build a WoT Servient with Node-WoT
- Demo using WoT Scripting API

Interoperability, Security, and Privacy: Interoperability challenges, Security mechanisms: HTTPS, OAuth2.0, Access Control, Privacy and anonymization, Trust and identity in WoT ecosystems

Activities:

- Present security issues in WoT
- Design secure architecture for smart home

Applications and Future Trends in WoT: Smart Cities, Smart Homes, e-Health, IIoT, Cloud and Edge integration, Data analytics and AI in WoT, Future directions: Digital twins, Metaverse

Activities:

- Present case studies of real-world WoT
- Brainstorm WoT smart city features

Integration to WoT: Integrating an Arduino to WoT – Integrating BeagleBone to WoT – Integrating an Intel Edition to WoT – Integrating other Embedded systems to WoT **Activities**: Create a simple project to integrate an embedded device with WoT
Weightage: Continuous Assessment: 40%, End Semester Theory Examinations: 60%
Assessment Methodology: Assignments (30), Quiz (10), Virtual Demo (25), Flipped Class Room (10), Review of Gate and IES Questions (25)
References 1. Michael Kuniavsky, *Smart Things: Ubiquitous Computing User Experience Design*, Morgan Kaufmann, 2010. 2. Dom Guinard & Vlad Trifa, *Building the Web of Things*, Manning Publications, 2016. 3. akima Chaouchi, *The Internet of Things: Connecting Objects to the Web*, Wiley, 2010. 4. Peter Waher, *Mastering Internet of Things*, Packt Publishing, 2018. 5. Alasdair Gilchrist, *Industry 4.0: The Industrial Internet of Things*, Apress, 2016.
Web resource W3C WoT Working Group Documents: https://www.w3.org/WoT/H

CP25001	Text and Speech Processing	L	T	P	C
		3	0	0	3

Course Objective:

To introduce the fundamental principles and practices of text and speech processing, equipping students with theoretical understanding and practical skills in natural language and speech technologies, including parsing, information extraction, speech recognition, and synthesis

Introduction to Text and Speech Processing: NLP pipeline overview Human speech production and acoustic phonetics Text normalization: tokenization, stemming, lemmatization Basics of speech signal: sampling, frequency, noise

Activities:

- Speech waveform analysis using Praat
- Tokenization and stemming with NLTK

Text Processing Techniques: Part-of-speech tagging Parsing techniques: CFG, Dependency Parsing Named Entity Recognition (NER), Chunking, Language modeling: N-gram models

Activities:

- POS tagging using spaCy
- Building a rule-based NER system

Speech Signal Processing: Feature extraction: MFCC, PLP, LPC Acoustic modeling: HMM, GMM Basics of ASR (Automatic Speech Recognition) Hidden Markov Models for speech

Activities:

- Extract MFCC features using Python Group
- Basic ASR using pre-trained models

Speech Synthesis and Applications: Text-to-speech (TTS) systems Speech databases and corpora Evaluation metrics: WER, CER Applications: Voice Assistants, Dictation Systems

Activities:

- TTS Demo using pyttsx3
- Compare open-source TTS tools

NLP and Speech in Practice: Sentiment analysis Chatbots and dialog systems Multilingual and low-resource NLP Ethics in speech and text technology

Activities:

- Building a chatbot using Rasa
- Ethical issues in speech surveillance

Models for NLP: RNN's and LSTM – The Transformer – Large Language Models – Masked Language Models

Activities: Create a debate to Compare and contrast the pros and cons of various models for NLP

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

60%
Assessment Methodology: Assignments (30), Quiz (10), Virtual Demo (25), Flipped Class Room (10), Review of Gate and IES Questions (25)
References: 1. Daniel Jurafsky and James H. Martin, Speech and Language Processing, Pearson, 3rd edition draft, 2023. 2. Lawrence Rabiner and Biing-Hwang Juang, Fundamentals of Speech Recognition, Pearson, 2011. 3. Christopher D. Manning and Hinrich Schütze, Foundations of Statistical Natural Language Processing, MIT Press, 1999. 4. Thomas Schultz and Katrin Kirchhoff, Multilingual Speech Processing, Academic Press, 2006. 5. Jacob Eisenstein, Introduction to Natural Language Processing, MIT Press, 2019.

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, models, and techniques involved in text and speech processing for natural language and audio-based applications.	--	--
CO2	**Analyze** text and speech processing algorithms to understand their performance, accuracy, and limitations in real-world language and audio processing tasks.	PO1 (3)	PSO1 (3)
CO3	**Evaluate** text and speech processing methods, models, and systems to assess their effectiveness for various linguistic and speech-based applications.	PO3 (2)	PSO2 (2)
CO4	**Design** text and speech processing solutions by selecting and integrating appropriate models, features, and techniques for intelligent language and speech applications.	PO2 (1)	PSO1 (3)

CP25C11	Advanced Deep Learning and Neural Networks	L	T	P	C
		3	0	0	3

Course Objective:

This course aims to provide a comprehensive understanding of neural network architectures and their training methodologies. The course explores optimization techniques, regularization methods, and state-of-the-art architectures including encoder-decoder models, attention mechanisms, variational auto encoders, GANs, and transformer-based models.

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 **Activity:** Understanding the working principle of Neural Networks
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. **Activity:** Reproduction of research paper related to CNN and RNN
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. **Activity:** Simple application using deep neural network with three layers
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. **Activity:** Design of any Regressive, Generative and Graph Models
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. **Activity:** Design of face recognition model.
Advanced applications of Deep learning: Deep learning for Computer Vision – Deep learning for Time Series – Deep learning for text – Generative Deep learning **Suggested Activity:** medical image segmentation and time series data analysis
Weightage: Continuous Assessment: 40%, End Semester Theory Examinations: 60%
Assessment Methodology: Assignments (30), Quiz (10), Virtual Demo (25), Flipped Class Room (10), Review of Gate and IES Questions (25)

References

1. François Chollet, "Deep Learning with Python ", 2nd Edition, Manning Pubns Co, 2021.
2. Josh Patterson, Adam Gibson," Deep Learning: A Practitioner‟s Approach", 1st Edition, O‟Reilly Media, Inc, 2017.
3. Bengio, Yoshua, Ian J. Goodfellow, and Aaron Courville. "Deep learning." An MIT Press book in preparation, 2016.
4. Dr.Adrian Rosebrock, Deep Learning for Computer Vision with Python: Starter Bundle‖, PyImage Search, 1st Edition, 2017.
5. Michael Nielsen, Neural Networks and Deep Learning, Determination Press, 2015.

Online Courses

1. CS7015: Deep Learning (iitm.ac.in)
2. Deep Learning - Course (nptel.ac.in)
3. Deep Learning Part 1 (IITM) - Course (nptel.ac.in)
4. Deep Learning - IIT Ropar - Course (nptel.ac.in)

<u>CO-PO-PSO Mapping</u>

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, architectures, and learning paradigms of deep learning and neural networks used in advanced intelligent systems.	**--**	**--**
CO2	Analyze deep learning models and neural network architectures to understand their learning behavior, performance, and optimization characteristics.	**PO1 (3)**	**PSO1 (3)**
CO3	Evaluate deep learning techniques, neural network models, and training strategies to assess their effectiveness for complex data-driven applications.	**PO3 (2)**	**PSO2 (2)**
CO4	Design advanced deep learning solutions by selecting and integrating suitable neural network architectures, training algorithms, and optimization methods for real-world problems.	**PO2 (1)**	**PSO1 (3)**

CP25C12	Quantum Cryptography	L	T	P	C
		3	0	0	3

Course Objective:

The objective of this course is to introduce the principles of quantum information and computation, along with key quantum algorithms and their practical applications. It also aims to provide a clear understanding of post-quantum and advanced quantum cryptographic techniques essential for secure communication in the quantum era.

Quantum Information: Qubit - Single and Multiple qubits – Mathematical Model for Quantum mechanics – Quantum measurements – Quantum Computation and No-cloning: Phase shift – Bit flips – Hadamard transform – Arbitrary Transforms – Entanglement - Quantum gates and circuits: CNOT gate – CCNOT gate – Reversible gates - Universal quantum gates - quantum circuit – quantum parallelism

Activities

- Build and simulate quantum circuits using Qiskit (or another quantum simulator)
- Create a Bell state using a Hadamard gate followed by a CNOT gate.

Quantum Algorithms: Deutsch's algorithm – Phase Kickback – Generalizing to n bits: Deutsch-Jozsa algorithm – Simon's algorithm: Analysis - Quantum Fourier Transform, Grover's Algorithm, Shor's Algorithm.

Activities

- Simulate both Deutsch's algorithm (1-bit) and the Deutsch-Jozsa algorithm (n-bit) using Qiskit.

Quantum Cryptanalysis: Quantum order finding – Factoring – Discrete logarithms, Hidden subgroup problems (HSP) – Key Exchange: Diffie-Hellman (DH) problems - Computational DH – Decisional DH – Indistinguishable Chosen Plaintext Attack (IND-CPA): Applications in RSA Encryption and Elgamal Encryption

Activities

- Simulate Shor's Algorithm using Qiskit or any quantum simulator to factor small numbers

Post-Quantum Cryptography: Post Quantum Crypto: Introduction to lattices – Codes – Isogenies - Lattice Problems. Learning with Errors (LWE) and Short Integer Solution (SIS) problem. Connection to dihedral hidden subgroup problem - Public Key Encryption (PKE) from LWE - Fully Homomorphic Encryption (FHE).

Activities

- Implement a simplified version of a public-key encryption scheme based on the LWE problem using Python or SageMath

Advanced Quantum Cryptography - I : Quantum Key Distribution and bit commitment – Random Oracles - Quantum One Time Pad and Encryption - Quantum PKE – Quantum FHE - Quantum Indifferentiability - Quantum Money **Activities** • Simulate the BB84 protocol for quantum key distribution between two parties in Python using Qiskit
Advanced Qunatum Cryptography – II : Quantum key distribution with imperfect devices – beyond point -to -point quantum key distribution – device independent Quantum cryptography **Activities:** • Simulate Quantum key distribution between two IoT based device in python using Qiskit
Weightage: Continuous Assessment: 40%, End Semester Theory Examinations: 60%
Assessment Methodology: Assignments (30), Quiz (10), Virtual Demo (25), Flipped Class Room (10), Review of Gate and IES Questions (25)
References: 1. Quantum Computation and Quantum Information, M. A. Nielsen and I. Chuang, Cambridge University Press, 2012. 2. Quantum Computing from the Ground Up, Riley Tipton Perry, World Scientific Publishing Ltd., 2012. 3. Quantum algorithms via linear Algebra Primer, Richard J. Lipton Kenneth W. Regan, The MIT Press Cambridge, 2014. 4. Quantum Computing: An Applied Approach, Jack D. Hidary, 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

CO-PO-PSO Mapping

CO	Description of CO	PO	PSO
CO1	Describe the fundamental principles, protocols, and security concepts of quantum cryptography and their significance in secure communication systems.	--	--
CO2	**Analyze** quantum cryptographic protocols and security mechanisms to understand their robustness, vulnerabilities, and performance characteristics.	PO1 (3)	PSO1 (3)
CO3	**Evaluate** quantum cryptographic techniques and implementations to assess their effectiveness in ensuring confidentiality and secure key distribution.	PO3 (2)	PSO2 (2)
CO4	**Design** secure communication systems by applying appropriate quantum cryptographic protocols, architectures, and security strategies for real-world	PO2 (1)	PSO1 (3)

	applications.				
CP25C13	**Quantum Machine Learning**	L	T	P	C
		3	0	0	3

Course Objective:

The objective of the Quantum Machine Learning (QML) course is to provide students with a comprehensive understanding of the intersection between quantum computing and machine learning**.** This course aims to equip students with the theoretical foundation and practical skills necessary to design, develop, and implement quantum algorithms for solving complex machine learning problems.

Introduction to Quantum Computing and Quantum Machine Learning: 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

Activities:

- Implement a quantum circuit using Qiskit to demonstrate superposition and entanglement (Create a simple Bell state).
- Use Qiskit to implement and visualize Grover's search algorithm for an unsorted database problem.

Quantum Information Theory and Quantum Algorithms: Quantum Entropy and Quantum Information - Quantum Teleportation and Superdense Coding - No-Cloning Theorem and Quantum Cryptography - Quantum Fourier Transform (QFT) - Quantum Phase Estimation (QPE)

Activities:

- Implement a Quantum Fourier Transform (QFT) using Qiskit and visualize the output on a simple example (e.g., the 3-qubit QFT).
- Simulate quantum teleportation using Qiskit to transfer quantum information between two qubits.

Quantum Algorithms for Machine Learning: Quantum Support Vector Machines (QSVM) - Quantum Principal Component Analysis (QPCA) - Quantum k-Nearest Neighbors (QkNN) - Quantum Neural Networks (QNN)

Activities:

- Implement Quantum Support Vector Machine (QSVM) using Qiskit to classify a small dataset (e.g., Iris dataset).
- Use Qiskit to perform Quantum Principal Component Analysis (QPCA) on a synthetic dataset for dimensionality reduction.

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

Activities:

- Implement the Variational Quantum Eigensolver (VQE) to find the ground state energy of a Hamiltonian using Qiskit.
- Create a Quantum Approximate Optimization Algorithm (QAOA) to solve a Max-Cut problem on a simple graph using Qiskit**.**

Applications, Tools, and Future of Quantum Machine Learning: 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

Activities:

- Use TensorFlow Quantum to build a simple quantum classifier and run it on a small dataset.
- Explore Qiskit and create a quantum-based recommendation system using hybrid quantum-classical methods.

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

Activities

Write a case study about combinatorial optimization with the Ising model and its quantum counterpart.

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

Activities: Write a case study about Quantum Approximate Optimization Algorithm

Observations from an Industry Exposure Visit: For the Quantum Machine Learning (QML) course, a hands-on learning experience can be organized where students visit a quantum computing research lab, a quantum software company, or a machine learning-focused organization that integrates quantum computing into real-world applications. During this visit, students will have the opportunity to observe cutting-edge workflows involving quantum algorithms, quantum optimization techniques, and hybrid classical-quantum models applied to machine learning tasks. Students will gain insights into how these technologies are deployed to solve complex, high-dimensional problems in industries like finance, healthcare, and optimization. Following the visit, students will write a reflection report summarizing their observations, discussing the tools and technologies encountered, and reflecting on the practical challenges faced in quantum machine learning applications.

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

Assessment Methodology: Assignments (30), Quiz (10), Virtual Demo (25), Flipped Class Room (10), Review of Gate and IES Questions (25)

References :

1. Claudio Conti, Quantum Machine Learning Thinking and Exploration in Neural

Network Models for Quantum Science and Quantum Computing, Springer Nature Link Publications 2024

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

E-resources/E-materials:

1. **Qiskit Documentation**
   https://qiskit.org/documentation/
   IBM Quantum Experience
   https://quantum-computing.ibm.com/

2. **Quantum Computing for the Very Curious**
   https://quantum.country/qcvc

3. **Google Quantum AI**
   https://quantumai.google/

4. **TensorFlow Quantum**
   https://www.tensorflow.org/quantum

NPTEL/MOOC/SWAYAM Courses:

1. **NPTEL – Introduction to Quantum Computing**
   https://nptel.ac.in/courses/106/105/106105226/
2. **NPTEL – Quantum Computing and Quantum Information**
   https://nptel.ac.in/courses/106/106/106106220/
3. **Swayam – Quantum Machine Learning: Algorithms and Applications**
   https://swayam.gov.in/nd1_noc21_cs56/
4. **Coursera – Quantum Computing and Machine Learning**
   https://www.coursera.org/learn/quantum-machine-learning

Udemy – Quantum Machine Learning
https://www.udemy.com/course/quantum-machine-learning/

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, models, and principles of quantum machine learning and their role in next-generation intelligent systems.	--	--
CO2	**Analyze** quantum machine learning algorithms and hybrid quantum–classical models to understand their computational advantages and limitations.	PO1 (3)	PSO1 (3)
CO3	**Evaluate** quantum machine learning techniques and frameworks to assess their effectiveness for complex data-driven and optimization problems.	PO3 (2)	PSO2 (2)
CO4	**Design** quantum machine learning solutions by selecting appropriate quantum circuits, learning models, and hybrid architectures for real-world applications.	PO2 (1)	PSO1 (3)

CP25C14	**AI in IoT**	L	T	P	C
		3	0	0	3

Course Objective:

- To gain a deep understanding of AI concepts, IoT architectures, and their convergence for intelligent systems.
- To build and train machine learning and deep learning models suited for IoT data processing using TensorFlow / Keras.
- To apply data normalization, filtering, and feature engineering to enhance AI model accuracy
- To deploy AI models on edge devices for low-latency decision-making in IoT environments.
- To implement AI-based cyber security solutions for intrusion detection and anomaly detection in IoT networks.

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.

Activities:

- AI's integration into IoT ecosystems.
- Generate and analyze IoT sensor data in Jupyter Notebook.

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

Activities:

- Develop a real-time sensor pre-processing framework
- Analyse studies on AI-enhanced IoT data filtering.

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

Activities

- Design an AI model for real-time IoT data classification
- Mini-tutorials on Deep Learning Models for IoT

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

Activities:

- Build an AI-powered anomaly detection model**.**
- Pre-session assignments where students analyze recent research on Security for IoT Systems

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

Activities:

- case studies on Industry Applications of AI in IoT.
- Implement an industry-specific AI in IoT solution

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

Activities:

- case studies on AI Driven Smart cities for sustainable Urban development
- Implement a real time IoT based decision making system

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

Assessment Methodology: Assignments (30), Quiz (10), Virtual Demo (25), Flipped Class Room (10), Review of Gate and IES Questions (25)

References

1. Fadi Al-Turjman, “Artificial Intelligence in IoT”, Springer, 2019
2. Ambika Nagaraj,” The Role of AI in Enhancing IoT-Cloud Applications”, Bentham Books, 2023
3. Rajkumar Buyya and Amir Vahid Dastjerdi, "Internet of Things: Principles

and Paradigms", Elsevier, 2016
4. Jason Ioannou, "Data Analytics for IoT and Smart Systems", Springer, 2020
5. Ian Goodfellow, Yoshua Bengio, and Aaron Courville, "Deep Learning", MIT Press, 2016
6. Weisong Shi, "Edge Computing: Vision and Challenges", Springer, 2020
7. Mauro Conti, "Security and Privacy in the Age of IoT", Springer, 2020
8. François Chollet, “Deep Learning with Python” (2nd Edition), Manning Publications, 2021
9. Stuart Russell & Peter Norvig, “Artificial Intelligence: A Modern Approach (4th Edition), Pearson, 2020
10. Arshdeep Bahga & Vijay Madisetti, “Internet of Things: A Hands-On Approach” (2nd Edition), Universities Press, 2016

Web References

https://www.springer.com/gp/campaign/artificial-intelligence/artificial-intelligence-in-iot
https://iotworldmagazine.com/2025/01/16/2675/list-of-top-50-best-ai-automation-tools- in-2025
https://www.opensourceforu.com/2025/02/open-source-ai-frameworks-integrating-ai-with-iot/

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, architectures, and applications of integrating artificial intelligence techniques with Internet of Things (IoT) systems.	--	--
CO2	**Analyze** AI-enabled IoT systems to understand data processing, learning models, and decision-making mechanisms in distributed environments.	PO1 (3)	PSO1 (3)
CO3	**Evaluate** AI techniques and IoT platforms to assess their effectiveness, scalability, and reliability for intelligent connected applications.	PO3 (2)	PSO2 (2)
CO4	**Design** intelligent IoT solutions by selecting appropriate AI models, sensing strategies, and system architectures for real-world applications.	PO2 (1)	PSO1 (3)

CP25C15	Web 3.0	L	T	P	C
		3	0	0	3

Course Objective:

- To understand the progression from Web 1.0 and Web 2.0 to Web 3.0, with an emphasis on decentralization, semantic interoperability, and enhanced privacy.
- To gain hands-on expertise in blockchain technology, consensus algorithms, smart contract development, and dApp design by employing open standards.
- To deploy NFT tokenization models and digital identity systems in accordance with W3C standards.
- To develop the ability to integrate emerging technologies (IoT, AI, Semantic Web) while ensuring accessibility, security, and global interoperability.
- To critically assess decentralized governance models and regulatory challenges, proposing innovative solutions based on industry best practices.

Introduction: 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.

Activities:

- Presentations on key historical transitions and share related research papers before in-class discussions.

Blockchain Technology & Distributed Ledger Systems: 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.

Activities:

- Pre-record lectures on consensus mechanisms and present comparative analyses.
- Reproduce comparative studies on consensus mechanisms using available simulation tools.

Smart Contracts, dApps, Block chain Integration

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

Activities

- Team-based development of the dApp, with roles distributed for front-end, smart contract, and oracle integration.
- Mini-tutorials on dApps and present case studies of dApps projects in class.

Digital Asset Tokenization (NFTs) and Decentralized Storage: 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

Activities

- Team-based development of the NFT smart contract
- Mini-tutorials on Decentralized storage .

Security, Digital Identity, and Peer-to-Peer (P2P) Networking: Security – Scalability – Interoperability with traditional systems – Digital Identity – Decentralized Identity (DIDs) self-sovereign identity principles, and W3C DID and verifiable credentials standards.

Activities:

- Group projects to develop and test decentralized storage solutions and identity modules.
- Pre-session assignments where students analyze recent research on decentralized identity and storage.

Decentralized Governance & Tech Integration: 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.

Activities:

- Case studies on DAO governance and present summaries of regulatory research.
- Collaboration in groups to design and prototype DAO governance models and IoT integration solutions.

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

Assessment Methodology: Assignments (30), Quiz (10), Virtual Demo (25), Flipped Class Room (10), Review of Gate and IES Questions (25)

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”, Latest Edition, 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.

Web References

1. https://ipfs.tech/
2. https://docs.filecoin.io/
3. https://identity.foundation/
4. https://www.w3.org/TR/did-1.0/
5. https://www.aragon.org/

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, technologies, and architectural principles of Web 3.0 and their impact on next-generation internet applications.	--	--
CO2	**Analyze** Web 3.0 platforms and decentralized technologies to understand their functionality, performance, and security implications.	PO1 (3)	PSO1 (3)
CO3	**Evaluate** Web 3.0 technologies, protocols, and applications to assess their suitability for decentralized and user-centric systems.	PO3 (2)	PSO2 (2)
CO4	**Design** Web 3.0–based applications by selecting appropriate decentralized architectures, protocols, and technologies for real-world use cases.	PO2 (1)	PSO1 (3)

CP25C16	Advanced Large Language Models	L	T	P	C
		3	0	0	3

Course Objective:

This course provides the basic knowledge to understand and harness the capabilities of cutting-edge language models. Large Language Models (LLM) have revolutionized the way we interact with text, enabling us to communicate, analyse. and generate content with unprecedented sophistication. This course will help the students to learn LLM from the introduction to the uses and limitations, the inner workings of LLMs, equipping the students with thc tools to explore their versatile applications in content creation, code generation, sentiment analysis, and more.

Building Blocks of Modern LLM's: Large language model basics, Large Language model architecture, Transformer Architecture, Attention all you need – Encoder – Decoder

Suggested Activities

- Introduces the large language model basics and building blocks of language models.
- A detailed study about language architecture and transformer architecture.
- Analyze a real-world application for the Modern LLM.

Pre-Training Methods: 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

Suggested Activities

- Study the algorithm for Auto-Regressive and Auto-Encoder language modelling and implement the code to analyses the model flow.
- Implement the decoding strategies, scaling methods and experiment the same.

LLM Pretraining Methods: 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.

Suggested Activities

- Study the algorithm for LLM pretraining and implement the code to analyses the model flow.
- Implement the LM decoding strategies and experiment the same

Parameter Efficient Tuning Methods: 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.

Suggested Activities

Implement the LoRA methods for the standard dataset using the python code and analyze the results.

In-Context Learning, Prompt Engineering and Alignment: In-Context Learning, Fine-tuning - 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.

Suggested Activities

- Learning the zero- shot learning, few-shot learning and prompt selection
- Build the question and answering systems
- Sentiment analysis using Multi-Language dataset

Generation Based Automatic Evaluation Methods: 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.

Suggested Activities

- Analyze the GLUE-GEM score for the training and testing dataset.

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

Assessment Methodology: Assignments (30), Quiz (10), Virtual Demo (25), Flipped Class Room (10), Review of Gate and IES Questions (25)

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

E-Resources

1. NPTEL Course: Introduction to Large Language Modelling
2. https://github.com/HandsOnLLM/Hands-On-Large-Language-Models

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, architectures, and capabilities of advanced large language models and their applications in intelligent systems.	--	--
CO2	**Analyze** large language model architectures and training methodologies to understand their performance, scalability, and limitations.	PO1 (3)	PSO1 (3)
CO3	**Evaluate** large language models and deployment strategies to assess their effectiveness, ethical considerations, and applicability in real-world scenarios.	PO3 (2)	PSO2 (2)
CO4	**Design** applications and systems leveraging advanced large language models by selecting appropriate architectures, fine-tuning strategies, and deployment frameworks.	PO2 (1)	PSO1 (3)

CP25C18	**Green Computing and Sustainability**	L	T	P	C
		3	0	0	3

Course Objective:

- To understand the principles of green computing.
- To assess environmental impacts of computing technologies and need for green computing.
- To explore sustainable IT practices and energy-efficient hardware system for sustainability growth.

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.

Activities: Review: E-waste management.

Green Hardware: Life cycle of hardware - Reuse, Recycle and Dispose Green Design and Manufacturing of Computers- Green Software - Energy-Saving Software Techniques

Activities: Demo: DAQ tools

Sustainable Software Development: attributes – metrics – Methodology - Energy-aware Software Engineering - Green Algorithms

Activities: Case study: Project's Carbon Footprint

Green Data Center: Infrastructure for Energy Efficiency – server power management

Activities: Poster presentation: Energy issues in Data Centers

Green Data Storage: Practices for Improving the Energy Efficiency of Data Storage - Optical Interconnects for Green Computers and Data Centers **Activities:** Case study: Energy-efficient cloud computing
Managing Green IT: Strategizing Green Initiatives – Implementation - Green IT Laws, Standards and Protocols- Regulatory Environment and IT Manufacturers **Activities:** Demo: Green IT Laws in India. Debate: Role of IT professional in Green Computing.
Nonregulatory Government Initiatives: Green Building Standards - Sustainability Reporting and Green Audits **Activities:** Perform Energy audit in computer lab using software tools.
Weightage: Continuous Assessment: 40%, End Semester Theory Examinations: 60%
Assessment Methodology: Assignments (30), Quiz (10), Virtual Demo (25), Flipped Class Room (10), Review of Gate and IES Questions (25)
References 1. San Murugesan, Harnessing Green IT: Principles and Practices, Wiley, 2012. 2. Suyel Namasudra(Editor), Green and Sustainable Computing, 1st Edition, Volume 87, Elsevier, 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, Seifedine Kadry, 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

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, principles, and practices of green computing and sustainability in modern information technology systems.	--	--
CO2	**Analyze** computing systems and technologies to assess their energy efficiency, environmental impact, and sustainability metrics.	PO1 (3)	PSO1 (3)

CO3	**Evaluate** green computing strategies, sustainable IT practices, and regulatory frameworks for their effectiveness in reducing environmental impact.	PO3 (2)	PSO2 (2)
CO4	**Design** sustainable computing solutions by selecting energy-efficient architectures, technologies, and practices for environmentally responsible systems.	PO2 (1)	PSO1 (3)

CP25C19	**Cognitive Computing**	L	T	P	C
		3	0	0	3

Course Objectives:

- 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.
- Develop cognitive computing applications.

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.

Activities:
1. Concept Quizzes: Human cognition and Cognitive Computing
2. Research paper review on cognitive systems

Cognitive architectures: ACT-R, Soar, OpenCog - Knowledge Representation and Reasoning - Machine Learning models in cognitive systems - Human-like Learning: Supervised, Unsupervised, Reinforcement

Activities: Demo on IBM Watson APIs (Language Translator)

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

Activities:
Ideathon: Innovative solution to health care problems.

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 **Activities:** Mini project: Develop health care chatbot. Prototype Demonstrations.
Process of Building a Cognitive Application - IBM's Watson as a Cognitive System-Business Implications of Cognitive Computing **Activities:** Research Paper Review: Cognitive Computing
Cognitive Computing Applications: Healthcare, Finance, Education, Legal, Robotics-Building a Cognitive Healthcare Application - Emerging Cognitive Computing Areas - Cognitive cloud- Cognitive IoT - Cognitive Chatbots - Neuromorphic computing and brain-inspired models. **Activities:** 1. Prototype Demonstrations. 2. Project demo and peer review.
Weightage: Continuous Assessment: 40%, End Semester Theory Examinations: 60%
Assessment Methodology: Assignments (30), Quiz (10), Virtual Demo (25), Flipped Class Room (10), Review of Gate and IES Questions (25)
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. 6. Fuchun Sun, Qinghu Meng, Zhumu Fu, Bin Fang (Editors), Communications in Computer and Information Science (CCIS, volume 1918) – Springer Series, 2024.

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, architectures, and paradigms of cognitive computing and their role in intelligent decision-making systems.	--	--
CO2	**Analyze** cognitive computing models and learning	PO1	PSO1

	mechanisms to understand perception, reasoning, and knowledge representation in intelligent systems.	(3)	(3)
CO3	**Evaluate** cognitive computing techniques, frameworks, and applications to assess their effectiveness in solving complex real-world problems.	PO3 (2)	PSO2 (2)
CO4	**Design** cognitive computing solutions by selecting appropriate models, algorithms, and system architectures for human-centric intelligent applications.	PO2 (1)	PSO1 (3)

CP25C20	**Agentic AI**	L	T	P	C
		3	0	0	3

Course Objectives:

- Understand the foundations and architecture of intelligent agents.
- Design agents capable of autonomous and rational decision-making.
- Apply reinforcement learning for adaptive agent behavior.
- Explore coordination in multi-agent and human-AI systems.
- Analyse the ethical implications of agentic systems.
- Develop real-world applications using modern agent-based AI frameworks.

Foundations of Intelligent Agents

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

Activity:

- Report preparation about types of Agents, role and architecture of each agent

Agent Decision-Making and Planning

Rationality and Utility Theory - Task Environment Analysis - AI Planning Techniques: STRIPS, Classical Planning - Decision-Making under Uncertainty: MDPs, Bayesian Models, Game Theory

Activity:

- Design a simple application for making decision using AI in python

Learning in Agentic Systems

Reinforcement Learning: Q-Learning, SARSA - Value and Policy Iteration - Deep RL: DQNs, Policy Gradient Methods, Actor-Critic - Tools: OpenAI Gym, PettingZoo, TensorFlow RL

Activity:

- Write a case study for Tools about OpenAI and TensorFlow with proper application

Multi-Agent Systems and Collaboration

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 NettLogo platform

LLM Agents and Embodied Cognition

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

Activity:

- Design a simple application based on LLM-based tool-using agents

Ethics and Applications of Agentic AI

AI Alignment, Fairness, Explainability - Social and Environmental Implications - Case Studies: Autonomous Vehicles, Conversational Agents - Course Project Evaluation and Presentations

Activity:

- Agent-based simulations for real-world problems

References:

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

Tools & Technologies:
Languages: Python
Libraries: PyTorch, TensorFlow, OpenAI Gym, LangChain
Frameworks: PettingZoo, Unity ML-Agents, NetLogo, Hugging Face
Platforms: Jupyter, VS Code, RoboCup Simulators

Weightage:	**Continuous Assessment: 40%**	**End Semester Theory Examinations: 60%**
	(i) Activity: 10% **(ii) Internal Theory Examination: 30%**	
Mandated Activities with Marks:		

Assignments (30), Quiz (10), Virtual Demo (25), Flipped Class Room (10), Review of Gate and IES Questions (25)
Internal Examinations: Two Tests

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, architectures, and principles of agentic artificial intelligence and autonomous agent systems.	--	--
CO2	**Analyze** agentic AI models and decision-making mechanisms to understand autonomy, goal-directed behavior, and interaction in intelligent agents.	PO1 (3)	PSO1 (3)
CO3	**Evaluate** agentic AI frameworks, learning strategies, and coordination techniques to assess their effectiveness in complex and dynamic environments.	PO3 (2)	PSO2 (2)
CO4	**Design** agentic AI solutions by selecting appropriate agent architectures, learning methods, and coordination mechanisms for real-world applications.	PO2 (1)	PSO1 (3)

		L	T	P	C
CP25C21	**Mixed Reality**	3	0	0	3

Course Objective:

This course to define mixed reality and differentiate between virtual reality and augmented reality. It is used to describe the core ideas behind holograms and start designing and developing 3D applications. It also used to understand what mixed reality offers in real-world apps.

Defining Virtual and Augmented Reality

Four Key Elements of Virtual Reality Experience Looking at Some Other Types of Virtual and Augmented Reality.

Activity:

- Assignment about Virtual Reality Experience for real world applications

Exploring the Current State of Virtual 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.

Activity:

- Create a flipped class room for exploring the current state of Virtual Reality

Exploring the Current State of Augmented Reality Mobile devices AR headsets. AR glasses. Hand tracking. Motion controllers Hand and Gesture Tracking, Whole Body Tracking **Activity:** • Create a flipped class room for exploring the current state of Augmented Reality
Simple VR Environment Development 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. **Activity:** • Create a Simple Virtual Environment for a simple real - world application
Creating 2D/3DProjects 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 **Activity:** • Create a simple 3D game using Visual Scripting
Setting Up for AR Development 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. **Activity:** • Write a case study about setting up for AR development
References: 1. Jonathan Lenowes, Augmented Reality with Unity AR Foundation,2021 Packt Publishing 2. William R. Sherman, Alan B. Craig, Understanding-Virtual-Reality-Interface-Application-and-Design-The-Morgan-Kaufmann-Series-in-Computer-Graphics 3. Virtual &Augmented Reality for Dummies by Paul Mealy, 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

Weightage:	**Continuous Assessment: 40%** **(i) Activity: 10%** **(ii) Internal Theory Examination: 30%**	**End Semester Theory Examinations: 60%**
Mandated Activities with Marks: Assignments (30), Quiz (10), Virtual Demo (25), Flipped Class Room (10), Review of Gate and IES Questions (25)		
Internal Examinations: Two Tests		

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, technologies, and components of mixed reality and their applications in immersive digital environments.	--	--
CO2	**Analyze** mixed reality systems and interaction techniques to understand performance, usability, and user experience in immersive applications.	PO1 (3)	PSO1 (3)
CO3	**Evaluate** mixed reality platforms, devices, and development frameworks to assess their suitability for various application domains.	PO3 (2)	PSO2 (2)
CO4	**Design** mixed reality applications by selecting appropriate hardware, software frameworks, and interaction models for real-world use cases.	PO2 (1)	PSO1 (3)

CP25C22	Blockchains Architecture and Design	L	T	P	C
		3	0	0	3

Course Objective:
The objective of this course is to study the Blockchain Technology and its uses across a range of fields, with an emphasis on Blockchain fundamentals, Smart contracts, Consensus processes and Security issues.

Introduction to Blockchain

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

Activity:
Quiz on core Blockchain concepts like hashing, signing data and building Merkle tree

Bitcoin and Consensus Mechanisms
Bitcoin Overview – Bitcoin Scripts – Bitcoin Transactions – Bitcoin Network and Payments – Consensus mechanism – Types of Consensus mechanism
Activity:
Debates and Discussion on the pros and cons of various consensus protocols
Permissioned & Permissionless Protocols
Proof of Work (PoW) – Consensus Protocols for Permissionless Blockchain Environment – Consensus Protocols for Permissioned Blockchain Environment.
Activity:
Identify and set up Ethereum testnet – Transfer ethers between accounts
Smart Contracts and Ethereum
Smart Contract – Definition – Ricardian Contracts – Introduction to Ethereum – Ethereum Network – Components of the Ethereum – Ethereum Virtual Machine (EVM) – Ethereum Development Environment – Solidity Language
Activity:
Presentation prepared on Smart Contract and Ethereum concepts from research papers
Web3 And Hyperledger Fabric
Exploring Web3 with Geth – Contract Deployment – Interacting with contracts via frontends – Truffle – Introduction to Hyperledger – Reference Architecture – Hyperledger Fabric – Hyperledger Sawtooth – Corda
Activity:
Explore and Discuss research papers related to Hyperledger and Blockchain technology
Blockchain Applications
Blockchain for Government: Border control – Voting – Citizen Identification – Blockchain for Health Industry – Blockchain for Finance: Insurance – Post Trade Settlement – Financial Crime Prevention
Activity:

Research and present use cases like supply chain, digital identity, etc.

Challenges And Future Directions

Scalability – Privacy – Emerging Trends – Limitations of Current Blockchain Platforms – Open Research Challenges

Activity:

Present mini-projects on performance optimization or privacy enhancement in Blockchain systems

Weightage:	**Continuous Assessment: 40%** **(i) Activity: 10%** **(ii) Internal Theory Examination: 30%**	**End Semester Theory Examinations: 60%**
Mandated Activities with Marks: Assignments (30), Quiz (10), Virtual Demo (25), Flipped Class Room (10), Review of Gate and IES Questions (25)		
Internal Examinations: Two Tests		

References:

1. Imran Bashir, "Mastering Blockchain: A deep dive into distributed ledgers, consensus protocols, smart contracts, Dapps, cryptocurrencies, Ethereum,and more", Third Edition, Packt Publishing, 2020.
2. Andreas M. Antonopoulos, David A. Harding, "Mastering Bitcoin: Programming the open Blockchain", Third Edition, O'Reilly Publishing, 2023.
3. Ahmed Banafa, "Blockchain Technology and Applications", First Edition, River Publishers, 2024.
4. Melanie Swan, "Blockchain: Blueprint for New Economy", First Edition, O'Reilly Publishing, 2015.
5. Bikramaditya Singhal, Gautam Dhameja, Priyansu Sekhar Panda, "Beginning

6. Blockchain : A Beginner's Guide to Building Blockchain Solutions" , 1st edition, Apress, NewYork, 2018.

E-Resources:

1. Zero to Blockchain - An IBM Redbooks course, by Bob Dill, David Smits https://www.redbooks.ibm.com/Redbooks.nsf/RedbookAbstracts/crse0401.html
2. Hyperledger Fabric - https://www.hyperledger.org/projects/fabric
3. https://www.tutorialspoint.com/blockchain/index.htm
4. https://www.nptel.ac.in/courses/106105184 by Prof.S.Chakraborty and P.Jayachandran,IIT Kharagpur

CO–PO–PSO Mapping:

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, components, and operational principles of blockchain architectures and distributed ledger technologies.	--	--
CO2	**Analyze** blockchain consensus mechanisms, data structures, and network architectures to understand security, performance, and scalability issues.	PO1 (3)	PSO1 (3)
CO3	**Evaluate** blockchain platforms, smart contract frameworks, and architectural designs to assess their suitability for various decentralized applications.	PO3 (2)	PSO2 (2)
CO4	**Design** blockchain-based systems by selecting appropriate architectures, consensus protocols, and smart contract models for real-world use cases.	PO2 (1)	PSO1 (3)

CP25C23	Human - Centered AI	L	T	P	C
		3	0	0	3

Course Objective:
The course aims to introduce the fundamentals of Human-Centered Artificial Intelligence (HCAI), focusing on its principles, frameworks, and societal impact. This course includes fairness, privacy, reliability, and ethical considerations in the development of intelligent systems. Additionally, it explores governance structures, best practices, and regulatory measures to ensure safe and trustworthy AI deployment.

Introduction To Human-Centered AI
Overview of AI development - Defining Human-Centered AI (HCAI) - Difference between traditional AI and HCAI - Process of design thinking - Cognitive models

Activity

- Explore the key innovations and challenges involved in human-AI collaboration across different sectors.

Human Factors and AI
People and Computers: Are They in the Same Category? - HCAI principles - Impact of AI on society - Importance of human values

Activity

- Explore the impact of AI on ethics and human centric approaches.

Fairness and Bias

Sources of bias, Group fairness-Exploratory data analysis, limitation of a dataset - Group fairness and Individual fairness-Counterfactual fairness

Activity

- Use AI platforms to detect and analyze various types of bias

Privacy Preservation:

Attack models-Privacy: Preserving Learning-Differential privacy-Fedrated learning

Activity

- Explore why a coin flip is used in sensitive surveys to protect privacy

HCAI Framework
Levels of automation - Defining reliable, safe, and trustworthy systems - Two-dimensional HCAI framework: life-critical systems - Golden rules for design - Application examples

Activity

- Prepare a presentation on Human-Centered AI: Model Explainability with

LIME and SHAP

Design Metaphor

Science and innovation goals - Intelligent agents - Tele-bots - Assured autonomy: design, control centers - Social robots: active appliances - Enhancing human capabilities

Activity

- Survey and analyze recent research papers emphasizing Human-Centered AI (HCAI)

HCAI Governance Structures

Ethical implications of emerging AI technologies - Software engineering best practices - Establishing safety cultures in organizations - Independent oversight and certification - Role of government regulation

Activity

- Explore national policies, ethical guidelines, and legal frameworks aimed at ensuring the safe and responsible use of Artificial Intelligence across the world

Weightage:	**Continuous Assessment: 40%**	**End Semester Theory Examinations: 60%**
	(i) Activity: 10% **(ii) Internal Theory Examination: 30%**	
Mandated Activities with Marks: Assignments (30), Quiz (10), Virtual Demo (25), Flipped Class Room (10), Review of Gate and IES Questions (25)		
Internal Examinations: Two Tests		

References:

1. **Human-Centered AI: A New Synthesis**, Ben Shneiderman, 1st Edition, Oxford University Press, 2022.
2. **Responsible Artificial Intelligence: How to Develop and Use AI in a Responsible Way**, Virginia Dignum, Springer Nature, 2019.
3. **Interpretable Machine Learning**, Christoph Molnar, 1st Edition, Lulu, 2019. [eBook, Available Online]
4. **Designing the User Interface: Strategies for Effective Human-Computer Interaction**, Ben Shneiderman, Catherine Plaisant, Maxine Cohen, Steven Jacobs, Niklas Elmqvist, 6th Edition, Pearson Education, 2018.
5. **Interaction Design: Beyond Human-Computer Interaction**, Jenny Preece, Helen Sharp, Yvonne Rogers, 5th Edition, Wiley Student Edition, Wiley, 2019.
6. **Human-Computer Interaction**, Alan Dix, Janet Finlay, Gregory Abowd, Russell Beale, 3rd Edition, Pearson Education, 2004.

7. **AI Ethics**, Paula Boddington, Springer, 2023.
8. **The Ethics of Artificial Intelligence: Principles, Challenges, and Opportunities**, Luciano Floridi, Springer, 2023.

E-Resources:

1. NPTEL Video Lecture Notes on “Responsible & Safe AI Systems”
2. https://aclanthology.org/2021.acl-long.330.pdf
3. https://arixv.org/pdf/2004.09456

CO-PO-PSO Mapping:

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, principles, and design philosophies of human-centered artificial intelligence for developing responsible and user-focused AI systems.	--	--
CO2	**Analyze** human-AI interaction models and ethical considerations to understand usability, transparency, and trust in intelligent systems.	PO1 (3)	PSO1 (3)
CO3	**Evaluate** human-centered AI approaches, frameworks, and applications to assess their effectiveness in enhancing user experience and societal impact.	PO3 (2)	PSO2 (2)
CO4	**Design** human-centered AI solutions by selecting appropriate interaction techniques, ethical guidelines, and evaluation strategies for real-world applications.	PO2 (1)	PSO1 (3)

CP25C24	Vibe Coding	L	T	P	C
		3	0	0	3

Course Objective:
This course introduces the basics of AI-assisted coding and Vibe Coding. Participants will learn to use AI tools, build simple applications, and understand the roles and ethics involved. By the end, they will create their own AI-powered projects.

Introduction To Vibe Coding and Human-Ai Collaboration

Introduction to Vibe Coding-Basic Prompting Techniques for Coding Tasks-Roles and -Responsibilities in Human-AI Collaboration-AI Assistant-Prompt Engineer-Developer-Designing Simple AI-Augmented Workflows-Interaction Between AI-Generated Code and Developer Feedback

Activities:

- Students take on roles such as AI, prompt engineer, and developer to simulate collaborative coding.
- Students learn prompting basics at home and participate in a class quiz and discussion.

Tools And Platforms for Ai-Assisted Coding

Overview of AI-Assisted Coding Tools-Cursor, v0, Bolt, etc.-Features and Use Cases of Each Tool-Setting Up and Navigating AI-Coding Platforms-Building Simple Web Applications with AI -Integration-Workflow Prototyping Using AI Tools

Activities:

- Students explore AI tools (e.g., Cursor or v0) at home and build a basic app in class.
- Groups design a simple development workflow using AI tools and explain tool selection and purpose.

AI Prompt Design, Oversight, and Ethical Coding Practices
6Vibe Coder’s Toolkit: Prompts, Context, and Flow-Prompt Optimization Techniques-Human Oversight in AI Coding Systems-Ethical and Responsible AI Usage-Challenges and Best Practices in AI-Augmented Development

Activities:

- Team presents a complete AI workflow, including prompt design, context handling, and oversight strategy.
- Class discussion on accountability, code quality, and risks of over-reliance on AI in software development.

The Unbundled Programmer and the Future of Coding Roles Evolving Developer Roles in the AI Era-The Shift from Coding to Orchestration and Strategy-AI Integration Across the Software Development Lifecycle-Human-AI Collaboration in Debug, Design, and Deployment-Strategic Thinking and Adaptability in AI-Augmented Teams **Activities** • Showcase how AI transforms developer roles across the SDLC. • Simulate a modern dev team using AI tools and complete a quiz based on workflow observations.
Code Quality, Reliability, and Ethical Development 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 Vibe Code-Ethical Coding Practices in AI-Augmented Environments-Navigating Cultural Shifts in 'Vibe Coding' and Accountability **Activities:** • Refactor example AI-generated code and answer related questions on quality and security. • Study ethical case studies at home and present findings in class.
Applied Projects and Real-World Ai Applications AI-Powered Task Manager and To-Do Lists-Resume Builder with Smart AI Suggestions-Note-Taking App with Auto-Organization-Recipe Generator with Ingredient Substitution Logic-Visual Memory Game Using AI Tools-Non-diagnostic Health Symptom Checker **Activities** • Build a working prototype (e.g., to-do list, resume builder) using AI-assisted tools. • Present the final project and complete a quiz on the design and development process.
Advanced Ai-Driven Web Application Deployment Planning and Scoping AI-Assisted Applications-Workflow Mapping from Idea to Deployment-Integrating Front-End, Back-End, and AI Logic-Testing and Deploying AI-Augmented Applications-Human Oversight and Continuous Improvement Loops-Project Ownership and Accountability in AI Codebases **Activities**

- Design and deploy a full AI-assisted web application as a team.
- Teams demo their solutions and reflect on human-AI collaboration challenges.

Weightage:	**Continuous Assessment: 40%**	**End Semester Theory Examinations: 60%**
	(i) Activity: 10% **(ii) Internal Theory Examination: 30%**	
Mandated Activities with Marks: Assignments (30), Quiz (10), Virtual Demo (25), Flipped Class Room (10), Review of Gate and IES Questions (25)		
Internal Examinations: Two Tests		

References:

1. Ted Winston, *Mastering Vibe Coding: Build, Debug, and Ship Software with AI Assistants like Cursor, Replit, and GPT*, Kindle Edition, April 25, 2025.
2. Ethan Voss, *Vibe Coding: Build & Sell Apps Without Coding Experience: How to Use AI to Create Profitable Apps Even If You've Never Written a Line of Code*, Kindle Edition, 26 March 2025.
3. Amit Iyer, *Beginner's Guide to Vibe Coding: Exploring Features, Use Cases and Understanding Vibe Coding*, Paperback, 19 March 2025.
4. Gene Kim, Steve Yegge, *Vibe Coding: Building Production-grade Software With GenAI, Chat, Agents, and Beyond*, Paperback, 21 October 2025.
5. Addy Osmani, *Beyond Vibe Coding*, O'Reilly Media, Inc., August 2025. ISBN: 9798341634756

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, philosophy, and practices of vibe coding and its role in modern, collaborative, and creative software development.	--	--
CO2	**Analyze** vibe coding workflows, tools, and collaborative practices to understand productivity, creativity, and developer experience.	PO1 (3)	PSO1 (3)
CO3	**Evaluate** vibe coding approaches, platforms, and community-driven practices to assess their effectiveness in real-world software development scenarios.	PO3 (2)	PSO2 (2)
CO4	**Design** software development solutions using vibe coding principles by selecting appropriate tools, collaboration strategies, and creative workflows.	PO2 (1)	PSO1 (3)

CP25C25	Federated Learning	L	T	P	C
		3	0	0	3

Course Objective:

1. Understand the fundamentals, architecture, and real-world relevance of Federated Learning.
2. Learn core FL algorithms, privacy techniques, and system challenges in decentralized settings.
3. Apply FL frameworks and tools to build practical applications through hands-on project work.

Introduction to Federated Learning

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.

Activity

Create a Quiz for understanding Federated Learning Basics

Federated Learning Algorithms

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.

Activity

Create a flipped class room for Understanding federated learning algorithms

Privacy and Security in FL

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.

Activity

Ask students to submit an assignment on Privacy and security in FL

System & Communication Challenges

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.

Activity
Create a flipped class room communication challenges

Applications and Frameworks
FL Applications in Healthcare, Finance, Smart City, Recommender Systems – Case Studies – Tools and Frameworks: TensorFlow Federated, PySyft, Flower – FL Simulators

Activity
Ask students to submit an Assignment on applications of FL

Project Work and Future Directions
Project Work: Building a Simple Federated Learning Model – Model Deployment and Evaluation – Future Directions: Split Learning, Hybrid FL, Federated Analytics, FL + AI/ML

Activity
Ask students to submit simple project on FL

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 ed., 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.

Weightage:	**Continuous Assessment: 40%**	**End Semester Theory Examinations: 60%**
	(i) Activity: 10% **(ii) Internal Theory Examination: 30%**	
Mandated Activities with Marks: Assignments (30), Quiz (10), Virtual Demo (25), Flipped Class Room (10), Review of Gate and IES Questions (25)		
Internal Examinations: Two Tests		

E-resources:
1. https://youtu.be/E3Kn5v-vdUc?si=UMnGqSNOB9NjheTS- NPTEL YouTube– Lecture 42: Introduction to Federated Learning
2. https://github.com/adap/flower- Flower Framework GitHub
3. https://github.com/OpenMined/PySyft- PySyft (OpenMined)

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, architectures, and operational principles of federated learning and privacy-preserving machine learning systems.	--	--
CO2	**Analyze** federated learning algorithms and communication strategies to understand model convergence, privacy, and system performance.	PO1 (3)	PSO1 (3)
CO3	**Evaluate** federated learning frameworks, security mechanisms, and deployment approaches to assess their effectiveness in distributed and sensitive data environments.	PO3 (2)	PSO2 (2)
CO4	**Design** federated learning solutions by selecting appropriate aggregation methods, communication protocols, and privacy techniques for real-world applications.	PO2 (1)	PSO1 (3)

CP25002	**Deep Learning for Computer Vision**	L	T	P	C
		3	0	0	3

Course Objective:

This course provides the automatic analysis and understanding of images and videos and Computer Vision. It occupies significant importance in applications including security, healthcare, entertainment, mobility, etc. The recent success of deep learning methods has revolutionized the field of computer vision, making new developments increasingly closer to deployment that benefits end users.

Introduction to Image Fundamentals

Introduction to Image Formation – Capture and Representation – Linear Filtering – Correlation – Convolution – Edge – Blobs – Corner Detection – Scale Space and Scale Selection – SIFT – SURF– HoG – LBP

Activity:

Ask the student to present the seminar on the Image Features

Introduction to Visual Matching and Deep Learning

Bag-of-words – VLAD – RANSAC – Hough transform – Pyramid Matching – Optical Flow – Review of Deep Learning – Multi-layer Perceptron – Backpropagation

Activity:

Create a Quiz to discuss about the basics of Deep Learning

Convolutional Neural Networks (CNN) 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) **Activity:** Ask student to submit assignments on principle behind Hallucination
Applications of CNN 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 **Activity:** Create a Discussion to differentiate R-CNN, Fast R-CNN and Faster R-CNN
Recurrent Neural Networks (RNN) and Attention Models Introduction to RNNs – CNN and RNN Models for Video Understanding: Spatio-temporal Models, Action/Activity Recognition – Introduction to Attention Models in Vision – Vision and Language: Image Captioning – Visual QA – Visual Dialog – Spatial Transformers – Transformer Networks **Activity:** Ask students to prepare a presentation about Transformers.
Deep Generative Models - Applications and variants of Generative Models in Vision 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 **Activity:** Create a flipped class room for Applications of Generative Models
Recent Trends of Deep Learning in Computer Vision Zero-shot – One-shot – Few-shot Learning – Self-supervised Learning – Reinforcement Learning in Vision – Applications **Activity:** Design a Simple Project on real world application of computer vision using deep learning algorithm

Weightage:	Continuous Assessment: 60%	End Semester Theory Examinations: 40%
	(i) Activity: 15% (ii) Internal Theory Examination: 35% (iii) Internal Laboratory Examinations: 15%	
Mandated Activities with Marks: Assignments (30), Quiz (10), Virtual Demo (25), Flipped Class Room (10), Review of Gate and IES Questions (25)		
Internal Examinations: Two Tests		

References:

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

E-resources:

1. https://opencv.org/blog/deep-learning-with-computer-vision/
2. https://onlinecourses.nptel.ac.in/noc25_cs93/preview
3. https://cs231n.stanford.edu/

CO-PO Mapping

CO	Description of CO	PO	PSO
CO1	Describe the fundamental concepts, architectures, and techniques of deep learning used for computer vision and image understanding tasks.	--	--
CO2	**Analyze** deep learning–based computer vision models to understand feature learning, performance metrics, and optimization challenges.	PO1 (3)	PSO1 (3)
CO3	**Evaluate** computer vision algorithms and deep learning architectures to assess their effectiveness for various visual recognition and analysis applications.	PO3 (2)	PSO2 (2)
CO4	**Design** computer vision solutions by selecting and integrating suitable deep learning models, datasets, and training strategies for real-world vision problems.	PO2 (1)	PSO1 (3)