MOUNT ZION

COLLEGE OF ENGINEERING AND TECHNOLOGY

(AUTONOMOUS)

(Approved by AICTE, Affiliated to Anna University & Accredited by NAAC with A+ Grade)
Lena Vilakku, Pilivalam P.O., Thirumayam TK., Pudukkottai - 622 507
E-mail: info@mzcet.in | www.mzcet.in

Curriculum and Syllabus Handbook

M.E – Computer Science and Engineering [Regulation 2025]

[CHOICE BASED CREDIT SYSTEM]

Approved by Academic Council on 24.06.2026

INSTITUTE VISION AND MISSION	
VISION	To be the preferred choice of students who aspire to become engineers of the highest calibre.
MISSION	• Provide world-class physical and digital infrastructure and learning environment. • Develop the competencies of students to make them job-ready and entrepreneurs. • Facilitate mental, physical, emotional and spiritual development of students and ensure their holistic development. • Preserve the environment through the implementation of eco-friendly and sustainable practices. • Serve the community through skill development and other need-based services. • Establish sustainable partnerships with industries and R&D laboratories for mutual benefit.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING	
VISION	To be a leader in computer science and engineering education by empowering students with the knowledge, skills, and values needed to excel in technology, innovation, research, and service to society while fostering creativity, ethical responsibility, and a commitment to lifelong learning.
MISSION	• To provide a robust foundation in computer science and engineering through a curriculum that blends theoretical knowledge, practical expertise, and emerging technologies. • To promote a culture of innovation and research by encouraging students and faculty to explore advanced technologies and address real-world challenges. • To nurture leadership, teamwork, and ethical responsibility, equipping students to face global challenges and thrive in diverse professional environments. • To forge strong partnerships with industry, academia, and society, enriching learning experiences, promoting entrepreneurship, and addressing societal needs through technological solutions.

PROGRAMME EDUCATIONAL OBJECTIVES (PEO)	The graduates of Computer Science and Engineering will be **PEO1:** Graduates will excel in their careers by applying technical expertise, problem-solving skills, and a commitment to lifelong learning to address challenges in computer science and engineering. **PEO2:** Graduates will demonstrate leadership, teamwork, and ethical values in their professional roles, contributing to the success of organizations and the betterment of society **PEO3:** Graduates will develop innovative, sustainable, and socially responsible solutions using their knowledge of computer science and engineering to address real-world challenges and advance technology.
PROGRAMME SPECIFIC OUTCOMES (PSO)	Students will be able to **PSO1:** Apply the principles of computer science and engineering to design, develop, and analyse software systems, utilizing programming languages, algorithms, and computing platforms effectively to solve real-world problems. **PSO2:** Innovate and apply cutting-edge technologies such as artificial intelligence, machine learning, and data science to tackle complex, interdisciplinary problems, enhancing system efficiency and functionality. **PSO3:** Demonstrate strong professional communication, teamwork, and ethical decision-making skills, enabling them to work effectively in collaborative environments and contribute responsibly to the development of technology solutions.

PROGRAM OUTCOMES:

At the end of this programme the students will be able to

PO	GRADUATE ATTRIBUTES	PROGRAMME OUTCOMES
PO1	Engineering Knowledge	Apply math, science, and engineering fundamentals to complex problems.
PO2	Problem Analysis	Identify and analyze complex problems using research and sustainability principles.
PO3	Design Solutions	Design systems and processes considering health, safety, cost, culture, and environment.
PO4	Investigations	Use experiments, modelling, and data analysis to reach valid conclusions.
PO5	Engineering Tools	Apply modern tools for modelling and problem-solving, recognizing their limits.
PO6	Society & Environment	Assess societal, legal, and environmental impacts of engineering solutions.
PO7	Ethics	Commit to ethics, human values, diversity, and legal compliance.
PO8	Teamwork	Work effectively as an individual and in multidisciplinary teams.
PO9	Communication	Communicate clearly in reports, presentations, and documentation across diverse groups.
PO10	Management & Finance	Apply management and economic principles in projects and teamwork.
PO11	Lifelong Learning	Engage in continuous learning, adapt to new technologies, and think critically.

MAPPING OF PROGRAMME EDUCATIONAL OBJECTIVES WITH PROGRAMME OUTCOMES

PEO	PROGRAMME OUTCOMES											
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1	3	3	2	2	2				1	2		3
2			1	1		2	1	3	3	3	2	2
3	2	2	3	2	3	2	3	2	2	2	1	2

MAPPING OF PROGRAMME SPECIFIC OBJECTIVES WITH PROGRAMME OUTCOMES

PSO	PROGRAMME OUTCOMES											
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
1	3	3	3	2	3				1	2		2
2	2	3	3	3	3	1	1		1	2	1	2
3			1	1		2	1	3	3	3	2	2

3: High Correlation 2: Average Correlation 1: Low Correlation

MOUNT ZION COLLEGE OF ENGINEERING AND TECHNOLOGY
AUTONOMOUS
REGULATIONS 2025
CHOICE BASED CREDIT SYSTEM
M.E. COMPUTER SCIENCE AND ENGINEERING
CURRICULUM

SEMESTER – I									
S.NO	**COURSE CODE**	**COURSE TITLE**	**COURSE TYPE**	**CATEGORY**	**PERIODS PER WEEK**			**TOTAL CONTACT PERIODS**	**C**
					L	**T**	**P**		
THEORY AND LAB INTERGATED THEORY									
1.	MA25C07	Advanced Mathematical Methods (CSIE)	T	BS	3	1	0	4	4
2.	CP25C01	Advanced Data Structures and Algorithms	LIT	ES (PC)	3	0	4	7	5
3.	CP25C02	Advanced Database Technologies	T	ES (PC)	3	0	0	3	3
4.	CP25C03	Advanced Operating Systems	T	ES (PC)	3	0	0	3	3
5.	CP25C04	Advanced Compiler Design	T	ES (PC)	3	0	0	3	3
6.	CP25101	Technical Seminar	-	SD	0	0	2	2	1
Total Credits								**22**	**19**

SEMESTER – II									
S.NO	**COURSE CODE**	**COURSE TITLE**	**COURSE TYPE**	**CATEGORY**	**PERIODS PER WEEK**			**TOTAL CONTACT PERIODS**	**C**
					L	**T**	**P**		
THEORY AND LAB INTERGATED THEORY									
1.	CP25201	Multicore Architectures	LIT	ES (PC)	3	0	2	5	4
2.	CP25C05	Artificial Intelligence and Machine Learning	T	ES (PC)	3	0	0	3	3
3.	CP25C06	Cloud and Big Data Analytics	T	ES (PC)	3	0	0	3	3
4.	CP25C07	Quantum Computing	T	ES (PC)	2	0	0	2	2
5.	***	Programme Elective I	T	ES (PE)	3	0	0	3	3
6.	***	Industry Oriented Course I	-	SD	1	0	0	1	1
7.	CP25202	Industrial Training	-	SD	-	-	-	-	2
8.	***	Self-Learning Course	-	-	-	-	-	-	1
TOTAL CREDITS								**17**	**19**

SEMESTER – III									
S.NO	**COURSE CODE**	**COURSE TITLE**	**COURSE TYPE**	**CATEGORY**	**PERIODS PER WEEK**			**TOTAL CONTACT PERIODS**	**C**
					L	**T**	**P**		
THEORY									
1.	***	Programme Elective II	T	ES(PE)	3	0	0	3	3
2.	***	Programme Elective III	T	ES(PE)	3	0	0	3	3
3.	***	Programme Elective IV	T	ES(PE)	3	0	0	3	3
4.	***	Programme Elective V	T	ES(PE)	3	0	0	3	3
5.	PC25I02	Industry-Oriented Course II	-	SD	1	0	0	1	1
6.	PC25301	Project Work I	-	SD	0	0	12	12	6
TOTAL CREDITS								**25**	**19**

SEMESTER – IV									
S.NO	**COURSE CODE**	**COURSE TITLE**	**COURSE TYPE**	**CATE-GORY**	**PERIODS PER WEEK**			**TOTAL CONTACT PERIODS**	**C**
					L	**T**	**P**		
THEORY & LAB INTERGATED THEORY									
1.	PC25401	Project Work II	-	SD	0	0	24	24	12
TOTAL CREDITS								**24**	**12**

Total Credits: 69

PROFESSIONAL ELECTIVE

S. No.	Course Code	Course Title	Type	Periods per week			TCP	Credits
				L	T	P		
1.	PC25P01	Advanced Software Testing and Quality Assurance	T	3	0	0	3	3
2.	PC25P02	Agile Methodologies	T	3	0	0	3	3
3.	PC25P03	Web of Things	T	3	0	0	3	3
4.	PC25P04	Text and Speech Processing	T	3	0	0	3	3
5.	PC25P05	Advanced Deep Learning and Neural Networks	T	3	0	0	3	3
6.	PC25P06	Quantum Cryptography	T	3	0	0	3	3
7.	PC25P07	Quantum Machine Learning	T	3	0	0	3	3
8.	CP25C14	AI in IoT	T	3	0	0	3	3
9.	PC25P08	Web 3.0	T	3	0	0	3	3
10.	PC25P09	Advanced Large Language Models	T	3	0	0	3	3
11.	PC25P10	Edge and Fog Computing	T	3	0	0	3	3
12.	PC25P11	Green Computing and Sustainability	T	3	0	0	3	3
13.	PC25P12	Cognitive Computing	T	3	0	0	3	3
14.	PC25P13	Agentic AI	T	3	0	0	3	3
15.	PC25P14	Mixed Reality	T	3	0	0	3	3
16.	PC25P15	Blockchains Architecture and Design	T	3	0	0	3	3
17.	PC25P16	Human-Centered AI	T	3	0	0	3	3
18.	PC25P17	Vibe Coding	T	3	0	0	3	3
19.	PC25P18	Federated Learning	T	3	0	0	3	3
20.	PC25P19	Deep Learning for Computer Vision	T	3	0	0	3	3

SEMESTER I

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

Course Objectives:

- Understand advanced linear algebra concepts for computing applications.
- Apply multivariate statistical techniques to data-driven problems.
- Analyze number theory concepts for computational and cryptographic applications.
- Apply induction, CRT, and primality testing to solve computational problems.
- Evaluate cryptographic foundations and advanced number theory techniques.

Course Outcomes The students will be able to		**Knowledge Level**
CO1	**Explain** advanced linear algebra concepts and applications.	**K2**
CO2	**Apply** multivariate statistical techniques to data-driven problems.	**K3**
CO3	**Analyze** number theory concepts for computational applications.	**K4**
CO4	**Apply** induction, CRT, and quadratic congruences to solve problems.	**K4**
CO5	**Evaluate** cryptographic foundations and advanced number theory concepts.	**K5**

Unit I - Linear Algebra	**9**

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.

Unit II - Multivariate Analysis	**9**

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.

Unit III - Elementary Number Theory	**9**

The division algorithm, Divisibility and the Euclidean algorithm, The fundamental theorem of arithmetic, Modular arithmetic and basic properties of congruences.

Unit IV - Mathematical Induction	**9**
Principles of mathematical induction and well ordering principle. Primality Testing algorithms, Chinese Remainder Theorem, Quadratic Congruence.	
Unit V - Advanced Number Theory	**9**
Advanced Number Theory, Primality Testing algorithms, Chinese Remainder Theorem, Quadratic Congruence, Discrete Logarithm, Factorization Methods, Side Channel Attacks, Shannon Theory, Perfect Secrecy, Semantic Security.	
TOTAL PERIODS:60	

References:

1. Gilbert Strang, "Linear Algebra and Its Applications", 4th Edition, Cengage Learning, 2011.
2. Richard A. Johnson & Dean W. Wichern, "Applied Multivariate Statistical Analysis", 6th Edition, Pearson, 2018.
3. Koblitz, N., "A Course in Number Theory and Cryptography", 2nd Edition, Springer-Verlag, New York, 1994.
4. Victor Shoup, "A Computational Introduction to Number Theory" and Algebra, 1st Edition, Cambridge University Press, 2005

Mapping of COs with POs / PSOs														
	Pos											**PSOs**		
COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
1	3	2	–	2	2	–	–	–	–	–	2	3	2	1
2	3	3	2	3	2	–	–	–	–	–	2	3	3	1
3	3	3	–	2	2	–	–	–	–	–	2	3	2	1
4	3	3	2	2	2	–	–	–	–	–	2	3	3	1
5	3	3	2	2	3	–	2	–	–	–	3	3	3	2
Avg.	**3.0**	**2.8**	**1.2**	**2.2**	**2.2**	**0.0**	**0.4**	**0.0**	**0.0**	**0.0**	**2.2**	**3.0**	**2.6**	**1.2**

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

Course Objectives:

- Understand advanced linear data structures and memory optimization techniques.
- Design and implement advanced tree data structures for efficient searching and indexing.
- Apply graph algorithms to solve optimization and network problems.
- Develop solutions using advanced algorithm design paradigms.
- Analyze computational complexity and emerging algorithmic approaches.

Course Outcomes The students will be able to		**Knowledge Level**
CO1	**Apply** advanced linear data structures, memory optimization techniques, and hashing methods for efficient data management.	**K3**
CO2	**Design** and implement advanced tree structures and indexing mechanisms for efficient searching and storage applications.	**K4**
CO3	**Analyze** and apply graph algorithms to solve shortest path, spanning tree, and network flow problems.	**K4**
CO4	**Develop** solutions using divide-and-conquer, greedy, dynamic programming, backtracking, and randomized algorithms.	**K5**
CO5	**Evaluate** computational complexity, approximation techniques, heuristic methods, and emerging algorithmic approaches.	**K5**

UNIT I- 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 **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.

UNIT - II 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.

UNIT III - 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.

UNIT IV 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.

UNIT V 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.

TOTAL PERIODS:45

LIST OF EXPERIMENT:

1. Implement skip lists and measure performance compared with balanced BST.
2. Experiment with cache-aware data structures and analyze memory utilization.
3. Implement B+ tree for database indexing use-case.
4. Design a suffix tree-based algorithm for DNA sequence matching.
5. Implement Johnson's algorithm for sparse graph shortest paths.
6. Demonstrate maximum flow in traffic or network routing simulation.
7. Implement Strassen's algorithm and compare with naive matrix multiplication.
8. Develop a randomized algorithm for primality testing (Miller–Rabin).
9. Implement approximation algorithm for vertex cover.
10. Perform complexity analysis of a chosen NP-hard problem and implement a heuristic.
11. Implement randomized and streaming algorithms on real-world datasets.
12. Design parallel and distributed algorithms.

TOTAL PERIODS:30

References:

1. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, "Introduction to Algorithms", 3rd Edition, MIT Press, 2009.
2. Marcello La Rocca, "Advanced Algorithms and Data Structures", 1st Edition, Manning Publications, 2021.
3. Michael T. Goodrich, Roberto Tamassia, and David M. Mount, "Data Structures and Algorithms in C++", 2nd Edition, John Wiley & Sons, 2011.
4. Mark Allen Weiss, "Data Structures and Algorithm Analysis in C++", 4th Edition, Pearson Education, 2014.
   Adam Drozdek, "Data Structures and Algorithms in C++", 4th Edition, Cengage Learning, 2013.

E-resources:

1. https://www.theiotacademy.co/blog/data-structures-and-algorithms-in-c/
2. https://github.com/afrid18/Data_structures_and_algorithms_incpp
3. https://www.udemy.com/course/introduction-to-algorithms-and-data-structures-in-srsltid=AfmBOorEZlkgV7QzaEh6IqzAaKLjC-IpFU1NGgWFoHMLhOos-uDVKjCK

Mapping of COs with POs / PSOs														
	Pos											PSOs		
COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
1	3	3	2	2	3	–	–	–	–	–	2	3	2	1
2	3	3	3	2	3	–	–	–	–	–	2	3	2	1
3	3	3	2	3	3	–	–	–	–	–	2	3	3	1
4	3	3	3	2	3	–	–	–	–	–	3	3	3	2
5	3	3	2	2	3	–	2	–	–	–	3	3	3	2
Avg.	**3.0**	**3.0**	**2.4**	**2.2**	**3.0**	**0.0**	**0.4**	**0.0**	**0.0**	**0.0**	**2.4**	**3.0**	**2.6**	**1.4**

CP25C02	Advanced Database Technologies	L	T	P	C
		3	0	4	5

Course Objectives:

- Understand EER models, relational schemas, indexing, and file structures.
- Apply object-oriented and object-relational database concepts and OQL.
- Learn advanced SQL features and database design techniques.
- Analyze query processing, optimization, and execution strategies.
- Explore distributed databases, NoSQL systems, and big data applications.

Course Outcomes The students will be able to		**Knowledge Level**
CO1	**Explain** EER modeling, relational schema design, indexing, file structures, and hashing mechanisms.	**K2**
CO2	**Apply** object-oriented and object-relational database concepts, OQL, and advanced SQL features.	**K3**
CO3	**Analyze** query processing, optimization strategies, execution plans, and cost-based optimization techniques.	**K4**
CO4	**Design** and evaluate distributed database systems using fragmentation, replication, allocation, and architectures.	**K5**
CO5	**Assess** NoSQL systems, big data platforms, graph databases, temporal, spatial, and information retrieval systems.	**K5**

UNIT I - Entity Relationship Model	**9**

Entity Relationship Model Revised-Subclasses, Superclasses and Inheritance - Specialization and Generalization-Union Types-Aggregation. **Enhanced Entity Relational Model:** Relational Model Revised, Converting ER and EER Model to Relational Model-SQL and Advanced Features, File Structures, Hashing, and Indexing.

UNIT II - Object Relational Databases	**9**
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. **Query Processing and Optimization:** Query Processing, Query Trees and Heuristics, Query Execution Plans, Cost Based Optimization.	
UNIT III- Distributed Databases:	**9**
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.	
UNIT IV - NOSQL Systems and Bigdata	9
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.	
UNIT V - Advanced Database Models, Systems and Applications:	9
Active Database Concepts and Triggers, Temporal Database Concepts, Spatial Database Concepts, Multimedia Database Concepts, Deductive Database Concepts, Introduction to Information Retrieval and Web Search.	
TOTAL PERIODS:45	

References:

1. Elmasri, R., & Navathe, S. B., "Fundamentals of Database Systems", 7th Edition, Pearson Education, 2016.
2. Silberschatz, A., Korth, H. F., & Sudarshan, S., "Database System Concepts", 7th Edition, McGraw Hill Education, 2020.
3. Ceri, S., & Pelagatti, G., "Distributed Databases: Principles and Systems", 1st Edition, McGraw Hill, 1984.
4. Ramakrishnan, R., & Gehrke, J., "Database Management Systems", 3rd Edition, McGraw Hill, 2004.

E-resources:

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

Mapping of COs with POs / PSOs														
	Pos											PSOs		
COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
1	3	2	2	2	3	–	–	–	–	–	2	3	2	1
2	3	3	3	2	3	–	–	–	–	–	2	3	2	1
3	3	3	2	3	3	–	–	–	–	–	2	3	2	1
4	3	3	3	2	3	2	–	–	–	–	2	3	3	1
5	3	3	2	2	3	2	–	–	–	–	3	3	3	2
Avg.	**3.0**	**2.8**	**2.4**	**2.2**	**3.0**	**0.8**	**0.0**	**0.0**	**0.0**	**0.0**	**2.2**	**3.0**	**2.4**	**1.2**

CP25C03	Advanced Operating Systems	L	T	P	C
		3	0	0	3

Course Objectives:

- Analyze architectures and design issues of advanced operating systems.
- Develop models for process synchronization and recovery in complex environments.
- Understand memory and resource management techniques in modern operating systems.
- Evaluate distributed systems, virtualization, and containerization technologies.
- Study security, real-time systems, and emerging operating system paradigms.

Course Outcomes The students will be able to		**Knowledge Level**
CO1	**Explain** advanced OS architectures, process management, multithreading, synchronization, and scheduling strategies.	**K2**
CO2	**Apply** memory management and resource allocation techniques in modern operating systems.	**K3**
CO3	**Analyze** virtualization, containerization, distributed operating systems, and distributed file systems.	**K4**
CO4	**Evaluate** OS security models, trust mechanisms, real-time scheduling algorithms, and embedded OS architectures.	**K5**
CO5	**Assess** emerging OS paradigms including edge computing, cloud-native systems, serverless computing, and next-generation platforms.	**K5**

UNIT I - Advanced Process and Thread Management	**9**

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

UNIT II - Memory and Resource Management in Modern OS	**9**
Virtual memory, demand paging, page replacement policies-Huge pages, NUMA-aware memory management-Resource allocation in cloud-native environments.	
UNIT III - Virtualization and Containerization	**9**
Hypervisors (Type I & II), KVM, QEMU, Xen-Containers: Docker, LXC, systemd-nspawn-OS-level virtualization and namespaces **Distributed Operating Systems and File Systems:** Distributed scheduling, communication, and synchronization-Distributed file systems: NFS, GFS, HDFS-Transparency issues and fault tolerance.	
UNIT IV - Security and Trust in Operating Systems	**9**
Access control models: DAC, MAC, RBAC-OS hardening techniques, sandboxing, SELinux, AppArmor-Secure boot, rootkit detection, trusted execution environments **Real-Time and Embedded Operating Systems:** Real-time scheduling algorithms (EDF, RM)-POSIX RT extensions, RTOS architecture-TinyOS, FreeRTOS case studies.	
UNIT V - Edge and Cloud OS: Future Paradigms	**9**
Serverless OS, unikernels, lightweight OS for edge computing-Mobile OS internals (Android, iOS)-OS for quantum and neuromorphic computing (intro).	
TOTAL PERIODS: 45	

References:

1. Andrew S. Tanenbaum & Herbert Bos, "Modern Operating Systems", 5th Edition, Pearson, 2023.
2. Rajkumar Buyya et al., "Content Delivery Networks and Emerging Operating Systems", 1st Edition, Springer, 2022.
3. Abraham Silberschatz, Peter B. Galvin & Greg Gagne, "Operating System Concepts", 10th Edition, Wiley, 2022.
4. Thomas Anderson & Michael Dahlin, "Operating Systems: Principles and Practice", 2nd Edition, Recursive Books, 2021.
5. Remzi H. Arpaci-Dusseau & Andrea C. Arpaci-Dusseau, "Operating Systems: Three Easy Pieces", 1st Edition, Arpaci-Dusseau Books, 2020.

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

Mapping of COs with POs / PSOs														
	Pos											PSOs		
COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
1	3	2	2	2	3	–	–	–	–	–	2	3	2	1
2	3	3	2	2	3	–	–	–	–	–	2	3	2	1
3	3	3	3	3	3	–	–	–	–	–	2	3	3	1
4	3	3	2	2	3	2	2	–	–	–	2	3	2	2
5	3	3	2	2	3	2	–	–	–	–	3	3	3	2
Avg.	**3.0**	**2.8**	**2.2**	**2.2**	**3.0**	**0.8**	**0.4**	**0.0**	**0.0**	**0.0**	**2.2**	**3.0**	**2.4**	**1.4**

CP25C04	Advanced Compiler Design	L	T	P	C
		3	0	0	3

Course Objective:

- Analyze modern compiler design principles and optimization techniques.
- Design efficient front-end and back-end compiler components.
- Understand intermediate representations and program analysis techniques.
- Evaluate advanced optimization and runtime compilation methods.
- Explore secure compilation, verification, and emerging compiler technologies.

Course Outcomes The students will be able to		**Knowledge Level**
CO1	**Explain** intermediate representations, control flow analysis, SSA form, and dominance relations in compiler design.	**K2**
CO2	**Apply** program analysis, data flow analysis, dependence analysis, and code transformation techniques.	**K3**
CO3	**Analyze** advanced optimization techniques, polyhedral compilation, and runtime compilation strategies.	**K4**
CO4	**Design** compiler solutions for DSLs, machine learning applications, and specialized hardware accelerators.	**K5**
CO5	**Evaluate** secure compilation, compiler verification methods, and emerging compiler technologies.	**K5**

UNIT I - Intermediate Representations and Control Flow Analysis	**9**

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.

UNIT II - Program Analysis and Transformations	**9**
Data flow analysis- live variable analysis-reaching definitions-Alias analysis and dependence analysis-Loop optimizations and transformations **Advanced Optimizations and Polyhedral Compilation:** Polyhedral model for loop nests-Tiling, skewing, fusion, and vectorization-Profile-guided and feedback-directed optimizations.	
UNIT III - Just-in-Time (JIT) and Runtime Compilation	**9**
JIT compilation models: tracing, method-based-GraalVM architecture, Java HotSpot internals-LLVM JIT and dynamic language support. **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.	
UNIT IV - Domain-Specific Languages (DSLs) and Compiler Extensions	**9**
Designing DSLs for AI/ML, DSP, graphics-Code generation for custom accelerators-Integration with TensorFlow XLA and Halide.	
UNIT V - Security, Verification, and Future Trends	**9**
Secure compilation and type-safe intermediate representations-Compiler fuzzing and formal verification (e.g., CompCert)-Quantum compilers, multi-target compilers, and neuromorphic systems.	
TOTAL PERIODS:45	

References:

1. Keith D. Cooper & Linda Torczon, "Engineering a Compiler", 3rd Edition, Morgan Kaufmann, 2023.
2. Dick Grune, Henri E. Bal, Ceriel J. H. Jacobs & Koen G. Langendoen, "Modern Compiler Design", 2nd Edition, Springer, 2012.
3. Alfred V. Aho, Monica S. Lam, Ravi Sethi & Jeffrey D. Ullman, "Compilers: Principles, Techniques, and Tools", 2nd Edition, Pearson, 2006.
4. Markus Völter, "DSL Engineering: Designing, Implementing and Using Domain-Specific Languages", 1st Edition, dslbook.org, 2013.
5. Sagar Sarda & Mayur Pandey, "LLVM Essentials", 1st Edition, Packt Publishing, 2015.

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"

Mapping of COs with POs / PSOs														
	Pos											PSOs		
COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
1	3	2	2	2	3	–	–	–	–	–	2	3	2	1
2	3	3	2	3	3	–	–	–	–	–	2	3	2	1
3	3	3	3	3	3	–	–	–	–	–	2	3	3	1
4	3	3	3	2	3	–	–	–	–	–	3	3	3	2
5	3	3	2	2	3	–	2	–	–	–	3	3	3	2
Avg.	**3.0**	**2.8**	**2.4**	**2.4**	**3.0**	**0.0**	**0.4**	**0.0**	**0.0**	**0.0**	**2.4**	**3.0**	**2.6**	**1.4**

SEMESTER II

CP25201	Multicore Architecture	L	T	P	C
		3	0	2	4

Course Objectives:

- Understand the evolution and architecture of multicore and manycore systems.
- Learn thread management and synchronization mechanisms in parallel programming.
- Develop parallel applications using OpenMP for shared memory systems.
- Implement message-passing programs using MPI for distributed memory systems.
- Evaluate the performance and scalability of parallel applications.

Course Outcomes The students will be able to		**Knowledge Level**
CO1	**Explain** the architecture, memory organization, and design principles of multicore and manycore systems.	**K2**
CO2	**Apply** thread management, synchronization techniques, and inter-thread communication mechanisms.	**K3**
CO3	**Develop** and optimize parallel applications using OpenMP for shared memory architectures.	**K4**
CO4	**Implement** message-passing solutions using MPI and evaluate communication strategies in distributed memory systems.	**K4**
CO5	**Analyze** performance, scalability, and advanced parallel applications using profiling tools and modern multicore paradigms.	**K5**

UNIT I - 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.

UNIT II – THREAD MANAGEMENT AND SYNCHRONIZATION TECHNIQUES

Synchronization primitives: Mutex – Locks – Semaphores – Data races and deadlocks – Inter-thread communication: Condition variables – Pipes.

UNIT III - PARALLEL PROGRAMMING USING OPENMP FOR SHARED MEMORY

OpenMP architecture and memory model – Directives and work-sharing constructs – Parallelizing Loops and Task Management – Performance optimization.

UNIT IV - MESSAGE PASSING PROGRAMMING WITH MPI

MPI programming concepts – Point-to-point and collective communication – Derived data types and synchronization – Performance tuning.

UNIT V - EVALUATING PARALLEL PERFORMANCE, SCALABILITY AND ADVANCED APPLICATIONS

Measuring and analyzing performance – Scalability metrics – Use of Profiling Tools for Performance Bottlenecks – Tree search, N-body simulation, graph algorithms – Comparative Analysis of OpenMP and MPI Implementations – Trends in multicore and manycore processors.

TOTAL PERIODS:45

PRACTICAL:

1. Write a program to create and execute multiple threads using Pthreads.
2. Write a program to demonstrate mutex locks using std::mutex.
3. Write a program to compute the Fibonacci series using OpenMP tasks.
4. Write a Rust program to spawn threads and use channels for communication.
5. Write a program using MPI to calculate the average of an array using collective communication.
6. Write a program using the threading module to demonstrate data race and synchronization.
7. Write a program to perform matrix multiplication using OpenMP.
8. Write a Rust program to parallelize a for loop using the rayon crate.
9. Write a program and profile its performance using `gprof` or `perf`.
10. Mini Project: Write a C program to implement parallel merge sort or breadth-first search using both OpenMP and MPI.Compare complexity.

TOTAL PERIODS:30

References:

1. Michael J. Quinn, "Parallel Programming in C with MPI and OpenMP", 1st Edition, McGraw-Hill, 2004.
2. Yan Solihin, "Fundamentals of Parallel Multicore Architecture", 1st Edition, CRC Press, 2015.
3. Victor Alessandrini, "Shared Memory Application Programming", 1st Edition, Morgan Kaufmann, 2015.
4. Rohit Chandra, Ramesh Menon, Leo Dagum, David Kohr, Dror Maydan & Jeff McDonald, "Parallel Programming in OpenMP", 1st Edition, Morgan Kaufmann, 2000.
5. Gerassimos Barlas, "Multicore and GPU Programming: An Integrated Approach", 1st Edition, 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

Mapping of COs with POs / PSOs														
	Pos											**PSOs**		
COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
1	3	2	2	2	3	–	–	–	–	–	2	3	2	1
2	3	3	2	2	3	–	–	–	–	–	2	3	2	1
3	3	3	3	3	3	–	–	–	–	–	2	3	3	1
4	3	3	3	3	3	–	–	–	–	–	2	3	3	1
5	3	3	2	3	3	–	–	–	–	–	3	3	3	2
Avg.	**3.0**	**2.8**	**2.4**	**2.6**	**3.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**2.2**	**3.0**	**2.6**	**1.2**

CP25C05	Artificial Intelligence and MachineLearning	L	T	P	C
		3	0	0	3

COURSE OBJECTIVES:

- Understand problem-solving techniques, knowledge representation, and logical reasoning in AI.
- Learn probabilistic models and reasoning techniques for uncertain environments.
- Understand fundamental concepts and evaluation methods of machine learning.
- Apply supervised learning algorithms for classification and regression.
- Analyze unsupervised and probabilistic learning techniques for pattern discovery and prediction.

Course Outcomes The students will be able to		**Knowledge Level**
CO1	**Explain** problem-solving techniques, knowledge representation, logical reasoning, and intelligent agent models in AI.	**K2**
CO2	**Apply** probabilistic reasoning, Bayesian models, HMMs, and decision-making frameworks to handle uncertainty.	**K3**
CO3	**Analyze** machine learning paradigms, hypothesis spaces, evaluation methods, and bias-variance trade-offs.	**K4**
CO4	**Develop** and evaluate supervised learning models for classification and regression.	**K4**
CO5	**Apply** unsupervised and probabilistic learning techniques for clustering, dimensionality reduction, and predictive analytics.	**K5**

UNIT I - PROBLEM SOLVING AND KNOWLEDGE REPRESENTATION	**9**
Solving problems by searching – adversarial search – constraint satisfaction problems – Logical agents – Propositional logic – First order logic – Forward chaining – Backward chaining – Ontological Representations and Reasoning Systems.	
UNIT II - UNCERTAIN KNOWLEDGE AND REASONING	**9**
Overview of uncertainty and basic probability – Baye's rule – Bayesian networks – Hidden Markov models – Kalman filters – Utility Functions – Decision Networks – Sequential decision problems – Game theory	
UNIT III - MACHINE LEARNING INTRODUCTION	**9**
Types of learning – Hypothesis space – Inductive bias – Evaluation – Cross validation – Bias variance – Bias Variance Trade off.	
UNIT IV - SUPERVISED LEARNING	**9**
Linear Regression – Logistic Regression – Decision trees: Classification and regression trees – Neural networks, multilayer perceptron – Support vector machines: linear and non-linear kernel functions – K nearest neighbours – Ensemble learning: bagging and boosting.	
UNIT V - UNSUPERVISED AND PROBABILISTIC LEARNING	**9**
Clustering: partition, hierarchical and density based clustering – Self organizing maps – Expectation maximization – Gaussian mixture models – Principal component analysis – Bayesian learning: Bayesoptimal classifier, Naïve Bayes classifier, Bayesian belief networks.	
TOTAL PERIODS: 45	

References:

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

E-resources:

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

Mapping of COs with POs / PSOs														
	Pos											PSOs		
COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
1	3	2	2	2	2	–	–	–	–	–	2	3	2	1
2	3	3	2	3	3	–	–	–	–	–	2	3	3	1
3	3	3	2	3	3	–	–	–	–	–	2	3	3	1
4	3	3	3	3	3	–	–	–	–	–	2	3	3	1
5	3	3	3	3	3	2	2	–	–	–	3	3	3	2
Avg.	**3.0**	**2.8**	**2.4**	**2.8**	**2.8**	**0.4**	**0.4**	**0.0**	**0.0**	**0.0**	**2.2**	**3.0**	**2.8**	**1.2**

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

COURSE OBJECTIVES:

- Understand cloud computing concepts, service models, deployment models, and virtualization technologies.
- Learn the Big Data ecosystem and Hadoop framework for large-scale data processing.
- Explore NoSQL databases and Hadoop tools for data storage and processing.
- Understand advanced big data analytics and machine learning using Apache Spark.
- Analyze cloud integration, security, and cloud-based analytics applications.

Course Outcomes The students will be able to		**Knowledge Level**
CO1	**Explain** cloud computing concepts, service models, deployment models, virtualization technologies, and cloud platforms.	**K2**
CO2	**Apply** Hadoop ecosystem components and distributed processing techniques for large-scale data storage and analysis.	**K3**
CO3	**Utilize** NoSQL databases, data ingestion tools, and Hadoop-based frameworks for managing and processing big data.	**K3**
CO4	**Analyze** and implement advanced big data analytics and machine learning solutions using Apache Spark and real-time processing frameworks.	**K4**
CO5	**Evaluate** cloud integration strategies, security mechanisms, and cloud-based analytics applications for real-world domains.	**K5**

UNIT I - CLOUD COMPUTING	**9**

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

UNIT II - BIG DATA ECOSYSTEM AND HADOOP	9
Big Data characteristics (Volume, Velocity, Variety, Veracity, Value) - Challenges in Big Data - Hadoop Architecture and HDFS - MapReduce Programming Model – YARN.	
UNIT III - DATA STORAGE AND PROCESSING TECHNIQUES	**9**
NoSQL Databases: MongoDB, Cassandra, HBase - Data Ingestion: Sqoop, Flume - Hive, Pig – Architecture and Querying - Data Cleaning and Preparation	
UNIT IV - ADVANCED BIG DATA ANALYTICS	**9**
Introduction to Apache Spark - Spark vs. MapReduce - Spark SQL, Data Frames, RDDs - Machine Learning using Spark MLlib - Real-time processing: Kafka and Spark Streaming.	
UNIT V - CLOUD INTEGRATION AND SECURITY	**9**
Data on Cloud – Storage and Compute Services - Data Security and Privacy in Cloud - Cloud-based Data Analytics Use Cases - Cloud-Native vs Traditional Applications - Case Study: Smart Cities / Healthcare Analytics.	
TOTAL PERIODS:45	

References:

1. Rajkumar Buyya, James Broberg & Andrzej Goscinski, "Cloud Computing: Principles and Paradigms", 1st Edition, Wiley, 2011.
2. Tom White, "Hadoop: The Definitive Guide", 4th Edition, O'Reilly Media, 2015.
3. Vignesh Prajapati, "Big Data Analytics with R and Hadoop", 1st Edition, Packt Publishing, 2013.
4. Martin Kleppmann, "Designing Data-Intensive Applications", 1st Edition, O'Reilly Media, 2017.
5. Alan Gates, "Programming Pig", 1st Edition, O'Reilly Media, 2011.
6. Jure Leskovec, Anand Rajaraman & Jeffrey D. Ullman, "Mining of Massive Datasets", 3rd Edition, Cambridge University Press, 2020.

E-resources:

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

Mapping of COs with POs / PSOs														
	Pos											PSOs		
COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
1	3	2	2	2	3	–	–	–	–	–	2	3	2	1
2	3	3	2	3	3	–	–	–	–	–	2	3	3	1
3	3	3	2	3	3	–	–	–	–	–	2	3	3	1
4	3	3	3	3	3	–	–	–	–	–	2	3	3	1
5	3	3	3	2	3	2	2	–	–	–	3	3	3	2
Avg.	**3.0**	**2.8**	**2.4**	**2.6**	**3.0**	**0.4**	**0.4**	**0.0**	**0.0**	**0.0**	**2.2**	**3.0**	**2.8**	**1.2**

CP25C07	Quantum Computing	L	T	P	C
		3	0	0	3

COURSE OBJECTIVES:

- Understand the physical principles and mathematical foundations of quantum computing.
- Learn quantum mechanics postulates and quantum gate operations.
- Develop quantum circuits using quantum states and computational basis representations.
- Analyze fundamental quantum algorithms and their computational advantages.
- Implement and evaluate quantum programs on real quantum computing platforms.

Course Outcomes The students will be able to		**Knowledge Level**
CO1	**Explain** the physical principles and mathematical foundations used in quantum computing.	**K2**
CO2	**Describe** quantum computing postulates, qubit representations, Bloch sphere, and quantum gate operations.	**K2**
CO3	**Design** and analyze quantum circuits using computational basis, Dirac notation, and quantum gates.	**K4**
CO4	**Apply** fundamental quantum algorithms such as Deutsch–Jozsa and Grover's algorithms to solve computational problems.	**K3**
CO5	**Implement** and evaluate quantum programs on IBM Quantum platforms using Qiskit.	**K5**

UNIT I – PHYSICAL PROPERTIES AND MATHEMATICAL FOUNDATIONS	**9**

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

UNIT II - QUANTUM COMPUTING POSTULATES AND GATES	9
Review of postulates, Bloch sphere, Single qubit states and gates, superposition; Two Qubit States and Gates - Bell States, Entanglement, CNOT gate, Phase oracles, Pauli Gates.	
UNIT III - QUANTUM COMPUTING CIRCUITS	**9**
Dirac's notation for quantum computing, Computational Basis, Orthonormality, Hadamard and Phase Gates - building quantum circuits	
UNIT IV - FUNDAMENTAL QUANTUM ALGORITHMS	**9**
Deutsch–Josza Algorithm, Grover search algorithm: Problem definition, Amplitude amplification, Grover oracle, diffuser, multiple solutions in the search space.	
UNIT V - PROGRAMMING ON A REAL QUANTUM COMPUTER	**9**
Coding a real time quantum computer via IBMQ to carry out basic quantum measurement and state analysis.	
TOTAL PERIODS: 45	

Text Books

1. Chuck Easttom, "Quantum Computing Fundamentals", 1st Edition, Addison-Wesley Professional, 2021.
2. Qiskit Team, "Qiskit Textbook", IBM Quantum, 2022. (Available online)

References

1. Venkateswaran Kasirajan, "Fundamentals of Quantum Computing", 1st Edition, Springer International Publishing, 2021.
2. Chris Bernhardt, "Quantum Computing for Everyone", 1st Edition, The MIT Press, 2020.
3. Michael A. Nielsen & Isaac L. Chuang, "Quantum Computation and Quantum Information", 10th Anniversary Edition, Cambridge University Press, 2013.

Mapping of COs with POs / PSOs														
	Pos											PSOs		
COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
1	3	2	1	2	2	–	–	–	–	–	2	3	2	1
2	3	2	2	2	2	–	–	–	–	–	2	3	2	1
3	3	3	3	3	3	–	–	–	–	–	2	3	3	1
4	3	3	2	3	3	–	–	–	–	–	2	3	3	1
5	3	3	3	3	3	–	–	–	–	–	3	3	3	2
Avg.	**3.0**	**2.6**	**2.2**	**2.6**	**2.6**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**2.2**	**3.0**	**2.6**	**1.2**

PC25P01	ADVANCED SOFTWARE TESTING AND QUALITY ASSURANCE	L	T	P	C
		3	0	0	3

Course Objectives:

- Understand software testing principles, STLC activities, and quality attributes.
- Apply testing techniques and risk-based testing strategies for defect detection.
- Design and evaluate test cases using advanced testing approaches.
- Utilize test automation frameworks and continuous testing practices.
- Analyze software quality assurance frameworks, metrics, and AI-driven testing trends.

Course Outcomes The students will be able to		**Knowledge Level**
CO1	**Explain** software testing fundamentals, STLC phases, quality attributes, and testing classifications.	**K2**
CO2	**Apply** testing types, risk-based testing approaches, and prioritization techniques to improve software quality.	**K3**
CO3	**Design** and evaluate test cases using black-box, white-box, model-based, and fault-based testing techniques.	**K4**
CO4	**Utilize** test automation frameworks, continuous testing practices, and industry-standard tools for software validation.	**K3**
CO5	**Analyze** software quality assurance frameworks, testing metrics, process improvement models, and AI-driven testing trends.	**K4**

UNIT I – FOUNDATIONS AND TEST LIFECYCLE	**9**

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**.**

UNIT II – TYPES OF TESTING AND RISK-BASED APPROACHES	**9**

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.

UNIT III – ADVANCED TEST DESIGN TECHNIQUES	**9**
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.	
UNIT IV – TEST AUTOMATION AND CONTINUOUS TESTING	**9**
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.	
UNIT V – SOFTWARE QUALITY ASSURANCE, PROCESS IMPROVEMENT AND EMERGING TRENDS	**9**
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. **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.	
TOTAL PERIODS: 45	

References:

1. Glenford J. Myers, Corey Sandler & Tom Badgett, "The Art of Software Testing", 3rd Edition, Wiley, 2011.
2. Paul C. Jorgensen, "Software Testing: A Craftsman's Approach", 4th Edition, CRC Press, 2013.
3. Ron Patton, "Software Testing", 2nd Edition, Sams Publishing, Pearson Education, 2007.
4. Aditya P. Mathur, "Foundations of Software Testing", 2nd Edition, Pearson Education, 2013.
5. Roger S. Pressman & Bruce R. Maxim, "Software Engineering: A Practitioner's Approach", 9th Edition, McGraw-Hill Education, 2019.

Mapping of COs with POs / PSOs														
	Pos											PSOs		
COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
1	3	2	2	2	2	–	1	–	–	–	2	3	2	1
2	3	3	2	3	3	–	1	–	–	–	2	3	2	1
3	3	3	3	3	3	–	1	–	–	–	2	3	2	1
4	3	3	2	2	3	–	1	–	–	–	2	3	2	1
5	3	3	2	3	3	2	2	–	–	–	3	3	3	2
Avg.	**3.0**	**2.8**	**2.2**	**2.6**	**2.8**	**0.4**	**1.2**	**0.0**	**0.0**	**0.0**	**2.2**	**3.0**	**2.2**	**1.2**

PC25P02	AGILE METHODOLOGIES	L	T	P	C
		3	0	0	3

Course Objectives:

- Understand Agile principles, values, and practices in software development.
- Develop an Agile mindset and apply Lean principles for collaboration and continuous improvement.
- Learn Agile frameworks such as Scrum, XP, Kanban, and SAFe.
- Apply Agile project management techniques, estimation methods, and metrics.
- Explore Agile engineering practices, DevOps integration, and real-world Agile adoption.

Course Outcomes The students will be able to		**Knowledge Level**
CO1	**Explain** Agile principles, values, Agile Manifesto, and the evolution of software development methodologies.	**K2**
CO2	**Apply** Agile mindset and Lean principles to improve team collaboration and continuous improvement.	**K3**
CO3	**Apply** Agile frameworks such as Scrum, XP, Kanban, and SAFe in software development projects.	**K3**
CO4	**Analyze** Agile project management techniques, estimation methods, and Agile metrics for project delivery.	**K4**
CO5	**Evaluate** Agile engineering practices, DevOps integration, and Agile adoption in diverse organizational contexts.	**K5**

UNIT I – EVOLUTION OF SOFTWARE DEVELOPMENT AND AGILE FOUNDATIONS	**9**

Evolution of Software Development Methodologies – Challenges in Traditional Models that led to Agile – The Agile Manifesto – Agile Values and Principles.

UNIT II – AGILE MINDSET AND CULTURAL TRANSFORMATION	9
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	
UNIT III – AGILE PROCESS MODELS AND FRAMEWORKS	**9**
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.	
UNIT IV – AGILE PROJECT MANAGEMENT AND PLANNING	**9**
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.	
UNIT V – AGILE ENGINEERING PRACTICES, SCALING AND INDUSTRY APPLICATIONS	**9**
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 – 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.	
TOTAL PERIODS: 45	

References:

1. Ken Schwaber & Jeff Sutherland, "Software in 30 Days: How Agile Managers Beat the Odds, Delight Their Customers, and Leave Competitors in the Dust", 1st Edition, Wiley, 2012.
2. Mike Cohn, "User Stories Applied: For Agile Software Development", 1st Edition, Addison-Wesley Professional, 2004.
3. Craig Larman & Bas Vodde, "Scaling Lean & Agile Development: Thinking and Organizational Tools for Large-Scale Scrum", 1st Edition, Addison-Wesley Professional, 2008.
4. Robert C. Martin, "Clean Code: A Handbook of Agile Software Craftsmanship", 1st Edition, Pearson Education (Prentice Hall), 2008.

E-Resources:

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

Mapping of COs with POs / PSOs														
	Pos											**PSOs**		
COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
1	2	2	1	–	2	–	1	2	2	1	2	2	1	2
2	2	2	2	–	2	–	2	3	3	2	2	2	1	3
3	2	3	3	2	3	–	1	3	2	2	2	3	2	3
4	2	3	3	2	2	–	1	3	3	3	2	3	2	3
5	2	3	3	2	3	1	2	3	3	2	3	3	2	3
Avg.	**2.0**	**2.6**	**2.4**	**1.2**	**2.4**	**0.2**	**1.4**	**2.8**	**2.6**	**2.0**	**2.2**	**2.6**	**1.6**	**2.8**

PC25P03	WEB OF THINGS	L	T	P	C
		3	0	0	3

Course Objectives:

- Understand Web of Things concepts, architectures, standards, and communication protocols.
- Model smart devices using W3C Thing Descriptions and semantic technologies.
- Design and implement WoT solutions using servients, scripting APIs, and protocol bindings.
- Understand interoperability, security, privacy, and trust mechanisms in WoT ecosystems.
- Develop scalable WoT applications by integrating embedded devices and smart platforms.

Course Outcomes The students will be able to		**Knowledge Level**
CO1	**Explain** Web of Things concepts, architectures, standards, and communication protocols.	**K2**
CO2	**Apply** W3C Thing Descriptions and semantic technologies to model and describe smart devices.	**K3**
CO3	**Design** and implement WoT solutions using servients, scripting APIs, and protocol bindings.	**K4**
CO4	**Analyze** interoperability, security, privacy, and trust mechanisms in WoT ecosystems.	**K4**
CO5	**Develop** scalable smart applications by integrating embedded devices with WoT platforms.	**K5**

UNIT I – INTRODUCTION TO WEB OF THINGS	**9**

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).

UNIT II – WOT STANDARDS AND THING DESCRIPTION (TD)	**9**
W3C WoT Architecture and Building Blocks – Thing Description (TD): Syntax, Vocabulary, JSON-LD Format – TD Schema Validation and Interpretation – Semantic Annotation in TD.	
UNIT III – IMPLEMENTATION OF WOT SERVIENTS	**9**
WoT Servient Structure: Exposed vs Consumed Things – Node-WoT Framework – WoT Scripting API – Protocol Bindings: HTTP, MQTT, CoAP.	
UNIT IV – INTEROPERABILITY, SECURITY, AND PRIVACY	**9**
Interoperability Challenges – Security Mechanisms: HTTPS, OAuth 2.0, Access Control – Privacy and Anonymization – Trust and Identity in WoT Ecosystems.	
UNIT V – APPLICATIONS, FUTURE TRENDS AND INTEGRATION TO WOT	**9**
Smart Cities, Smart Homes, e-Health, IIoT – Cloud and Edge Integration – Data Analytics and AI in WoT – Future Directions: Digital Twins, Metaverse – Integrating an Arduino to WoT – Integrating BeagleBone to WoT – Integrating an Intel Edition to WoT – Integrating Other Embedded Systems to WoT.	
TOTAL PERIODS: 45	

References:

1. Michael Kuniavsky, "Smart Things: Ubiquitous Computing User Experience Design", 1st Edition, Morgan Kaufmann, 2010.
2. Dominique Guinard & Vlad Trifa, "Building the Web of Things", 1st Edition, Manning Publications, 2016.
3. Hakima Chaouchi, "The Internet of Things: Connecting Objects to the Web", 1st Edition, Wiley, 2010.
4. Peter Waher, "Mastering Internet of Things: Design and Build Advanced IoT Solutions Using the State-of-the-Art Technology", 1st Edition, Packt Publishing, 2018.
5. Alasdair Gilchrist, "Industry 4.0: The Industrial Internet of Things", 1st Edition, Apress, 2016.

Web Resource:

1. W3C WoT Working Group Documents: https://www.w3.org/WoT/

Mapping of COs with POs / PSOs														
	Pos											PSOs		
COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
1	3	2	2	2	3	–	–	–	–	–	2	3	2	1
2	3	3	2	3	3	–	–	–	–	–	2	3	3	1
3	3	3	3	3	3	–	–	–	–	–	2	3	3	1
4	3	3	2	3	3	2	2	–	–	–	2	3	3	1
5	3	3	3	3	3	2	2	–	–	–	3	3	3	2
Avg.	**3.0**	**2.8**	**2.4**	**2.8**	**3.0**	**0.8**	**0.8**	**0.0**	**0.0**	**0.0**	**2.2**	**3.0**	**2.8**	**1.2**

PC25P04	TEXT AND SPEECH PROCESSING	L	T	P	C
		3	0	0	3

Course Objectives:

- Understand the fundamental concepts of natural language processing and speech processing systems.
- Learn linguistic analysis, text processing methods, and language modeling techniques.
- Understand speech signal processing, feature extraction, acoustic modeling, and speech recognition.
- Study speech synthesis technologies and applications of text and speech processing.
- Apply modern NLP and speech processing models, including Transformers and Large Language Models.

Course Outcomes The students will be able to		**Knowledge Level**
CO1	**Explain** the fundamentals of NLP, speech processing, text normalization, and speech signal concepts.	**K2**
CO2	**Apply** linguistic analysis, parsing techniques, named entity recognition, and language models for text processing.	**K3**
CO3	**Apply** speech signal processing techniques, feature extraction methods, and acoustic models for speech recognition.	**K3**
CO4	**Analyze** speech synthesis systems, evaluation metrics, and practical speech applications.	**K4**
CO5	**Apply** modern NLP and speech processing models, including RNNs, LSTMs, Transformers, and Large Language Models, to intelligent applications.	**K5**

UNIT I – INTRODUCTION TO TEXT AND SPEECH PROCESSING	**9**

NLP Pipeline Overview – Human Speech Production and Acoustic Phonetics – Text Normalization: Tokenization, Stemming, Lemmatization – Basics of Speech Signal: Sampling, Frequency, Noise.

UNIT II – TEXT PROCESSING TECHNIQUES	**9**
Part-of-Speech Tagging – Parsing Techniques: CFG, Dependency Parsing – Named Entity Recognition (NER), Chunking – Language Modeling: N-gram Models.	
UNIT III – SPEECH SIGNAL PROCESSING	**9**
Feature Extraction: MFCC, PLP, LPC – Acoustic Modeling: HMM, GMM – Basics of ASR (Automatic Speech Recognition) – Hidden Markov Models for Speech.	
UNIT IV – SPEECH SYNTHESIS AND APPLICATIONS	**9**
Text-to-Speech (TTS) Systems – Speech Databases and Corpora – Evaluation Metrics: WER, CER – Applications: Voice Assistants, Dictation Systems.	
UNIT V – NLP AND SPEECH IN PRACTICE AND MODELS FOR NLP	**9**
Sentiment Analysis – Chatbots and Dialog Systems – Multilingual and Low-Resource NLP – Ethics in Speech and Text Technology – RNNs and LSTM – The Transformer – Large Language Models – Masked Language Models.	
TOTAL PERIODS: 45	

References:

1. Daniel Jurafsky & James H. Martin, "Speech and Language Processing", 3rd Edition (Draft), Pearson, 2023.
2. Lawrence Rabiner & Biing-Hwang Juang, "Fundamentals of Speech Recognition", 1st Edition, Pearson, 2011.
3. Christopher D. Manning & Hinrich Schütze, "Foundations of Statistical Natural Language Processing", 1st Edition, MIT Press, 1999.
4. Thomas Schultz & Katrin Kirchhoff, "Multilingual Speech Processing", 1st Edition, Academic Press, 2006.
5. Jacob Eisenstein, "Introduction to Natural Language Processing", 1st Edition, MIT Press, 2019.

Mapping of COs with POs / PSOs														
	Pos											PSOs		
COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
1	3	2	2	2	2	–	–	–	–	–	2	3	2	1
2	3	3	2	3	3	–	–	–	–	–	2	3	3	1
3	3	3	2	3	3	–	–	–	–	–	2	3	3	1
4	3	3	3	3	3	–	–	–	–	–	2	3	3	1
5	3	3	3	3	3	2	2	–	–	–	3	3	3	2
Avg.	**3.0**	**2.8**	**2.4**	**2.8**	**2.8**	**0.4**	**0.4**	**0.0**	**0.0**	**0.0**	**2.2**	**3.0**	**2.8**	**1.2**

PC25P05	ADVANCED DEEP LEARNING AND NEURAL NETWORKS	L	T	P	C
		3	0	0	3

Course Objectives:

- Understand neural network architectures and training methodologies.
- Explore optimization techniques, regularization methods, and advanced deep learning architectures.
- Analyze optimization and generalization techniques in deep neural networks.
- Study advanced models such as encoder-decoders, attention mechanisms, VAEs, GANs, GCNs, and DBNs.
- Design intelligent deep learning solutions for real-world applications in vision, NLP, speech, and time-series analysis.

Course Outcomes The students will be able to		**Knowledge Level**
CO1	**Explain** neural network architectures, activation functions, optimization techniques, and training methodologies.	**K2**
CO2	**Apply** regularization methods and optimization algorithms to improve deep learning model performance and generalization.	**K3**
CO3	**Analyze** CNNs, RNNs, encoder-decoder architectures, and attention mechanisms for complex learning tasks.	**K4**
CO4	**Design** and implement advanced generative models such as VAEs, GANs, GCNs, and DBNs.	**K6**
CO5	**Evaluate** transformer-based models and deep learning solutions for real-world AI applications.	**K5**

UNIT I – NEURAL NETWORKS	**9**
Introduction – Perceptron – Multilayer Feed forward networks – Back propagation – Activation functions – Loss Function – Regularization: Data Augmentation – Noise Robustness – Early Stopping – Bagging – Dropout – batch normalization.	

UNIT II – CNN AND RNN	**9**
Convolutional Neural Networks – Convolution Operation – Architecture Overview – Input layers – Convolutional layers – pooling layers – fully connected layers Recurrent Neural networks – LSTM – Bidirectional RNNs – RNN Language model – Word Level RNN - Deep Recurrent Networks – Recursive Neural Networks.	

UNIT III – OPTIMIZATION AND DEEP NETWORKS	**9**
Optimization in deep learning – Non-convex optimization for deep networks- Stochastic Optimization- Generalization in neural networks - Spatial Transformer Networks Tuning Deep Networks – Basic Concepts – Matching input data and network architecture – relating model goal and output layers-Working layer count, parameter count and Memory – weight initialization strategies-using activation functions – applying loss function – Dealing with overfitting.	
UNIT IV – ENCODER / DECODER MODELS AND REGRESSIVE MODELS	**9**
Encoder Decoder Models - Attention Mechanism - Attention over images – Hierarchical Attention – Variational auto encoders – Autoregressive models – NADE – MADE - PixelRNN – Generative Adversarial Networks (GANs) – Graph Convolution Network – Deep Belief Network.	
UNIT V – ADVANCED APPLICATIONS OF NEURAL NETWORKS & DEEP LEARNING	**9**
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 - Deep learning for Computer Vision – Deep learning for Time Series – Deep learning for text – Generative Deep learning.	
TOTAL PERIODS: 45	

References:

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

Mapping of COs with POs / PSOs														
	Pos											**PSOs**		
COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
1	3	2	2	2	3	–	–	–	–	–	2	3	2	1
2	3	3	2	3	3	–	–	–	–	–	2	3	3	1
3	3	3	3	3	3	–	–	–	–	–	2	3	3	1
4	3	3	3	3	3	–	–	–	–	–	3	3	3	2
5	3	3	3	3	3	2	2	–	–	–	3	3	3	2
Avg.		**2.8**	**2.6**	**2.8**	**3.0**	**0.4**	**0.4**	**0.0**	**0.0**	**0.0**	**2.4**	**3.0**	**2.8**	**1.4**

PC25P06	QUANTUM CRYPTOGRAPHY	L	T	P	C
		3	0	0	3

Course Objectives:

- Understand the principles of quantum information and quantum computation.
- Learn quantum algorithms and their applications in cryptography.
- Analyze quantum cryptanalysis and its impact on classical cryptosystems.
- Explore post-quantum cryptographic techniques and secure communication methods.
- Design and evaluate advanced quantum cryptographic protocols.

Course Outcomes The students will be able to		**Knowledge Level**
CO1	**Explain** quantum information concepts, qubits, quantum gates, circuits, and computation models.	**K2**
CO2	**Apply** quantum algorithms such as Deutsch, Deutsch-Jozsa, Simon, Grover, and Shor algorithms to computational problems.	**K3**
CO3	**Analyze** quantum cryptanalysis techniques and their impact on RSA, Diffie-Hellman, and ElGamal cryptosystems.	**K4**
CO4	**Evaluate** post-quantum cryptographic schemes based on lattices, codes, isogenies, LWE, SIS, and homomorphic encryption.	**K5**
CO5	**Design** and assess advanced quantum cryptographic protocols including QKD, quantum encryption, quantum money, and device-independent cryptography.	**K6**

UNIT I – QUANTUM INFORMATION	**9**

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.

UNIT II – QUANTUM ALGORITHMS	**9**
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.	
UNIT III – QUANTUM CRYPTANALYSIS	**9**
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.	
UNIT IV – POST-QUANTUM CRYPTOGRAPHY	**9**
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).	
UNIT V - ADVANCED QUANTUM CRYPTOGRAPHY – I & II	**9**
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 - **Advanced Quantum Cryptography – II:** Quantum key distribution with imperfect devices – beyond point -to -point quantum key distribution – device independent Quantum cryptography.	
TOTAL PERIODS: 45	

References:

1. Michael A. Nielsen & Isaac L. Chuang, **"Quantum Computation and Quantum Information"**, 10th Anniversary Edition, Cambridge University Press, 2012.
2. Riley Tipton Perry, **"Quantum Computing from the Ground Up"**, 1st Edition, World Scientific Publishing, 2012.
3. Richard J. Lipton & Kenneth W. Regan, **"Quantum Algorithms via Linear Algebra: A Primer"**, 1st Edition, MIT Press, 2014.
4. Jack D. Hidary, **"Quantum Computing: An Applied Approach"**, 1st Edition, Springer, 2019.

E-resources:

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

Mapping of COs with POs / PSOs														
	Pos											PSOs		
COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
1	3	2	2	2	3	–	–	–	–	–	2	3	2	1
2	3	3	2	3	3	–	–	–	–	–	2	3	3	1
3	3	3	3	3	3	1	2	–	–	–	2	3	3	1
4	3	3	3	3	3	2	2	–	–	–	3	3	3	1
5	3	3	3	3	3	2	3	–	–	–	3	3	3	2
Avg.	**3.0**	**2.8**	**2.6**	**2.8**	**3.0**	**1.0**	**1.4**	**0.0**	**0.0**	**0.0**	**2.4**	**3.0**	**2.8**	**1.2**

PC25P07	QUANTUM MACHINE LEARNING	L	T	P	C
		3	0	0	3

Course Objectives:

- Understand the fundamentals of quantum computing and quantum machine learning.
- Design and implement quantum algorithms for machine learning problems.
- Analyze quantum machine learning models such as QSVM, QPCA, QkNN, and QNN.
- Explore quantum optimization and hybrid quantum-classical approaches such as VQE and QAOA.
- Develop and evaluate quantum machine learning solutions using Qiskit and TensorFlow Quantum.

Course Outcomes The students will be able to		**Knowledge Level**
CO1	**Explain** the fundamentals of quantum computing, quantum information theory, and principles of quantum machine learning.	**K2**
CO2	**Apply** quantum algorithms such as QFT, QPE, Grover's, and Shor's algorithms to computational and machine learning problems.	**K3**
CO3	**Analyze** quantum machine learning models including QSVM, QPCA, QkNN, and Quantum Neural Networks for intelligent data processing.	**K4**
CO4	**Evaluate** quantum optimization techniques, hybrid quantum-classical models, and variational algorithms for machine learning applications.	**K5**
CO5	**Design** and assess advanced quantum machine learning solutions using Qiskit, TensorFlow Quantum, and emerging frameworks for real-world applications.	**K6**

UNIT I – INTRODUCTION TO QUANTUM COMPUTING AND QUANTUM MACHINE LEARNING	**9**

Basics of Quantum Computing: Qubits, Superposition, and Entanglement - Quantum Gates, Circuits, and Measurement - Quantum Algorithms: Grover’s and Shor’s Algorithm - Introduction to Quantum Machine Learning (QML) - Classical vs Quantum ML - Applications of Quantum ML.

UNIT II – QUANTUM INFORMATION THEORY AND QUANTUM ALGORITHMS	**9**
Quantum Entropy and Quantum Information - Quantum Teleportation and Superdense Coding - No-Cloning Theorem and Quantum Cryptography - Quantum Fourier Transform (QFT) - Quantum Phase Estimation (QPE).	
UNIT III – QUANTUM ALGORITHMS, QUANTUM OPTIMIZATION AND HYBRID QUANTUM-CLASSICAL MODELS FOR MACHINE LEARNING	**9**
Quantum Support Vector Machines (QSVM) - Quantum Principal Component Analysis (QPCA) - Quantum k-Nearest Neighbors (QkNN) - Quantum Neural Networks (QNN) - Variational Quantum Eigensolver (VQE) - Quantum Approximate Optimization Algorithm (QAOA) - Hybrid Quantum-Classical Algorithms - Applications of Quantum Optimization in Machine Learning.	
UNIT IV – APPLICATIONS, TOOLS, AND FUTURE OF QUANTUM MACHINE LEARNING	**9**
Applications of Quantum Machine Learning in Finance, Healthcare, NLP, and Drug Discovery - Quantum Programming Tools: Qiskit, TensorFlow Quantum - The Future of Quantum ML: Challenges and Opportunities.	
UNIT V - ADVANCED QUANTUM MACHINE LEARNING – I & II	**9**
Advanced Quantum Machine Learning – I: Quantum Mechanics and Data-Driven Physics - Kernelizing Quantum Mechanics - Qubit Maps - One-Qubit Transverse-Field Ising Model and Variational Quantum Algorithms. **Advanced Quantum Machine Learning – II:** Two-Qubit Transverse-Field Ising Model and Entanglement - Variational Algorithms, Quantum Approximate Optimization Algorithm, and Neural Network Quantum States with Two Qubits - Quantum Reservoir Computing.	
TOTAL PERIODS: 45	

References:

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

Mapping of COs with POs / PSOs

	Pos											**PSOs**		
COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
1	3	2	2	2	3	–	–	–	–	–	2	3	2	1
2	3	3	2	3	3	–	–	–	–	–	2	3	3	1
3	3	3	3	3	3	–	–	–	–	–	2	3	3	1
4	3	3	3	3	3	1	1	–	–	–	3	3	3	1
5	3	3	3	3	3	2	2	–	–	–	3	3	3	2
Avg.	**3.0**	**2.8**	**2.6**	**2.8**	**3.0**	**0.6**	**0.6**	**0.0**	**0.0**	**0.0**	**2.4**	**3.0**	**2.8**	**1.2**

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

Course Objectives:

- 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.

Course Outcomes The students will be able to		**Knowledge Level**
CO1	**Explain** the fundamentals of Artificial Intelligence, Internet of Things architectures, and their integration in intelligent systems.	**K2**
CO2	**Apply** data acquisition, preprocessing, feature engineering, and AI techniques for effective IoT data analytics.	**K3**
CO3	**Develop and evaluate** machine learning and deep learning models for classification, clustering, anomaly detection, and decision-making in IoT environments.	**K4**
CO4	**Implement** Edge AI solutions and AI-powered security mechanisms for real-time and resource-constrained IoT systems.	**K5**
CO5	**Analyze** emerging trends, ethical considerations, sustainability challenges, and large-scale deployment strategies of AI-enabled IoT applications.	**K5**

UNIT I – INTRODUCTION TO AI IN IOT	**9**

Fundamentals of AI and IoT - IoT Architecture and AI Integration - AI Models for IoT (Machine Learning, Deep Learning) - AI-Driven IoT Applications (Smart Cities, Healthcare, Industrial IoT) - Ethical Considerations in AI-Enabled IoT – Case Study: Smart Agriculture with AI-Enabled IoT Sensors.

UNIT II DATA ACQUISITION & PRE-PROCESSING IN AI-ENABLED IOT	**9**
Sensor Data Collection & Processing - AI-Based Data Normalization, Filtering & Feature Engineering - Challenges in Data Handling & Storage for AI Models - Ethical Considerations in AI-Driven Data Processing – Case Study: Predictive Maintenance in Smart Factories.	
UNIT III – AI MODELS FOR IOT DATA PROCESSING & DECISION-MAKING	**9**
Supervised & Unsupervised AI Models in IoT - Classification, Clustering & Anomaly Detection - Deep Learning Models for IoT: CNNs, RNNs - Performance Metrics & Model Optimization – Case Study: AI-Driven Traffic Monitoring System.	
UNIT IV – EDGE AI & SECURITY FOR IOT SYSTEMS	**9**
Edge AI Deployment Strategies - Edge AI for Real-Time Decision-Making - Deploying AI Models in Resource-Constrained IoT Devices - AI-Powered Security & Intrusion Detection in IoT Networks - Ethical Implications & Privacy Concerns – Case Study: AI- Based Intrusion Detection in Healthcare IoT Networks.	
UNIT V - AI IN IOT – INDUSTRY TRENDS & ETHICS AND FUTURE TRENDS	**9**
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 - Monetization Strategies & AI Business Models in IoT - Green AI and IoT Regulations and Standardization - Edge AI advancements for real-time IoT decision-making - Quantum AI & IoT interoperability for future applications - Case Study: AI-Driven Smart Cities for Sustainable Urban Development.	
TOTAL PERIODS: 45	

References:

1. Fadi Al-Turjman, "Artificial Intelligence in IoT", 1st Edition, Springer, 2019.
2. Ambika Nagaraj, "The Role of AI in Enhancing IoT-Cloud Applications", 1st Edition, Bentham Books, 2023.
3. Rajkumar Buyya & Amir Vahid Dastjerdi, "Internet of Things: Principles and Paradigms", 1st Edition, Elsevier, 2016.
4. Jason Ioannou, "Data Analytics for IoT and Smart Systems", 1st Edition, Springer, 2020.
5. Ian Goodfellow, Yoshua Bengio & Aaron Courville, "Deep Learning", 1st Edition, MIT Press, 2016.
6. Weisong Shi, "Edge Computing: Vision and Challenges", 1st Edition, Springer, 2020.
7. Mauro Conti, "Security and Privacy in the Age of IoT", 1st Edition, 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.

Mapping of COs with POs / PSOs

	Pos											**PSOs**		
COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
1	3	2	1	1	2	2	2	1	1	–	2	2	2	1
2	3	3	2	2	3	1	1	1	1	–	2	3	3	1
3	3	3	3	3	3	1	1	2	1	1	2	3	3	2
4	3	3	3	2	3	2	2	2	1	1	2	3	3	2
5	2	3	2	2	2	3	3	2	2	2	3	2	3	3
Avg.	**2.8**	**2.8**	**2.2**	**2.0**	**2.6**	**1.8**	**1.8**	**1.6**	**1.2**	**0.8**	**2.2**	**2.6**	**2.8**	**1.8**

PC25P08	Web 3.0	L	T	P	C
		3	0	0	3

Course Objectives:

- 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.

Course Outcomes The students will be able to		**Knowledge Level**
CO1	**Explain** the evolution, architecture, principles, and technologies underlying Web 3.0, including decentralization, semantic interoperability, and distributed systems.	**K2**
CO2	**Apply** blockchain concepts, consensus mechanisms, distributed ledger technologies, and decentralized networking models in Web 3.0 applications.	**K3**
CO3	**Design and develop** smart contracts and decentralized applications (dApps) using blockchain platforms and open web standards.	**K4**
CO4	**Implement** digital asset tokenization, NFTs, decentralized storage systems, and interoperable metadata frameworks in Web 3.0 ecosystems.	**K5**
CO5	**Analyze** security, digital identity, decentralized governance, scalability, interoperability, and regulatory challenges in Web 3.0 environments.	**K5**

UNIT I –INTRODUCTION TO WEB 3.0	**9**

Evolution of the Internet – Transition from Web 1.0 (static/read-only) to Web 2.0 (interactive/centralized) and on to Web 3.0 (decentralized) – Core Concepts – Principles – Decentralisation – Semantic Interoperability - Exploration of Web 3.0's layered structure: Blockchain, Decentralized storage, P2P networks, and Semantic Web protocols.

UNIT II – BLOCKCHAIN TECHNOLOGY AND DISTRIBUTED LEDGER SYSTEMS	**9**
Blockchain Architecture – Consensus Mechanism – Proof-of-Work (PoW), Proof-of-Stake (PoS), and hybrid models – Pros and Cons - Directed Acyclic Graphs (DAGs) and emphasis on RESTful API design following open standards.	
UNITIII –SMART CONTRACTS, DAPPS AND BLOCKCHAIN INTEGRATION	**9**
Fundamentals of design, deployment, and management of smart contracts on Blockchain platforms - Decentralized Applications (dApps) - Integration of Blockchain back-ends with front-end systems - HTML5, semantic HTML, and ARIA/WCAG.	
UNIT IV –DIGITAL ASSET TOKENIZATION (NFTs) AND DECENTRALIZED STORAGE	**9**
Digital Asset Tokenization & NFTs - ERC-721/1155 standards - Metadata structuring with JSON-LD for interoperability - Oracle Services – Dynamic and real-time data - Decentralized Storage Systems – Storage Protocols – IPFS – Filecoin.	
UNIT V - SECURITY, DIGITAL IDENTITY, PEER-TO-PEER NETWORKING AND DECENTRALIZED GOVERNANCE	**9**
Security – Scalability – Interoperability with traditional systems – Digital Identity – Decentralized Identity (DIDs) self-sovereign identity principles, and W3C DID and verifiable credentials standards.	
TOTAL PERIODS: 45	

References:

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

Mapping of COs with POs / PSOs														
	Pos											PSOs		
COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
1	3	2	1	1	2	2	2	1	1	–	2	2	2	1
2	3	3	2	2	3	1	1	1	1	–	2	3	2	1
3	3	3	3	2	3	1	1	2	1	1	2	3	3	2
4	3	2	3	2	3	2	2	1	1	1	2	3	3	2
5	2	3	2	2	2	3	3	2	2	2	3	2	3	3
Avg.	**2.8**	**2.6**	**2.2**	**1.8**	**2.6**	**1.8**	**1.8**	**1.4**	**1.2**	**0.8**	**2.2**	**2.6**	**2.6**	**1.8**

PC25P09	Advanced Large Language Models	L	T	P	C
		3	0	0	3

Course Objectives:

- To understand the architecture, components, and foundational concepts of Large Language Models (LLMs).
- To learn various pre-training methods and language modeling approaches used in modern LLMs.
- To understand large-scale pretraining data preparation, language model training, and decoding strategies.
- To explore parameter-efficient tuning methods and ethical considerations in LLM deployment.
- To apply prompt engineering, alignment techniques, and evaluation methods for analyzing LLM performance.

Course Outcomes The students will be able to		**Knowledge Level**
CO1	**Explain** the architecture, components, and foundational concepts of Large Language Models, including Transformers, attention mechanisms, and encoder-decoder architectures.	**K2**
CO2	**Apply** language modeling and pre-training techniques such as autoregressive, autoencoder, masked language modeling, and denoising-based approaches in LLM development.	**K3**
CO3	**Analyze** large-scale pretraining data preparation, language model training processes, decoding strategies, and scaling techniques used in modern LLMs.	**K4**
CO4	**Evaluate** parameter-efficient tuning methods, explainability techniques, and ethical considerations for adapting and deploying LLMs.	**K5**
CO5	**Design and assess** prompt engineering, alignment strategies, in-context learning techniques, and evaluation methodologies for improving LLM performance across applications.	**K5**

UNIT I – BUILDING BLOCKS OF MODERN LLMs	**9**

Large language model basics, Large Language model architecture, Transformer Architecture, Attention all you need – Encoder – Decoder.

UNIT II – PRE-TRAINING METHODS	**9**
Pre-training and Language Modeling, Auto-Regressive Language Modelling and Auto-Encoder Language Modelling, Early experiments with decoder and auto-encoder pretraining - Transformer Decoding Pretraining by Denoising Task.	
UNIT III – LLM PRETRAINING METHODS	9
Masked Language Modeling, BERT Pretraining with Masked LM, Autoregressive LM and Masked LM, LLM pretraining data, preprocessing clean text, Scaling Up pretraining data using the web, Language Model (LM)-Decoding Strategies.	
UNIT IV – PARAMETER EFFICIENT TUNING METHODS	9
Categories of PETM – Prefix tuning, Prompt tuning - Adapters, Compacters - Layer freezing - Bias terms, Fine-tuning - diff pruning, Low Rank Adaptation (LoRA) – Advantages and Pitfalls, Explainability and LLMs, ethical considerations.	
UNIT V - IN-CONTEXT LEARNING, PROMPT ENGINEERING, ALIGNMENT AND EVALUATION METHODS	**9**
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. 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.	
TOTAL PERIODS: 45	

References:

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

E-Resources:

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

Mapping of COs with POs / PSOs

Pos												**PSOs**		
COs	1	2	3	4	5	6	7	8	9	10	11	1	2	3
1	3	2	1	1	2	1	1	1	1	–	2	2	2	1
2	3	3	2	2	3	1	1	1	1	–	2	3	3	1
3	3	3	2	3	3	1	1	1	1	1	2	3	3	1
4	3	3	2	2	3	2	3	1	1	1	2	2	3	2
5	3	3	3	3	3	2	2	2	2	1	3	3	3	3
Avg.	3.0	2.8	2.0	2.2	2.8	1.4	1.6	1.2	1.2	0.6	2.2	2.6	2.8	1.6

PC25P10	EDGE AND FOG COMPUTING	L	T	P	C
		3	0	0	3

Course Objectives:

- To understand the evolution of cloud, fog, and edge computing paradigms and their architectural requirements.
- To learn edge offloading strategies, optimization techniques, and resource management mechanisms for distributed systems.
- To understand resource federation, network slicing, and middleware technologies for edge environments.
- To explore data management, machine learning, analytics, and security mechanisms at the edge.
- To analyze emerging technologies and future trends in secure, intelligent, and sustainable edge computing.

Course Outcomes The students will be able to		**Knowledge Level**
CO1	**Explain** cloud, fog, and edge computing paradigms and their architectural models.	**K2**
CO2	**Analyze** edge offloading, system optimization, and performance management techniques in distributed environments.	**K3**
CO3	**Apply** resource federation, network slicing, and middleware technologies for efficient edge computing systems.	**K3**
CO4	**Design** intelligent edge analytics solutions using distributed machine learning and real-time processing frameworks.	**K4**
CO5	**Evaluate** advanced technologies, security mechanisms, and future trends in edge computing ecosystems.	**K3**

UNIT I – CLOUD TO EDGE COMPUTING PARADIGMS	**9**
Evolution from Cloud to Fog and Edge Computing - IoT Architecture Requirements and Latency Limitations - Fog vs Edge vs Cloud: Differences in Computation and Storage - Multi-access Edge Computing (MEC) Concepts - Multi-tier Architectural Models for Distributed Systems.	
UNIT II – EDGE OFFLOADING AND SYSTEM OPTIMIZATION	**9**
Edge Offloading Strategies and Placement Issues - Challenges in Latency, Energy Efficiency, QoS, and Privacy - Queuing Theory and System Performance Metrics - Optimization Techniques using Machine Learning for Slice Orchestration.	
UNIT III – RESOURCE FEDERATION AND NETWORK SLICING	**9**
Cloud-to-Fog-to-Things (C2F2T) Integration - Network Slicing for 5G/6G: Slice Types and Lifecycle - Resource Virtualization: SDN, NFV, and Container-Based Federation - Modeling and Simulation for Resource Management.	
UNIT IV – MIDDLEWARE FOR EDGE SYSTEMS	**9**
Middleware Design Goals: Heterogeneity, Discovery, Scalability – Service Orchestration at the Edge - Lightweight Containers: Docker and Kubernetes on Constrained Devices - Edge Cluster Formation and Node Coordination - Communication Protocols: MQTT, CoAP, and Microservices - Distributed Storage, Synchronization, and Fault Recovery.	
UNIT V – INTELLIGENCE, ANALYTICS, SECURITY AND FUTURE DIRECTIONS IN EDGE COMPUTING	**9**
Data Management at the Fog Layer: Caching, Placement, Lifecycle - Federated Learning and Distributed ML for Edge Analytics - Stream Processing and Real-Time Analytics Frameworks - Use Cases: Smart Cities, Healthcare, Smart Homes - Security Challenges: ML-Based Threat Detection, Behavioral Analysis - Privacy Techniques: Differential Privacy, Federated Anonymization. Advanced Technologies for Secure and Efficient Edge Computing: Blockchain for Fog Resource Security and Transparency - Software-defined Edge Networks and Intent-Based Networking - Green and Sustainable Edge Computing - Security and Privacy Challenges in Edge Environments. Future Directions and Emerging Trends: Edge AI and TinyML: Neural Compression, Quantization, Model Deployment - 6G Vision: Ultra-Low Latency, Terahertz Communications - Advanced Paradigms: Digital Twins, Serverless Edge, Cognitive Network.	

References:

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

Mapping of COs with POs / PSOs														
	Pos											PSOs		
COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
1	3	2	1	1	2	1	1	1	1	–	2	2	2	1
2	3	3	2	2	3	1	1	1	1	–	2	3	2	1
3	3	3	3	2	3	1	1	2	1	1	2	3	3	2
4	3	3	3	3	3	2	1	2	1	1	2	3	3	2
5	2	3	2	2	3	3	3	2	2	1	3	2	3	3
Avg.	**2.8**	**2.8**	**2.2**	**2.0**	**2.8**	**1.6**	**1.4**	**1.6**	**1.2**	**0.6**	**2.2**	**2.6**	**2.6**	**1.8**

PC25P11	GREEN COMPUTING AND SUSTAINABILITY	L	T	P	C
		3	0	0	3

Course Objectives:

- To understand the concepts, principles, and environmental impact of Green Computing.
- To learn energy-efficient hardware design, manufacturing, and software development practices.
- To understand sustainable software engineering methodologies and green algorithms.
- To explore green data center infrastructure and energy-efficient storage technologies.
- To analyze Green IT management practices, standards, regulations, and sustainability initiatives.

Course Outcomes The students will be able to		**Knowledge Level**
CO1	**Explain** the scope, benefits, challenges, and frameworks of Green Computing and Green IT.	**K2**
CO2	**Apply** green hardware and software practices for energy-efficient computing systems.	**K3**
CO3	**Analyze** sustainable software development methodologies and energy-aware software engineering techniques.	**K3**
CO4	**Evaluate** green data center and storage technologies for improving energy efficiency.	**K4**
CO5	**Assess** Green IT initiatives, standards, regulations, and sustainability reporting practices in organizations.	**K3**

UNIT I – INTRODUCTION TO GREEN COMPUTING	**9**

Scope - Environmental Impact of Computing - Need for Sustainable Computing – Power-Aware Resource Management - Green IT - OCED Green IT Framework – Greening IT - Benefits and Challenges - Quantifying IT Energy Efficiency.

UNIT II – GREEN HARDWARE	**9**
Life cycle of hardware - Reuse, Recycle and Dispose - Green Design and Manufacturing of Computers - Green Software - Energy-Saving Software Techniques.	
UNIT III – SUSTAINABLE SOFTWARE DEVELOPMENT	**9**
Attributes – Metrics – Methodology - Energy-aware Software Engineering - Green Algorithms.	
UNIT IV – GREEN DATA CENTER AND GREEN DATA STORAGE	**9**
Green Data Center: Infrastructure for Energy Efficiency – Server Power Management. Green Data Storage: Practices for Improving the Energy Efficiency of Data Storage - Optical Interconnects for Green Computers and Data Centers.	
UNIT V – MANAGING GREEN IT	**9**
Strategizing Green Initiatives – Implementation - Green IT Laws, Standards and Protocols - Regulatory Environment and IT Manufacturers. Nonregulatory Government Initiatives: Green Building Standards - Sustainability Reporting and Green Audits.	

References:

1. San Murugesan, "Harnessing Green IT: Principles and Practices", 1st Edition, Wiley, 2012.
2. Suyel Namasudra (Ed.), "Green and Sustainable Computing", 1st Edition, Elsevier, 2022.
3. Toby Velte, Anthony Velte & Robert Elsenpeter, "Green IT: Reduce Your Information System's Environmental Impact While Adding to the Bottom Line", 1st Edition, McGraw-Hill, 2008.
4. Bhuvan Unhelkar, "Green IT Strategies and Applications: Using Environmental Intelligence", 1st Edition, CRC Press, 2011.
5. Balamurugan Balusamy, Naveen Chilamkurti & Seifedine Kadry, "Green Computing in Smart Cities: Simulation and Techniques", 1st Edition, Springer Singapore, 2020.
6. Mike Halsey, "The Green IT Guide: How to Make Your IT Systems and Business Sustainable and Carbon Neutral", 1st Edition, Apress, 2025.

E-Resources

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

Mapping of COs with POs / PSOs													
	Pos											PSOs	
COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
1	2	2	1	1	2	3	2	1	1	–	2	2	1	2
2	3	3	2	2	3	2	1	1	1	1	2	3	2	2
3	3	3	2	2	3	2	1	1	1	1	2	3	2	2
4	3	3	3	3	3	2	1	1	1	1	2	3	2	2
5	2	3	2	2	2	3	3	2	2	2	3	2	2	3
Avg.	**2.6**	**2.8**	**2.0**	**2.0**	**2.6**	**2.4**	**1.6**	**1.2**	**1.2**	**1.0**	**2.2**	**2.6**	**1.8**	**2.2**

PC25P12	COGNITIVE COMPUTING	L	T	P	C
		3	0	0	3

Course Objectives:

- To understand the fundamentals of human cognition and cognitive computing systems.
- To explore cognitive architectures and learning models.
- To analyze the role of NLP in cognitive systems.
- To study cognitive agents, decision-making systems, and ethics.
- To examine real-world applications and emerging trends in cognitive computing.

Course Outcomes The students will be able to		**Knowledge Level**
CO1	**Explain** the fundamentals of cognitive computing and human cognition	**K2**
CO2	**Analyze** cognitive architectures and learning models	**K3**
CO3	**Apply** NLP techniques in cognitive systems	**K3**
CO4	**Evaluate** cognitive agents and decision-making systems with ethical considerations	**K4**
CO5	**Assess** Green IT initiatives, standards, regulations, and sustainability reporting practices in organizations.	**K3**

UNIT I – FOUNDATIONS OF COGNITIVE COMPUTING	**9**

Human Cognition – Introduction to Cognitive Computing – Elements of a Cognitive System – Design Principles for Cognitive Systems – Artificial Intelligence in Cognitive Computing – Natural Language Processing in Cognitive Computing – Machine Learning in Cognitive Computing

UNIT II – COGNITIVE ARCHITECTURES AND LEARNING MODELS	**9**

Cognitive Architectures: ACT-R, Soar, OpenCog – Knowledge Representation and Reasoning – Machine Learning Models in Cognitive Systems – Human-like Learning: Supervised Learning, Unsupervised Learning, Reinforcement Learning

UNIT III – NATURAL LANGUAGE PROCESSING IN COGNITIVE SYSTEMS	**9**
Role of NLP in a Cognitive System – NLP Pipeline: Part-of-Speech (POS) Tagging, Named Entity Recognition (NER), Parsing – Sentiment Analysis – Question Answering Systems – Semantic Web – Semantic Understanding: Ontologies and Knowledge Graphs	
UNIT IV – COGNITIVE AGENTS AND DECISION SYSTEMS	**9**
Cognitive Agents – Rule-based Agents – Expert Systems – Decision Support Systems – Emotion-aware Computing – Human-Computer Interaction – Cognitive Bias and Ethics in Decision Making	
UNIT V – COGNITIVE APPLICATIONS AND EMERGING TRENDS	**9**
Process of Building a Cognitive Application – IBM Watson as a Cognitive System – Business Implications of Cognitive Computing – Cognitive Computing Applications: Healthcare, Finance, Education, Legal, Robotics – Building a Cognitive Healthcare Application – Emerging Areas: Cognitive Cloud, Cognitive IoT, Cognitive Chatbots, Neuromorphic Computing and Brain-inspired Models	

References:

1. Velankar, Mahalle & Shinde, "Cognitive Computing for Machine Thinking", 1st Edition, CRC Press, 2024.
2. Judith S. Hurwitz, Marcia Kaufman & Adrian Bowles, "Cognitive Computing and Big Data Analytics", 1st Edition, Wiley, 2015.
3. Peter Fingar & Meghan-Kiffer Press, "Cognitive Computing: A Brief Guide for Game Changers", 1st Edition, 2015.
4. IBM Redbooks, "Building Cognitive Applications with IBM Watson Services", 1st Edition, IBM Redbooks, 2017.
5. Steven Bird, Ewan Klein & Edward Loper, "Natural Language Processing with Python", 1st Edition, O'Reilly Media, 2009.
6. Fuchun Sun, Qinghu Meng, Zhumu Fu & Bin Fang (Eds.), "Communications in Computer and Information Science (CCIS), Volume 1918", Springer Series, 2024.

E-Resources

- Prof. Sharba Bandyopadhyay, Prof. Rajlakshmi Guha | IIT Kharagpur
  Link https://onlinecourses.nptel.ac.in/noc22_ee122/preview
- Cognition and its Computation, IIT Kharagpur Prof. Sharba Bandyopadhyay, Prof. Rajlakshmi Guha
  Link "https://nptel.ac.in/courses/108105185"

Mapping of COs with POs / PSOs														
	Pos											**PSOs**		
COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
1	3	2	1	1	2	1	1	1	1	–	2	2	2	1
2	3	3	2	2	3	1	1	1	1	–	2	3	3	1
3	3	3	2	2	3	1	1	1	1	1	2	3	3	2
4	3	3	3	2	3	2	3	2	1	1	2	3	3	2
5	2	3	2	2	2	3	3	2	2	2	3	2	3	3
Avg.	**2.8**	**2.8**	**2.0**	**1.8**	**2.6**	**1.6**	**1.8**	**1.4**	**1.2**	**0.8**	**2.2**	**2.6**	**2.8**	**1.8**

PC25P13	AGENTIC AI	L	T	P	C
		3	0	0	3

Course Objectives:

- To understand the fundamentals of intelligent agents and agentic AI
- To analyze agent architectures, environments, and decision-making models
- To explore learning techniques in agent-based systems
- To study multi-agent collaboration and human-AI interaction
- To evaluate ethical issues and real-world applications of agentic AI

Course Outcomes The students will be able to		**Knowledge Level**
CO1	**Explain** the concepts of intelligent agents and agentic AI systems	**K2**
CO2	**Analyze** decision-making and planning techniques in agents	**K3**
CO3	**Apply** reinforcement learning and deep learning techniques in agent systems	**K3**
CO4	**Evaluate** multi-agent systems and human-AI collaboration models	**K4**
CO5	**Assess** ethical implications and design real-world agentic AI applications	**K3**

UNIT I – FOUNDATIONS OF INTELLIGENT AGENTS	**9**

Introduction to Agentic AI: History, Motivation, Applications – Agents and Environments: Sensors, Actuators, Environment Types – Types of Agents: Simple Reflex, Goal-Based, Utility-Based, Learning Agents – Architectures: Reactive, Deliberative, Hybrid, Subsumption

UNIT II – AGENT DECISION-MAKING AND PLANNING	**9**

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

UNIT III – LEARNING IN AGENTIC SYSTEMS	**9**
Reinforcement Learning: Q-Learning, SARSA – Value Iteration and Policy Iteration – Deep Reinforcement Learning: Deep Q-Networks (DQNs), Policy Gradient Methods, Actor-Critic Methods – Tools: OpenAI Gym, PettingZoo, TensorFlow	
UNIT IV – MULTI-AGENT SYSTEMS AND COLLABORATION	**9**
Multi-Agent Coordination: Cooperation and Competition – Agent Communication and Negotiation – Human-AI Collaboration: Interactive and Mixed-Initiative Systems – Platforms: NetLogo, RoboCup	
UNIT V – LLM AGENTS, EMBODIED COGNITION AND ETHICS	**9**
Language-based Agents: Tool Use, Planning, and Reasoning – Memory-Augmented and Situated Agents – Agents in Robotics and Smart Environments – Tools: Unity ML-Agents, Hugging Face Tools – Ethics in Agentic AI: AI Alignment, Fairness, Explainability – Social and Environmental Implications – Case Studies: Autonomous Vehicles, Conversational Agents	

References:

1. Stuart Russell & Peter Norvig, "Artificial Intelligence: A Modern Approach", 4th Edition, Pearson, 2021.
2. Yoav Shoham & Kevin Leyton-Brown, "Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations", 1st Edition, Cambridge University Press, 2009.
3. Richard S. Sutton & Andrew G. Barto, "Reinforcement Learning: An Introduction", 2nd Edition, 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

Mapping of COs with POs / PSOs														
	Pos											PSOs		
COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
1	3	2	1	1	2	1	1	1	1	–	2	2	2	1
2	3	3	2	2	3	1	1	1	1	1	2	3	3	1
3	3	3	3	2	3	1	1	1	1	1	2	3	3	2
4	3	3	3	3	3	2	3	3	2	1	2	3	3	3
5	2	3	2	2	2	3	3	2	2	2	3	2	3	3
Avg.	**2.8**	**2.8**	**2.2**	**2.0**	**2.6**	**1.6**	**1.8**	**1.6**	**1.4**	**1.0**	**2.2**	**2.6**	**2.8**	**2.0**

PC25P14	MIXED REALITY	L	T	P	C
		3	0	0	3

Course Objectives:

- To understand the fundamentals of Virtual Reality (VR) and Augmented Reality (AR)
- To explore current VR/AR technologies and devices
- To develop basic virtual environments and interactive applications
- To design 2D/3D applications using game engines
- To implement augmented reality applications using modern frameworks

Course Outcomes The students will be able to		**Knowledge Level**
CO1	**Explain** the concepts and technologies of VR and AR	**K2**
CO2	**Analyze** current VR/AR devices and interaction techniques	**K3**
CO3	**Develop** basic virtual environments and VR applications	**K3**
CO4	**Design** and implement 2D/3D applications using Unity	**K4**
CO5	**Build and deploy** augmented reality applications	**K3**

UNIT I – DEFINING VIRTUAL AND AUGMENTED REALITY	**9**

Defining Virtual Reality and Augmented Reality – Four Key Elements of Virtual Reality Experience – Types of Virtual Reality – Types of Augmented Reality – Applications of VR and AR in real-world scenarios

UNIT II – CURRENT STATE OF VIRTUAL AND AUGMENTED REALITY	**9**

Exploring the Current State of Virtual Reality – Available Form Factors and Features – Room-scale vs Stationary Experiences – Inside-out Tracking – Tracking Devices – Haptic Feedback – Audio Systems – Controllers – Exploring the Current State of Augmented Reality – Mobile AR Devices – AR Headsets and Glasses – Hand Tracking – Motion Controllers – Gesture and Whole-body Tracking

UNIT III – VIRTUAL ENVIRONMENT DEVELOPMENT	**9**

Interacting with Virtual Worlds – Interaction Techniques: Manipulation, Navigation, Communication – Creating Basic Objects using SketchUp – Grouping and Components – Development of Virtual Environments using SketchUp – Integration with Meta Quest 2 using VR Sketch – Navigation in Virtual Environments

UNIT IV – 2D/3D APPLICATION DEVELOPMENT USING UNITY	**9**

Getting Started with Unity – Installing Unity Hub and Unity Editor – Unity Interface – Creating Projects using Templates – Asset Store – Developing Simple 2D Games using Game Objects – Scripting using C# – Visual Scripting – Developing Simple 3D Games using Imported Assets – Building Applications for Desktop and Mobile Platforms

UNIT V – AUGMENTED REALITY DEVELOPMENT	**9**

Setting up AR Development Environment – Installing XR Plugins – AR Foundation Package – Creating Simple AR Scene – Importing Sample Assets – AR Session and AR Session Origin – AR Camera – AR Raycast Manager – Object Placement on Planes – Prefabs Creation – Building and Deploying AR Applications

References:

1. Jonathan Linowes, "Augmented Reality with Unity AR Foundation", 1st Edition, Packt Publishing, 2021.
2. William R. Sherman & Alan B. Craig, "Understanding Virtual Reality: Interface, Application, and Design", 2nd Edition, Morgan Kaufmann, 2018.
3. Paul Mealy, "Virtual & Augmented Reality For Dummies", 1st Edition, Wiley Publications, 2013.
4. Jessie Y. C. Chen (Ed.), "Virtual, Augmented and Mixed Reality: Applications and Case Studies", 1st Edition, Springer, 2019.
5. Rajiv Chopra, "Virtual and Augmented Reality", 1st Edition, Khanna Publishing House, 2021.

Mapping of COs with POs / PSOs														
	Pos											PSOs		
COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
1	3	2	1	1	2	1	1	1	1	–	2	2	2	1
2	3	3	2	2	3	1	1	1	1	–	2	3	2	1
3	3	3	3	2	3	1	1	2	1	1	2	3	3	2
4	3	3	3	3	3	1	1	2	2	1	2	3	3	2
5	3	3	3	2	3	1	1	2	2	1	3	3	3	2
Avg.	**3.0**	**2.8**	**2.4**	**2.0**	**2.8**	**1.0**	**1.0**	**1.6**	**1.4**	**0.6**	**2.2**	**2.8**	**2.6**	**1.6**

PC25P15	BLOCKCHAINS ARCHITECTURE AND DESIGN	L	T	P	C
		3	0	0	3

Course Objectives:

- To understand the fundamental concepts and architecture of blockchain
- To analyze Bitcoin and various consensus mechanisms
- To explore smart contracts and Ethereum platform
- To study Web3 technologies and enterprise blockchain frameworks
- To evaluate blockchain applications, challenges, and future directions

Course Outcomes The students will be able to		**Knowledge Level**
CO1	**Explain** blockchain fundamentals and cryptographic concepts	**K2**
CO2	**Analyze** Bitcoin transactions and consensus mechanisms	**K3**
CO3	**Evaluate** different blockchain protocols and architectures	**K3**
CO4	**Develop and deploy** smart contracts using Ethereum platform	**K4**
CO5	**Assess** blockchain applications and identify future research challenges	**K3**

UNIT I – INTRODUCTION TO BLOCKCHAIN	**9**

Introduction to Blockchain – Layers of Blockchain – Types of Blockchain – Generic Elements of Blockchain – Basic Cryptographic Primitives used in Blockchain – Hash Functions – Merkle Tree – Structure of a Block – Linking Blocks in the Blockchain – Mining the Block – Validating a New Block – Chain Formation and Selection

UNIT II – BITCOIN AND CONSENSUS MECHANISMS	**9**

Bitcoin Overview – Bitcoin Scripts – Bitcoin Transactions – Bitcoin Network and Payments – Consensus Mechanism – Types of Consensus Mechanisms

UNIT III – CONSENSUS PROTOCOLS AND BLOCKCHAIN TYPES	**9**
Proof of Work (PoW) – Consensus Protocols for Permissionless Blockchain Environments – Consensus Protocols for Permissioned Blockchain Environments	
UNIT IV – SMART CONTRACTS AND BLOCKCHAIN PLATFORMS	**9**
Smart Contracts – Ricardian Contracts – Introduction to Ethereum – Ethereum Network – Components of Ethereum – Ethereum Virtual Machine (EVM) – Ethereum Development Environment – Solidity Programming Language – Web3 Concepts – Tools: Geth, Truffle – Enterprise Blockchain Platforms: Hyperledger – Hyperledger Fabric – Hyperledger Sawtooth – Corda	
UNIT V – BLOCKCHAIN APPLICATIONS AND FUTURE DIRECTIONS	**9**
Blockchain Applications in Government: Border Control, Voting, Citizen Identification – Blockchain in Healthcare – Blockchain in Finance: Insurance, Post-Trade Settlement, Financial Crime Prevention – Scalability Issues – Privacy Concerns – Limitations of Current Blockchain Platforms – Emerging Trends – Open Research Challenges	

References:

1. Imran Bashir, "Mastering Blockchain: A Deep Dive into Distributed Ledgers, Consensus Protocols, Smart Contracts, DApps, Cryptocurrencies, Ethereum, and More", 3rd Edition, Packt Publishing, 2020.
2. Andreas M. Antonopoulos & David A. Harding, "Mastering Bitcoin: Programming the Open Blockchain", 3rd Edition, O'Reilly Media, 2023.
3. Ahmed Banafa, "Blockchain Technology and Applications", 1st Edition, River Publishers, 2024.
4. Melanie Swan, "Blockchain: Blueprint for a New Economy", 1st Edition, O'Reilly Media, 2015.
5. Bikramaditya Singhal, Gautam Dhameja & Priyansu Sekhar Panda, "Beginning Blockchain: A Beginner's Guide to Building Blockchain Solutions", 1st Edition, Apress, New York, 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
   https://www.nptel.ac.in/courses/106105184 by Prof.S.Chakraborty and P.Jayachandran,IIT Kharagpur

Mapping of COs with POs / PSOs														
	Pos											PSOs		
COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
1	3	2	1	1	2	1	1	1	1	–	2	2	2	1
2	3	3	2	2	3	1	1	1	1	–	2	3	2	1
3	3	3	2	2	3	1	1	1	1	1	2	3	3	1
4	3	3	3	3	3	1	1	2	1	1	2	3	3	2
5	2	3	2	2	2	3	3	2	2	2	3	2	3	3
Avg.	**2.8**	**2.8**	**2.0**	**2.0**	**2.6**	**1.4**	**1.4**	**1.4**	**1.2**	**0.8**	**2.2**	**2.6**	**2.6**	**1.6**

PC25P16	HUMAN-CENTERED AI	L	T	P	C
		3	0	0	3

Course Objectives:

- To introduce the fundamental concepts, principles, and design philosophies of Human-Centered Artificial Intelligence (HCAI).
- To develop an understanding of human-AI interaction, human factors, and the societal impact of intelligent systems.
- To provide knowledge of fairness, bias mitigation, privacy preservation, and trustworthy AI methodologies.
- To familiarize students with HCAI frameworks, design metaphors, and approaches for developing reliable and safe AI systems.
- To enable learners to evaluate governance structures, ethical guidelines, and regulatory mechanisms for responsible AI deployment.

Course Outcomes The students will be able to		**Knowledge Level**
CO1	**Explain** the fundamental concepts, principles, design thinking approaches, and human factors involved in Human-Centered Artificial Intelligence.	**K2**
CO2	**Analyze** fairness, bias, and ethical challenges in AI systems and evaluate their impact on individuals and society.	**K3**
CO3	**Apply** privacy-preserving techniques and security mechanisms to support trustworthy and responsible AI development.	**K3**
CO4	**Evaluate** Human-Centered AI frameworks, automation levels, and design metaphors for developing reliable, safe, and user-centric intelligent systems.	**K4**
CO5	**Assess** governance structures, regulatory frameworks, and software engineering best practices for ensuring responsible deployment of AI technologies.	**K3**

UNIT I – INTRODUCTION TO HUMAN-CENTERED AI AND HUMAN FACTORS	**9**

Overview of AI Development – Defining Human-Centered AI (HCAI) – Difference between Traditional AI and HCAI – Process of Design Thinking – Cognitive Models – People and Computers: Are They in the Same Category? – HCAI Principles – Impact of AI on Society – Importance of Human Values.

UNIT II – FAIRNESS AND BIAS	**9**
Sources of Bias – Group Fairness – Exploratory Data Analysis – Limitation of a Dataset – Group Fairness and Individual Fairness – Counterfactual Fairness.	
UNIT III – PRIVACY PRESERVATION	**9**
Attack Models – Privacy Preserving Learning – Differential Privacy – Federated Learning.	
UNIT IV – HCAI FRAMEWORK AND DESIGN METAPHORS	**9**
Levels of Automation – Defining Reliable, Safe, and Trustworthy Systems – Two-Dimensional HCAI Framework: Life-Critical Systems – Golden Rules for Design – Application Examples – Science and Innovation Goals – Intelligent Agents – Tele-bots – Assured Autonomy: Design, Control Centers – Social Robots: Active Appliances – Enhancing Human Capabilities.	
UNIT V – HCAI GOVERNANCE STRUCTURES	**9**
Ethical Implications of Emerging AI Technologies – Software Engineering Best Practices – Establishing Safety Cultures in Organizations – Independent Oversight and Certification – Role of Government Regulation.	

References:

1. Ben Shneiderman, "Human-Centered AI: A New Synthesis", 1st Edition, Oxford University Press, 2022.
2. Virginia Dignum, "Responsible Artificial Intelligence: How to Develop and Use AI in a Responsible Way",1st Edition, Springer Nature, 2019.
3. Christoph Molnar, "Interpretable Machine Learning", 1st Edition, Lulu Press, 2019.
4. Ben Shneiderman, Catherine Plaisant, Maxine Cohen, Steven Jacobs & Niklas Elmqvist, "Designing the User Interface: Strategies for Effective Human-Computer Interaction", 6th Edition, Pearson Education, 2018.
5. Jenny Preece, Helen Sharp & Yvonne Rogers, "Interaction Design: Beyond Human-Computer Interaction", 5th Edition, Wiley Student Edition, 2019.
6. Alan Dix, Janet Finlay, Gregory Abowd & Russell Beale, "Human-Computer Interaction", 3rd Edition, Pearson Education, 2004.
7. Paula Boddington, "AI Ethics", 1st Edition, Springer, 2023.
8. Luciano Floridi, "The Ethics of Artificial Intelligence: Principles, Challenges, and Opportunities", 1st Edition, 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://arxiv.org/pdf/2004.09456

Mapping of COs with POs / PSOs

	Pos											**PSOs**		
COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
1	2	2	1	1	2	2	2	1	1	–	2	2	1	2
2	2	3	2	2	2	3	3	2	2	1	2	2	2	3
3	3	3	2	2	3	2	2	1	1	1	2	3	2	2
4	3	3	3	3	3	2	3	2	2	1	2	3	3	3
5	2	3	2	2	2	3	3	2	2	2	3	2	2	3
Avg.	**2.4**	**2.8**	**2.0**	**2.0**	**2.4**	**2.4**	**2.6**	**1.6**	**1.6**	**1.0**	**2.2**	**2.4**	**2.0**	**2.6**

PC25P17	VIBE CODING	L	T	P	C
		3	0	0	3

Course Objectives:

- To introduce the concepts, principles, and workflows of AI-assisted coding and Human-AI collaboration in software development.
- To provide practical knowledge of AI-assisted coding tools, platforms, and prompt engineering techniques for application development.
- To develop the ability to apply ethical, responsible, and quality-focused practices in AI-augmented software engineering.
- To explore the evolving roles of developers and the impact of AI across the software development lifecycle.
- To enable learners to design, build, deploy, and evaluate AI-assisted applications using modern development tools and collaborative workflows.

Course Outcomes The students will be able to		**Knowledge Level**
CO1	**Explain** the concepts, principles, workflows, and collaborative practices involved in AI-assisted coding and Vibe Coding.	**K2**
CO2	**Apply** AI-assisted coding tools, prompt engineering techniques, and workflow design strategies to support software development activities.	**K3**
CO3	**Analyze** ethical, security, reliability, and quality considerations associated with AI-generated code and Human-AI collaboration.	**K3**
CO4	**Evaluate** AI-augmented development practices, evolving developer roles, and real-world AI application development approaches.	**K4**
CO5	**Design, develop, and deploy** AI-assisted software applications by integrating appropriate tools, workflows, and governance practices.	**K3**

UNIT I – INTRODUCTION TO VIBE CODING AND AI-ASSISTED DEVELOPMENT	**9**
Introduction to Vibe Coding – Basic Prompting Techniques for Coding Tasks – Roles and Responsibilities in Human-AI Collaboration – AI Assistant – Prompt Engineer – Developer – Designing Simple AI-Augmented Workflows – Interaction Between AI-Generated Code and Developer Feedback – Overview of AI-Assisted Coding Tools – Cursor, v0, Bolt, etc. – Features and Use Cases of Each Tool – Setting Up and Navigating AI-Coding Platforms.	
UNIT II – AI TOOLS, PROMPT DESIGN, AND ETHICAL CODING PRACTICES	**9**
Building Simple Web Applications with AI Integration – Workflow Prototyping Using AI Tools – Vibe Coder's Toolkit: Prompts, Context, and Flow – Prompt Optimization Techniques – Human Oversight in AI Coding Systems – Ethical and Responsible AI Usage – Challenges and Best Practices in AI-Augmented Development.	
UNIT III – THE FUTURE OF CODING ROLES AND CODE QUALITY	**9**
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 – 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.	
UNIT IV – APPLIED PROJECTS AND REAL-WORLD AI APPLICATIONS	**9**
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.	

UNIT V – ADVANCED AI-DRIVEN WEB APPLICATION DEPLOYMENT	9

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.

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, March 26, 2025.
3. Amit Iyer, Beginner's Guide to Vibe Coding: Exploring Features, Use Cases and Understanding Vibe Coding, Paperback, March 19, 2025.
4. Gene Kim and Steve Yegge, Vibe Coding: Building Production-grade Software with GenAI, Chat, Agents, and Beyond, Paperback, October 21, 2025.
5. Addy Osmani, Beyond Vibe Coding, O'Reilly Media, Inc., August 2025.

Mapping of COs with POs / PSOs														
	Pos											PSOs		
COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
1	2	2	1	1	2	1	1	2	2	1	2	2	2	3
2	3	3	2	2	3	1	1	2	2	1	2	3	3	3
3	3	3	2	2	2	2	3	2	2	1	2	3	2	3
4	3	3	3	2	3	2	2	3	3	2	3	3	3	3
5	3	3	3	3	3	2	3	2	3	2	3	3	3	3
Avg.	**2.8**	**2.8**	**2.2**	**2.0**	**2.6**	**1.6**	**2.0**	**2.2**	**2.4**	**1.4**	**2.4**	**2.8**	**2.6**	**3.0**

PC25P18	Federated Learning	L	T	P	C
		3	0	0	3

Course Objectives:

- To understand the concepts, architecture, and workflow of Federated Learning and its applications.
- To learn federated learning algorithms and techniques for handling distributed and heterogeneous data.
- To analyze privacy, security, and robustness challenges in federated learning systems.
- To understand system-level and communication challenges in deploying federated learning on edge devices.
- To explore practical applications, frameworks, and emerging research directions in federated learning.

Course Outcomes The students will be able to		**Knowledge Level**
CO1	**Explain** the fundamentals, architecture, types, and workflow of Federated Learning.	**K2**
CO2	**Apply** Federated Learning algorithms to build decentralized models in heterogeneous environments.	**K3**
CO3	**Analyze** privacy and security issues in Federated Learning and evaluate mitigation techniques.	**K3**
CO4	**Assess** system and communication challenges and implement efficient FL strategies.	**K4**
CO5	Design and evaluate Federated Learning applications using modern frameworks and emerging techniques.	**K3**

UNIT I – INTRODUCTION TO FEDERATED LEARNING	**9**

Definition and Scope – Motivation: Data Privacy, Decentralization, Compliance (GDPR, HIPAA) – Centralized vs. Federated vs. Distributed Learning – Types of FL: Horizontal, Vertical, and Transfer FL – Real-World Applications: Mobile Keyboard (GBoard), Healthcare, IoT – FL System Architecture – Federated Learning Workflow and Lifecycle – Communication Protocols.

UNIT II – FEDERATED LEARNING ALGORITHMS	**9**
Federated Averaging (FedAvg) – Federated SGD (FedSGD) – FedProx for handling system heterogeneity – Personalized Federated Learning (pFedMe, FedPer, FedBN) – FL with Non-IID Data – Aggregation and Weight Update Strategies – Convergence Issues – Performance Metrics.	
UNIT III – PRIVACY AND SECURITY IN FEDERATED LEARNING	**9**
Security Threats in FL – Differential Privacy (DP): Concept, Application in FL – Secure Multiparty Computation (SMC) – Homomorphic Encryption (Overview) – Secure Aggregation Protocols – Byzantine Robustness – Model Inversion Attacks – Data Poisoning – Mitigation Strategies.	
UNIT IV – SYSTEM AND COMMUNICATION CHALLENGES	**9**
Device Heterogeneity – Intermittent Availability – Resource Constraints (Memory, Battery, Network) – Client Selection Strategies – Communication Efficiency: Quantization, Compression, Sparsification – Dropout and Partial Client Participation – FL over Edge Devices – Energy-aware FL.	
UNIT V – APPLICATIONS, FRAMEWORKS, PROJECT WORK AND FUTURE DIRECTIONS	**9**
FL Applications in Healthcare, Finance, Smart City, Recommender Systems – Case Studies – Tools and Frameworks: TensorFlow Federated, PySyft, Flower – FL Simulators. 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.	

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.

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)

Mapping of COs with POs / PSOs														
	Pos											**PSOs**		
COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
1	3	2	1	1	2	1	1	1	1	–	2	2	2	1
2	3	3	2	2	3	1	1	1	1	1	2	3	3	1
3	3	3	2	2	3	2	3	1	1	1	2	3	3	2
4	3	3	3	3	3	2	2	2	1	1	2	3	3	2
5	3	3	3	2	3	2	2	2	2	1	3	3	3	2
Avg.	**3.0**	**2.8**	**2.2**	**2.0**	**2.8**	**1.6**	**1.8**	**1.4**	**1.2**	**0.8**	**2.2**	**2.8**	**2.8**	**1.6**

PC25P19	Deep Learning for Computer Vision	L	T	P	C
		3	0	0	3

Course Objectives:

- To understand image fundamentals, feature extraction techniques, and visual matching methods used in computer vision.
- To learn deep learning concepts and convolutional neural network architectures for visual recognition tasks.
- To analyze CNN-based techniques for object recognition, detection, and segmentation.
- To understand recurrent neural networks, attention models, and vision-language applications.
- To explore deep generative models and recent trends in deep learning for computer vision.

Course Outcomes The students will be able to		**Knowledge Level**
CO1	**Apply** image processing fundamentals and feature extraction techniques for visual analysis.	**K2**
CO2	**Analyze** deep learning architectures and convolutional neural networks for computer vision applications.	**K3**
CO3	**Design** and evaluate CNN-based solutions for recognition, detection, and segmentation tasks.	**K3**
CO4	**Design** and evaluate CNN-based solutions for recognition, detection, and segmentation tasks.	**K4**
CO5	**Assess** deep generative models and emerging trends in computer vision research and applications.	**K3**

UNIT I – INTRODUCTION TO IMAGE FUNDAMENTALS AND VISUAL MATCHING	**9**

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

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

UNIT II – CONVOLUTIONAL NEURAL NETWORKS (CNN)	**9**

Introduction to CNNs – Evolution of CNN Architectures – AlexNet – ZFNet – VGG – InceptionNets – ResNets – DenseNets – Visualization of Kernels – Backpropagation to image/Deconvolution Methods – Deep Dream – Hallucination – Neural Style Transfer – CAM – Grad-CAM – Grad CAM++ – Recent Methods (IG, Segment-IG, SmoothGrad).

UNIT III – APPLICATIONS OF CNN	**9**

CNNs for Recognition and Verification (Siamese Networks, Triplet Loss, Contrastive Loss, Ranking Loss) – CNNs for Detection: Background of Object Detection – R-CNN – Fast R-CNN – Faster R-CNN – YOLO – SSD – RetinaNet – CNNs for Segmentation: FCN – SegNet – U-Net – Mask-RCNN.

UNIT IV – RECURRENT NEURAL NETWORKS (RNN) AND ATTENTION MODELS	**9**

Introduction to RNNs – CNN and RNN Models for Video Understanding: 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.

UNIT V – DEEP GENERATIVE MODELS, APPLICATIONS, VARIANTS AND RECENT TRENDS OF DEEP LEARNING IN COMPUTER VISION	**9**

Deep Generative Models: GANs – VAEs – Other Generative Models: PixelRNNs, NADE, Normalizing Flows - Applications: Image Editing – Inpainting – Superresolution - 3D Object Generation – Security - Variants: CycleGANs – Progressive GANs – StackGANs - Pix2Pix.

Recent Trends of Deep Learning in Computer Vision: Zero-shot – One-shot – Few-shot Learning – Self-supervised Learning – Reinforcement Learning in Vision – Applications.

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/

Mapping of COs with POs / PSOs

	Pos											**PSOs**		
COs	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**1**	**2**	**3**
1	3	2	1	1	2	1	1	1	1	–	2	2	2	1
2	3	3	2	2	3	1	1	1	1	1	2	3	3	1
3	3	3	2	2	3	2	3	1	1	1	2	3	3	2
4	3	3	3	3	3	2	2	2	1	1	2	3	3	2
5	3	3	3	2	3	2	2	2	2	1	3	3	3	2
Avg.	**3.0**	**2.8**	**2.2**	**2.0**	**2.8**	**1.6**	**1.8**	**1.4**	**1.2**	**0.8**	**2.2**	**2.8**	**2.8**	**1.6**