POSTGRADUATE DEGREE

COURSE

M. Tech. I to IV Semester

Computer Science

University of Technology

Vatika Road, Jaipur Rajasthan 303903

Program Outcomes (POs)

PO 1	Engineering Knowledge: Apply knowledge of mathematics, science, and Civil, Computer Science, Electronics & Communication, Electrical, Mechanical engineering to solve numerical techniques problems.
PO 2	Problem Analysis: Identify, formulate, and solve Civil, Computer Science, Electronics & Communication, Electrical, Mechanical engineering problems.
PO 3	Design/Development of Solutions: Design and develop Civil, Computer Science, Electronics & Communication, Electrical, Mechanical engineering systems, components, or processes.
PO 4	Conduct Investigations: Conduct experiments and simulations to analyze Civil, Computer Science, Electronics & Communication, Electrical, Mechanical engineering problems.
PO 5	Modern Tool Usage: Use modern tools and software to analyze and solve Civil, Computer Science, Electronics & Communication, Electrical, Mechanical engineering problems.
PO 6	The Engineer and Society: Apply Civil, Computer Science, Electronics & Communication, Electrical, Mechanical engineering solutions to societal problems.
PO 7	Environment and Sustainability: Understand the impact of Civil, Computer Science, Electronics & Communication, Electrical, Mechanical engineering solutions on the environment.
PO 8	Ethics: Apply ethical principles to civil engineering practice.
PO 9	Individual and Team Work: Work effectively as an individual and in teams to solve civil engineering problems.
PO 10	Communication: Communicate Civil, Computer Science, Electronics & Communication, Electrical, Mechanical engineering ideas and solutions effectively.
PO 11	Project Management: Manage civil engineering projects and resources effectively.
PO 12	Lifelong Learning: Engage in lifelong learning to stay current with Civil, Computer Science, Electronics & Communication, Electrical, Mechanical engineering developments.

Program Specific Outcomes (PSOs):

1. **PSO 1:** Demonstrate a thorough understanding of computer science fundamentals, algorithms, data structures, and programming languages to develop efficient software solutions.
2. **PSO 2:** Apply modern software engineering principles and tools to design, develop, test, and deploy complex software applications, ensuring quality, performance, and scalability.
3. **PSO 3:** Utilize computational thinking, problem-solving skills, and emerging technologies in fields like Artificial Intelligence, Data Science, and Cloud Computing to tackle real-world challenges.

Program Educational Objectives (PEOs):

1. **PEO 1:** Prepare graduates with a strong foundation in computer science and mathematics to solve complex computing problems and develop innovative solutions.
2. **PEO 2:** Equip graduates with practical and technical knowledge in software development, system administration, networking, and data management to excel in diverse computing roles.
3. **PEO 3:** Foster communication, teamwork, and ethical responsibility in graduates, preparing them for leadership roles in the global technology industry.
4. **PEO 4:** Cultivate an ability in graduates to adapt to emerging technologies and engage in lifelong learning to remain competitive in the rapidly evolving field of computer science.

Course Overview and Objectives:

The Computer Science program is designed to provide students with a comprehensive understanding of computational theory, programming languages, data structures, software engineering, and cutting-edge technologies like Artificial Intelligence, Machine Learning, and Cloud Computing. The objective of this program is to equip students with the knowledge and skills to analyze, design, and implement complex software systems, solve computational problems efficiently, and adapt to technological advancements in the ever-evolving field of computer science. Students will also develop critical thinking, teamwork, and communication abilities to work in interdisciplinary environments and contribute effectively to the technology-driven world.

Scheme of

POSTGRADUATE DEGREE COURSE

M. Tech.

Computer Science

UNIVERSITY OF TECHNOLOGY, JAIPUR
Scheme and Syllabus
M.Tech 1st Year, I Sem: Computer Science

Teaching and Examination Scheme
ACADEMIC SESSION 2023-2024

S. No.	Course Name	Contact Hours per Week			Marks				Cr
		L	T	P	Exam Hrs	IA	ETE	Total	
1	Advanced Data Structures	5	0	0	3	25	100	125	5
2	Software System Design	5	0	0	3	25	100	125	5
3	Critical System Design	5	0	0	3	25	100	125	5
4	Information System Security	5	0	0	3	25	100	125	5
5	Software System Lab	0	0	4		60	40	100	2
	Total	**20**	**0**	**4**		**160**	**440**	**600**	**22**

***L:** Lecture, **T:** Tutorial, **P:** Practical, **Cr:** Credits*
***ETE:** End Term Exam, **IA:** Internal Assessment*
***: Not to be considered for award of division*

UNIVERSITY OF TECHNOLOGY, JAIPUR
Scheme and Syllabus
M.Tech 1st Year, II Sem: Computer Science

Teaching and Examination Scheme
ACADEMIC SESSION 2023-2024

S. No.	Course Name	Contact Hours per Week			Marks				Cr
		L	T	P	Exam Hrs	IA	ETE	Total	
1	Advanced Database Management Systems	5	0	0	3	25	100	125	5
2	Design of Embedded Systems	5	0	0	3	25	100	125	5
3	Distributed Algorithms	5	0	0	3	25	100	125	5
4	Advanced Computer Graphics	5	0	0	3	25	100	125	5
5	Advanced Database Lab	0	0	4		60	40	100	2
	Total	**20**	**0**	**4**		**160**	**440**	**600**	**22**

L: *Lecture,* ***T:*** *Tutorial,* ***P:*** *Practical,* ***Cr:*** *Credits*
ETE: *End Term Exam,* ***IA:*** *Internal Assessment*
**: Not to be considered for award of division*

UNIVERSITY OF TECHNOLOGY, JAIPUR
Scheme and Syllabus
M.Tech 2nd Year, III Sem: Computer Science

Teaching and Examination Scheme
ACADEMIC SESSION 2023-2024

S. No.	Course Name	Contact Hours per Week			Marks				Cr
		L	T	P	Exam Hrs	IA	ETE	Total	
1	Parallel and Distributed Computing	5	0	0	3	25	100	125	5
2	Network System Design	5	0	0	3	25	100	125	5
3	Seminar	0	0	0		90	60	150	6
4	Dissertation- I	0	0	0		100	0	100	4
	Total	**10**	**0**	**0**		**240**	**260**	**500**	**20**

L: *Lecture,* ***T:*** *Tutorial,* ***P:*** *Practical,* ***Cr:*** *Credits*
ETE: *End Term Exam,* ***IA:*** *Internal Assessment*
**: Not to be considered for award of division*

UNIVERSITY OF TECHNOLOGY, JAIPUR
Scheme and Syllabus
M.Tech 2nd Year, IV Sem: Computer Science

Teaching and Examination Scheme
ACADEMIC SESSION 2023-2024

S. No	Course Name	Contact Hours per Week			Marks				Cr
		L	T	P	Exam Hrs	IA	ETE	Total	
1	Dissertation-II	0	0	0		500	0	500	16
	Total					**500**	**0**	**500**	**16**

L: *Lecture,* ***T:*** *Tutorial,* ***P:*** *Practical,* ***Cr:*** *Credits*
ETE: *End Term Exam,* ***IA:*** *Internal Assessment*
**: Not to be considered for award of division*

UNIVERSITY OF TECHNOLOGY, JAIPUR

Syllabus of

POSTGRADUATE DEGREE COURSE

M. Tech

Computer Science

UNIVERSITY OF TECHNOLOGY, JAIPUR
Scheme and Syllabus
M.Tech 1st Year, I Sem: Computer Science

Teaching and Examination Scheme
ACADEMIC SESSION 2023-2024

S. No.	Course Name	Contact Hours per Week			Marks				Cr
		L	T	P	Exam Hrs	IA	ETE	Total	
1	Advanced Data Structures	5	0	0	3	25	100	125	5
2	Software System Design	5	0	0	3	25	100	125	5
3	Critical System Design	5	0	0	3	25	100	125	5
4	Information System Security	5	0	0	3	25	100	125	5
5	Software System Lab	0	0	4		60	40	100	2
	Total	**20**	**0**	**4**		**160**	**440**	**600**	**22**

L: *Lecture,* ***T:*** *Tutorial,* ***P:*** *Practical,* ***Cr:*** *Credits*
ETE: *End Term Exam,* ***IA:*** *Internal Assessment*
**: Not to be considered for award of division*

UNIVERSITY OF TECHNOLOGY, JAIPUR

1st Year - I Semester: M. Tech. (Computer Science)

Sub: Advanced Data Structures

Credit: 5 **Max. Marks**: 125(IA:25, ETE:100)

5L+0T+0P **End Term Exam**: 3 Hours

S.No.	CONTENTS	CONTACT HOURS
1	**Introduction**: Objective, scope, and outcome of the course.	01
2	**Advanced Trees**: AVL trees, Red-Black trees, B-trees, B+ trees, Splay trees, Segment trees, Range trees, Fenwick trees, K-d trees, Quad-trees, Applications of advanced trees in graphics, databases, and machine learning.	08
3	**Hashing Techniques**: Perfect hashing, Universal hashing, Dynamic hashing, Bloom filters, Cuckoo hashing, Cryptographic hash functions (SHA, MD5), Consistent hashing, Hashing in distributed systems, Applications in databases and caching systems.	08
4	**Graph Algorithms**: Shortest paths (Dijkstra, Floyd-Warshall, A*), Minimum spanning trees (Prim's, Kruskal's), Maximal matching, Network flow algorithms (Ford-Fulkerson, Edmonds-Karp), Bipartite graph matching, Eulerian and Hamiltonian graphs, Planarity testing, Graph coloring, Dynamic graph algorithms.	08
5	**String Matching and Pattern Matching**: KMP, Boyer-Moore, Rabin-Karp, Suffix arrays, Suffix trees, Tries, Z-algorithm, Aho-Corasick, Longest common substring, Applications in bioinformatics, plagiarism detection, and text search.	08
6	**Advanced Data Structures for Problem Solving**: Persistent data structures, Union-find, Skip lists, Tries, Dynamic segment trees, DSU with path compression, Dynamic programming on trees, Applications in computational geometry and real-time systems.	08

Text/Reference Books

1. Thomas H. Cormen et al., *Introduction to Algorithms*, MIT Press.
2. Mark Allen Weiss, *Data Structures and Algorithm Analysis in C++*, Pearson.
3. Michael T. Goodrich, *Data Structures and Algorithms in Java*, Wiley.
4. Peter Brass, *Advanced Data Structures*, Cambridge University Press.
5. Robert Sedgewick, *Algorithms in C++*, Addison-Wesley.

Course Outcomes (COs)

CO	Description
CO1	Analyze and implement advanced tree and graph structures for real-world applications.
CO2	Apply and evaluate advanced hashing techniques for secure and efficient data management.
CO3	Design and optimize algorithms for graph-related problems in various domains.
CO4	Implement string processing techniques for efficient pattern matching in complex systems.
CO5	Develop memory-efficient solutions using advanced data structures for system-level programming.

Course Description

The course "Advanced Data Structures" focuses on the study and implementation of complex data structures that are essential for solving real-world computing problems. Students will learn to analyze, design, and implement advanced tree, graph, and hashing structures, with a focus on efficient algorithms and memory management for system-level programming.

Course Objectives

1. To provide an in-depth understanding of advanced data structures such as trees, graphs, and hashing techniques.
2. To enable students to design efficient algorithms for real-world problems involving large datasets.
3. To apply advanced data structures and algorithms for solving domain-specific challenges in computer science.
4. To improve students' problem-solving and programming skills, with a focus on memory optimization and system-level solutions.

Mapping Table
The mapping table uses a scale of 1 to 3 to indicate the level of correlation between COs and POs/PSOs, where:

1. Slight (Low)
2. Moderate (Medium)
3. Substantial (High)

CO-PO Mapping

Course Outcomes (CO)	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	3	2	3	2	1	1	3	2	1	2
CO2	3	3	3	2	3	2	1	1	3	2	1	2
CO3	3	3	3	3	3	2	2	1	3	2	1	2
CO4	2	3	3	2	3	2	2	1	3	2	1	2
CO5	3	3	3	3	3	3	1	2	3	3	2	3

CO-PSO Mapping

Course Outcomes (CO)	PSO1	PSO2	PSO3
CO1	3	3	2
CO2	3	3	2
CO3	3	3	3
CO4	3	3	2
CO5	3	3	3

UNIVERSITY OF TECHNOLOGY, JAIPUR

1st Year - I Semester: M. Tech. (Computer Science)

Sub: Software System Design

Credit: 5 **Max. Marks**: 125(IA:25, ETE:100)

5L+0T+0P **End Term Exam**: 3 Hours

S.No.	CONTENTS	CONTACT HOURS
1	**Introduction**: Objective, scope, and outcome of the course.	01
2	**Software Design Principles**: SOLID principles, Design patterns (Singleton, Factory, Observer, Decorator, Strategy, Proxy), Agile design, Layered architecture, Domain-driven design, MVC, Aspect-oriented programming (AOP), Clean code practices, Dependency injection.	07
3	**Requirements Engineering**: Requirement elicitation techniques, Functional and non-functional requirements, Use cases, User stories, Prototyping, Requirement traceability, Prioritization techniques, Requirement validation and analysis, Agile vs. traditional requirements engineering.	07
4	**Software Architecture**: Architectural patterns (Monolithic, Microservices, Event-driven, Layered), Component-based architecture, Service-oriented architecture, API design, Scalability and performance considerations, Secure architectural design, Cloud-native architecture, Design for maintainability.	08
5	**Software Metrics and Estimation**: Function point analysis, LOC metrics, Cyclomatic complexity, Halstead complexity measures, Software quality metrics (ISO/IEC 25010), Effort estimation models (COCOMO, Agile estimation), Risk management in estimation.	07
6	**Software Testing Strategies**: Unit testing, Integration testing, System testing, Acceptance testing, Regression testing, TDD, BDD, Automated testing tools (JUnit, Selenium, Postman), Performance testing (JMeter), Security testing strategies.	07

Text/Reference Books

1. Ian Sommerville, *Software Engineering*, Pearson.
2. Len Bass, *Software Architecture in Practice*, Addison-Wesley.
3. Martin Fowler, *Patterns of Enterprise Application Architecture*, Addison-Wesley.
4. Roger S. Pressman, *Software Engineering: A Practitioner's Approach*, McGraw-Hill.
5. Kent Beck, *Test-Driven Development by Example*, Addison-Wesley.

Course Outcomes (COs)

CO	Description
CO1	Apply fundamental design principles to develop efficient and maintainable software systems.
CO2	Integrate effective requirements engineering practices in software development processes.
CO3	Design scalable and secure architectures tailored to modern applications.
CO4	Implement software quality measures and estimations for robust development.
CO5	Adapt to and utilize emerging tools and technologies in software system design.

Course Description

The course "Advanced Data Structures" focuses on the study and implementation of complex data structures that are essential for solving real-world computing problems. Students will learn to analyze, design, and implement advanced tree, graph, and hashing structures, with a focus on efficient algorithms and memory management for system-level programming.

Course Objectives

1. To provide an in-depth understanding of advanced data structures such as trees, graphs, and hashing techniques.
2. To enable students to design efficient algorithms for real-world problems involving large datasets.
3. To apply advanced data structures and algorithms for solving domain-specific challenges in computer science.
4. To improve students' problem-solving and programming skills, with a focus on memory optimization and system-level solutions.

Mapping Table

The mapping table uses a scale of 1 to 3 to indicate the level of correlation between COs and POs/PSOs, where:

1. Slight (Low)
2. Moderate (Medium)
3. Substantial (High)

CO-PO Mapping

Course Outcomes (CO)	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	-	3	-	-	-	-	-	-	-	1
CO2	3	1	3	3	-	-	-	-	-	-	-	-
CO3	3	2	-	3	-	-	-	-	-	-	-	2
CO4	2	1	3	2	2	-	-	-	-	-	-	1
CO5	3	2	-	3	3	-	-	-	-	-	-	3

CO-PSO Mapping

Course Outcomes (CO)	PSO1	PSO2	PSO3
CO1	3	-	-
CO2	2	3	1
CO3	-	-	-
CO4	-	3	3
CO5	3	2	3

1st Year - I Semester: M. Tech. (Computer Science)

Sub: Critical System Design

Credit: 5 **Max. Marks**: 125(IA:25, ETE:100)

5L+0T+0P **End Term Exam**: 3 Hours

S.No.	CONTENTS	CONTACT HOURS
1	**Introduction**: Objective, scope, and outcome of the course.	01
2	**Fundamentals of Critical Systems**: Definitions, Types of critical systems (safety-critical, mission-critical), Fault tolerance, Reliability engineering, Availability analysis, Redundancy management, Real-time requirements, Lifecycle of critical systems.	08
3	**Risk Management in Critical Systems**: Hazard analysis, Fault tree analysis (FTA), Failure mode and effects analysis (FMEA), Event tree analysis, Risk assessment models, Cost-benefit analysis, Risk control strategies, Case studies (aviation, healthcare, etc.).	08
4	**Design for Fault Tolerance**: Fault detection, Error correction codes, Checkpointing and recovery, N-version programming, Watchdog timers, Byzantine fault tolerance, Redundancy techniques (hardware, software, information).	08
5	**Real-Time System Design**: Scheduling algorithms (Rate-monotonic, EDF), Resource management in real-time systems, Task prioritization, Deadlock avoidance, Time-critical protocols, Synchronization in real-time systems.	08
6	**Security in Critical Systems**: Cybersecurity fundamentals, Secure communication protocols, Vulnerability assessment, Secure coding practices, Penetration testing, Role of cryptography in critical systems, Case studies on security breaches.	08

Text/Reference Books

1. Nancy Leveson, *Engineering a Safer World*, MIT Press.
2. Charles Perrow, *Normal Accidents: Living with High-Risk Technologies*, Princeton University Press.
3. James L. H. Anderson, *Real-Time Systems Design*, Wiley.
4. Kim Fowler, *Mission-Critical and Safety-Critical Systems Handbook*, Elsevier.
5. Ravi Sandhu, *Secure Systems Design*, Addison-Wesley.

Course Outcomes (COs)

CO	Description
CO1	Understand the principles of designing and maintaining critical systems.
CO2	Apply risk management techniques to enhance system reliability.
CO3	Design fault-tolerant systems for robust and error-resilient performance.
CO4	Develop secure critical systems to mitigate cybersecurity threats.
CO5	Explore modern trends and technologies in critical system design.

Course Description:

The "Critical System Design" course focuses on the principles, techniques, and tools required to design, develop, and maintain systems that are essential for safety, security, and performance in various engineering applications. It emphasizes the creation of fault-tolerant, secure, and reliable systems, with a focus on mitigating risks and handling potential threats in critical environments.

Course Objectives:

1. To understand the principles involved in designing and maintaining critical systems in diverse industries.
2. To apply risk management techniques to enhance the reliability and safety of systems.
3. To design fault-tolerant systems that ensure robust and error-resilient performance.
4. To develop secure critical systems by addressing cybersecurity threats and vulnerabilities.
5. To explore current trends and technologies in the design and implementation of critical systems.

Mapping Table

The mapping table uses a scale of 1 to 3 to indicate the level of correlation between COs and POs/PSOs, where:

1. Slight (Low)
2. Moderate (Medium)
3. Substantial (High)

CO-PO Mapping

Course Outcomes (CO)	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	2	2	-	-	-	-	-	-	-	2
CO2	3	3	3	3	3	-	-	-	-	-	-	-
CO3	3	3	3	3	3	-	-	-	-	-	-	2
CO4	3	3	3	3	3	2	2	-	-	-	-	2
CO5	3	3	3	3	3	2	2	-	-	2	2	3

CO-PSO Mapping

Course Outcomes (CO)	PSO1	PSO2	PSO3
CO1	3	-	-
CO2	3	3	-
CO3	3	3	-
CO4	3	3	3
CO5	3	3	3

UNIVERSITY OF TECHNOLOGY, JAIPUR

1st Year - I Semester: M. Tech. (Computer Science)

Sub: Cloud Computing and Big Data Systems

Credit: 5 **Max. Marks**: 125 (IA:25, ETE:100)

5L+0T+0P **End Term Exam**: 3 Hours

S.No.	CONTENTS	CONTACT HOURS
1	**Introduction**: Objective, scope, and outcome of the course.	01
2	**Foundations of Cloud Computing**: Virtualization, Cloud service models (IaaS, PaaS, SaaS), Deployment models (public, private, hybrid), Cloud computing architecture, Cloud storage and resource pooling, Cloud security fundamentals, Case studies of AWS, Azure, GCP.	08
3	**Distributed Systems for Cloud**: Distributed file systems (HDFS, Ceph), Consensus algorithms (Paxos, Raft), Distributed databases (Cassandra, DynamoDB), Event-driven architecture, CAP theorem, Message queuing systems (Kafka, RabbitMQ).	08
4	**Big Data Systems and Frameworks**: Hadoop ecosystem, Spark architecture, Dataflow programming, Stream processing frameworks (Apache Flink, Storm), NoSQL databases (MongoDB, Redis), ETL pipelines, Graph databases (Neo4j).	08
5	**Cloud Security and Privacy**: Identity and access management (IAM), Encryption in the cloud, Data residency and compliance (GDPR, HIPAA), Secure APIs, Cloud incident response, Cloud compliance frameworks (ISO 27001, SOC 2).	08
6	**Advanced Topics in Cloud and Big Data**: Edge and fog computing, Serverless architecture, AI on the cloud, Cloud-native app development, Kubernetes for container orchestration, Quantum computing in the cloud, Cost optimization strategies.	08

Text/Reference Books

1. Rajkumar Buyya, *Cloud Computing: Principles and Paradigms*, Wiley.
2. Tom White, *Hadoop: The Definitive Guide*, O'Reilly Media.
3. Arun C. Murthy, *Apache Hadoop YARN*, Addison-Wesley.
4. Tim Mather, *Cloud Security and Privacy*, O'Reilly Media.
5. Zaharia Matei, *Learning Spark*, O'Reilly Media.

Course Outcomes (COs)

CO	Description
CO1	Understand the principles and architecture of cloud computing systems.
CO2	Apply distributed systems concepts to design scalable cloud solutions.
CO3	Implement big data processing frameworks for large-scale data analytics.
CO4	Develop secure cloud applications following industry standards.
CO5	Explore emerging trends in cloud and big data technologies.

Course Description:

The "Cloud Computing and Big Data Systems" course introduces students to the concepts, architecture, and technologies behind cloud computing and big data systems. It covers the design of scalable cloud solutions, the use of distributed systems for data processing, and the development of secure cloud applications. The course also explores current trends and future advancements in cloud and big data technologies.

Course Objectives:

1. To understand the principles and architecture of cloud computing systems.
2. To apply distributed systems concepts in designing scalable and efficient cloud solutions.
3. To implement big data processing frameworks for large-scale data analytics.
4. To develop secure cloud applications following best practices and industry standards.
5. To explore emerging trends and technologies in cloud computing and big data.

Mapping Table

The mapping table uses a scale of 1 to 3 to indicate the level of correlation between COs and POs/PSOs, where:

1. Slight (Low)
2. Moderate (Medium)
3. Substantial (High)

CO-PO Mapping

Course Outcomes (CO)	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	2	-	3	-	-	-	-	-	-	-
CO2	2	3	-	3	3	-	-	-	-	-	-	3
CO3	1	3	3	3	2	-	-	-	-	-	-	2
CO4	1	3	2	3	-	-	-	-	2	-	-	1
CO5	3	3	3	3	3	-	2	-	-	-	-	2

CO-PSO Mapping

Course Outcomes (CO)	PSO1	PSO2	PSO3
CO1	2	-	-
CO2	3	3	-
CO3	-	-	1
CO4	2	3	3
CO5	3	-	3

1st Year - I Semester: M. Tech. (Computer Science)

Lab: Software System Lab

Credit: 2 **Max. Marks**: 100(IA:60, ETE:40)

0L+0T+4P

S. No.	Practical
1	Familiarize students with IDEs and version control systems like Git.
2	Implement file operations (read, write, and modify) and explore serialization and deserialization in programming languages like Java or Python.
3	Develop programs demonstrating thread creation, inter-thread communication, and synchronization.
4	Establish database connections and execute CRUD operations using JDBC/ODBC or ORM frameworks.
5	Create and deploy REST APIs and test them using tools like Postman.
6	Apply creational, structural, and behavioural design patterns in developing a mini software system.
7	Write unit tests and perform integration testing using tools like JUnit or PyTest.
8	Perform branch creation, merging, conflict resolution, and collaborative development using Git commands and GitHub workflows.
9	Analyze the performance of a given software module and optimize its resource usage and efficiency.
10	Deploy a software application on cloud platforms like AWS, Azure, or Google Cloud and explore containerization using Docker.

UNIVERSITY OF TECHNOLOGY, JAIPUR
Scheme and Syllabus
M.Tech 1st Year, II Sem: Computer Science

Teaching and Examination Scheme
ACADEMIC SESSION 2023-2024

S. No.	Course Name	Contact Hours per Week			Marks				Cr
		L	T	P	Exam Hrs	IA	ETE	Total	
1	Advanced Database Management Systems	5	0	0	3	25	100	125	5
2	Design of Embedded Systems	5	0	0	3	25	100	125	5
3	Distributed Algorithms	5	0	0	3	25	100	125	5
4	Advanced Computer Graphics	5	0	0	3	25	100	125	5
5	Advanced Database Lab	0	0	4		60	40	100	2
	Total	**20**	**0**	**4**		**160**	**440**	**600**	**22**

L: *Lecture,* ***T:*** *Tutorial,* ***P:*** *Practical,* ***Cr:*** *Credits*
ETE: *End Term Exam,* ***IA:*** *Internal Assessment*
**: Not to be considered for award of division*

UNIVERSITY OF TECHNOLOGY, JAIPUR

1st Year - II Semester: M. Tech. (Computer Science)

Sub: Advanced Database Management Systems (ADBMS)

Credit: 5 **Max. Marks**: 125(IA:25, ETE:100)

5L+0T+0P **End Term Exam**: 3 Hours

S.No.	CONTENTS	CONTACT HOURS
1	**Introduction**: Objective, scope, and outcome of the course.	01
2	**Advanced SQL and Query Optimization**: Complex queries, Recursive queries, Window functions, Common Table Expressions (CTEs), Materialized views, Query execution plans, Cost-based optimization, Indexing strategies, Use of NoSQL with SQL, Parallel query execution, Query tuning.	08
3	**Distributed Databases**: Distributed database design, Fragmentation, Replication, Distributed query processing, Two-phase and three-phase commit protocols, CAP theorem, BASE properties, Distributed database recovery techniques, Case studies (Spanner, Amazon Aurora).	08
4	**Database Security and Privacy**: Role-based access control (RBAC), Attribute-based access control (ABAC), SQL injection and prevention, Data masking, Encryption techniques (column-level, tablespace), Secure storage in databases, GDPR and HIPAA compliance in database systems, Blockchain-based data storage.	08
5	**Big Data and NoSQL Databases**: Hadoop Distributed File System (HDFS), MapReduce programming model, NoSQL databases (MongoDB, Cassandra, Neo4j), Schema design in NoSQL, Consistency models (eventual, strong), ACID vs BASE, Column-store databases, Graph databases, Data sharding.	08
6	**Emerging Trends in Databases**: Temporal databases, Spatial databases, JSON and XML databases, Real-time analytics databases, Cloud-native databases (Firebase, Snowflake), Time-series databases, Database-as-a-service (DBaaS).	08

Text/Reference Books

1. Elmasri & Navathe, *Fundamentals of Database Systems*, Pearson.
2. Raghu Ramakrishnan, *Database Management Systems*, McGraw-Hill.
3. C.J. Date, *An Introduction to Database Systems*, Pearson.
4. Han et al., *Data Mining: Concepts and Techniques*, Elsevier.
5. Guy Harrison, *Next Generation Databases*, Apress.

Course Outcomes (COs)

CO	Description
CO1	Formulate and execute advanced SQL queries with optimization techniques.
CO2	Design and manage distributed database systems with fault tolerance and consistency.
CO3	Implement security mechanisms to protect database integrity and privacy.
CO4	Explore NoSQL and big data technologies for modern data storage challenges.
CO5	Apply emerging trends to solve complex data management problems.

Course Description:

The course delves into the principles and techniques for designing and managing complex database systems. It covers advanced topics such as optimization of SQL queries, distributed database systems, security mechanisms, and the use of NoSQL and big data technologies to address modern data storage and management challenges. The course also explores emerging trends in database technologies.

Course Objectives:

1. To formulate and execute advanced SQL queries with optimization techniques to improve performance.
2. To design and manage distributed database systems ensuring fault tolerance and consistency.
3. To implement security mechanisms that protect database integrity and privacy.
4. To explore NoSQL and big data technologies to address modern data storage challenges.
5. To apply emerging trends and technologies to solve complex data management problems.

Mapping Table

The mapping table uses a scale of 1 to 3 to indicate the level of correlation between COs and POs/PSOs, where:

1. Slight (Low)
2. Moderate (Medium)
3. Substantial (High)

CO-PO Mapping

Course Outcomes (CO)	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	-	3	3	-	-	-	-	-	-	-	2
CO2	3	-	-	3	-	-	-	1	-	-	-	3
CO3	2	1	3	3	3	-	-	-	-	-	-	2
CO4	3	2	3	2	2	-	-	-	2	2	-	-
CO5	2	3	2	3	1	2	1	-	-	-	-	1

CO-PSO Mapping

Course Outcomes (CO)	PSO1	PSO2	PSO3
CO1	3	2	-
CO2	3	2	-
CO3	1	3	2
CO4	2	3	3
CO5	3	3	3

UNIVERSITY OF TECHNOLOGY, JAIPUR

1st Year - II Semester: M. Tech. (Computer Science)

Sub: Design of Embedded Systems

Credit: 5 **Max. Marks**: 125 (IA:25, ETE:100)

5L+0T+0P **End Term Exam**: 3 Hours

S.No.	CONTENTS	CONTACT HOURS
1	**Introduction**: Objective, scope, and outcome of the course.	01
2	**Embedded System Basics**: Characteristics, Design challenges, Real-time systems, Embedded hardware components, Microcontroller architectures (ARM, AVR, RISC-V), Development environments, Bootloader design, Embedded OS concepts.	08
3	**Embedded System Design Techniques**: Hardware-software co-design, Partitioning methods, State machine modeling, Task scheduling, Design verification, Resource optimization, Integration of peripherals, Low-power design techniques.	08
4	**Real-Time Operating Systems (RTOS)**: Concepts and architecture, Scheduling policies, Inter-process communication (IPC), Memory management in RTOS, Examples of RTOS (FreeRTOS, VxWorks, QNX), RTOS for IoT.	08
5	**Embedded Communication Protocols**: UART, I2C, SPI, CAN, LIN, USB, Ethernet, Wireless protocols, Middleware protocols (MQTT, CoAP), Protocol stack implementation.	08
6	**Applications of Embedded Systems**: Automotive systems, Healthcare devices, Consumer electronics, Smart grid systems, Robotics, IoT systems, Edge AI systems, Security and encryption in embedded applications.	08

Text/Reference Books

1. Marilyn Wolf, *Computers as Components*, Morgan Kaufmann.
2. Raj Kamal, *Embedded Systems: Architecture, Programming and Design*, McGraw-Hill.
3. Frank Vahid, *Embedded System Design*, Wiley.
4. Qing Li, *Real-Time Concepts for Embedded Systems*, CMP Books.
5. Jonathan W. Valvano, *Embedded Systems*, CreateSpace Independent Publishing.

Course Outcomes (COs)

CO	Description
CO1	Understand the principles and components of embedded systems.
CO2	Apply design techniques to develop optimized embedded solutions.
CO3	Implement real-time operating systems for multitasking embedded applications.
CO4	Integrate communication protocols in embedded systems for diverse applications.
CO5	Explore domain-specific embedded solutions with advanced features.

Course Description

The course "Design of Embedded Systems" provides an in-depth understanding of embedded system architecture, design techniques, real-time operating systems, and communication protocols. Students will explore various methodologies to develop optimized embedded solutions for diverse applications, including advanced domain-specific solutions.

Course Objectives

- Understand the principles and components of embedded systems.
- Learn design techniques for developing efficient embedded solutions.
- Gain expertise in real-time operating systems and multitasking in embedded systems.
- Integrate communication protocols for diverse embedded system applications.
- Explore advanced features and domain-specific embedded solutions.

Mapping Table

The mapping table uses a scale of 1 to 3 to indicate the level of correlation between COs and POs/PSOs, where:

1. Slight (Low)
2. Moderate (Medium)
3. Substantial (High)

CO-PO Mapping

Course Outcomes (CO)	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	2	1	2	3	2	1	2	-	-	1	2
CO2	1	3	3	1	-	3	2	1	-	3	-	2
CO3	-	-	3	2	-	3	2	-	2	-	2	3
CO4	2	3	2	3	-	1	3	1	2	3	1	2
CO5	3	3	2	3	3	2	2	2	-	3	3	2

CO-PSO Mapping

Course Outcomes (CO)	PSO1	PSO2	PSO3
CO1	3	2	1
CO2	2	3	2
CO3	3	-	-
CO4	2	-	-
CO5	3	3	3

UNIVERSITY OF TECHNOLOGY, JAIPUR

1st Year - II Semester: M. Tech. (Computer Science)

Sub: Distributed Algorithms

Credit: 5 **Max. Marks**: 125(IA:25, ETE:100)

5L+0T+0P **End Term Exam**: 3 Hours

S.No.	CONTENTS	CONTACT HOURS
1	**Introduction**: Objective, scope, and outcome of the course.	01
2	**Fundamentals of Distributed Systems**: Models of distributed computing, Synchronous vs asynchronous systems, Challenges in distributed computing, Message passing and shared memory models, Lamport timestamps, Logical clocks.	08
3	**Consensus and Agreement Algorithms**: Paxos, Raft, Byzantine fault tolerance, Leader election, Termination detection, Mutual exclusion algorithms (Ricart-Agrawala, Token-based), Distributed snapshots.	08
4	**Distributed Data Management**: Consistency models (eventual, strong, causal), Distributed hash tables (Chord, CAN), Distributed caching, Replication techniques, Consistency and availability trade-offs (CAP theorem).	08
5	**Fault Tolerance and Recovery**: Checkpointing, Rollback recovery, Reliable broadcast, Replicated state machines, Fault-tolerant protocols, Deadlock detection in distributed systems.	08
6	**Applications and Advanced Topics**: Blockchain consensus algorithms, Distributed AI, Distributed file systems, Cloud resource scheduling, Gossip protocols, Edge and fog computing, Real-time distributed systems.	08

Text/Reference Books

1. Nancy Lynch, *Distributed Algorithms*, Morgan Kaufmann.
2. Sukumar Ghosh, *Distributed Systems: An Algorithmic Approach*, CRC Press.
3. Tanenbaum & Van Steen, *Distributed Systems*, Pearson.
4. Vijay K. Garg, *Elements of Distributed Computing*, Wiley.
5. Coulouris et al., *Distributed Systems: Concepts and Design*, Pearson.

Course Outcomes (COs)

CO	Description
CO1	Analyze the principles and challenges of distributed algorithms.
CO2	Implement consensus and agreement protocols in distributed systems.
CO3	Develop efficient and consistent distributed data management systems.
CO4	Apply fault tolerance techniques to enhance system reliability.
CO5	Explore emerging applications and advanced concepts in distributed algorithms.

Course Description

The course on Distributed Algorithms introduces the fundamental principles and techniques used in distributed systems. It covers key topics such as consensus protocols, fault tolerance, distributed data management, and their applications in real-world systems. The course aims to equip students with the knowledge to analyze and design efficient algorithms for distributed environments.

Course Objectives

1. To understand the theoretical foundations of distributed algorithms and their practical challenges.
2. To implement consensus and agreement protocols in distributed systems.
3. To develop efficient and consistent methods for distributed data management.
4. To apply fault tolerance techniques to ensure the reliability and robustness of distributed systems.

Mapping Table

The mapping table uses a scale of 1 to 3 to indicate the level of correlation between COs and POs/PSOs, where:

1. Slight (Low)
2. Moderate (Medium)
3. Substantial (High)

CO-PO Mapping

Course Outcomes (CO)	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	-	-	-	-	-	-	-	-	-	2
CO2	3	3	3	-	2	-	-	-	-	-	-	3
CO3	3	3	3	3	3	-	-	-	-	-	2	3
CO4	2	3	3	3	3	2	-	-	-	-	-	3
CO5	3	3	3	3	3	2	2	-	-	-	-	3

CO-PSO Mapping

Course Outcomes (CO)	PSO1	PSO2	PSO3
CO1	3	-	-
CO2	3	2	-
CO3	3	3	3
CO4	3	3	3
CO5	3	3	3

UNIVERSITY OF TECHNOLOGY, JAIPUR

1st Year - II Semester: M. Tech. (Computer Science)

Sub: Advanced Computer Graphics

Credit: 5 **Max. Marks**: 125(IA:25, ETE:100)

5L+0T+0P **End Term Exam**: 3 Hours

S.No.	CONTENTS	CONTACT HOURS
1	**Introduction**: Objective, scope, and outcome of the course.	01
2	**3D Graphics Fundamentals**: Coordinate transformations, 3D rendering pipeline, Perspective and orthographic projections, Clipping algorithms, Hidden surface removal (Z-buffer, Painter's algorithm).	08
3	**Advanced Rendering Techniques**: Ray tracing, Radiosity, Global illumination, Photon mapping, Monte Carlo rendering, Shader programming (GLSL, HLSL), Physically-based rendering (PBR).	08
4	**Modeling and Animation**: Procedural modeling, Spline-based modeling, Rigging and skinning, Keyframe animation, Motion capture, Particle systems, Fluid simulations.	08
5	**Interactive Graphics**: Real-time rendering, Game engine architecture, Virtual reality (VR) systems, Augmented reality (AR) frameworks, Gesture recognition, Haptic interfaces.	08
6	**Emerging Topics in Computer Graphics**: AI-based rendering, Neural radiance fields (NeRF), Real-time ray tracing (RTX), Cloud-based rendering, Quantum rendering, Metaverse graphics.	08

Text/Reference Books

1. Foley et al., *Computer Graphics: Principles and Practice*, Addison-Wesley.
2. Alan Watt, *3D Computer Graphics*, Pearson.
3. Tomas Akenine-Moller, *Real-Time Rendering*, A K Peters.
4. Rick Parent, *Computer Animation: Algorithms and Techniques*, Elsevier.
5. Steve Marschner, *Fundamentals of Computer Graphics*, CRC Press.

Course Outcomes (COs)

CO	Description
CO1	Understand 3D graphics and rendering techniques.
CO2	Apply advanced rendering algorithms for realistic visual effects.
CO3	Develop interactive graphics applications with modern frameworks.
CO4	Implement animation and modeling techniques in real-world scenarios.
CO5	Explore cutting-edge trends in computer graphics.

Course Description

The course on Advanced Computer Graphics focuses on the theory and application of advanced techniques in computer graphics. It covers topics like 3D graphics, rendering algorithms, animation, and interactive graphics development. The course also explores the latest trends and technologies in computer graphics to enable students to create realistic and dynamic visual experiences.

Course Objectives

1. To understand and apply 3D graphics and advanced rendering techniques for visual effects.
2. To implement modern rendering algorithms for realism in graphics.
3. To develop interactive graphics applications using advanced frameworks.
4. To learn and apply animation and modeling techniques in real-world graphics scenarios.
5. To explore cutting-edge trends and innovations in computer graphics, including virtual and augmented reality applications.

Mapping Table

The mapping table uses a scale of 1 to 3 to indicate the level of correlation between COs and POs/PSOs, where:

1. Slight (Low)
2. Moderate (Medium)
3. Substantial (High)

CO-PO Mapping

Course Outcomes (CO)	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	-	-	-	-	-	-	-	-	-	2
CO2	3	3	2	3	-	-	-	-	-	-	-	-
CO3	3	2	-	2	3	-	-	-	-	-	-	2
CO4	3	3	3	3	3	-	-	-	-	-	-	3
CO5	3	2	3	3	3	-	-	-	-	-	-	3

CO-PSO Mapping

Course Outcomes (CO)	PSO1	PSO2	PSO3
CO1	3	-	-
CO2	3	3	-
CO3	2	3	-
CO4	3	3	3
CO5	3	3	3

UNIVERSITY OF TECHNOLOGY, JAIPUR

1st Year - II Semester: M. Tech. (Computer Science)

Lab: Advanced Database Lab

Credit: 2 **Max. Marks**: 100(IA:60, ETE:40)

0L+0T+4P

S. No.	Practical
1	Install and configure database systems like PostgreSQL, MongoDB, or Oracle, and understand their architecture.
2	Create optimized database schemas using normalization techniques and enforce constraints.
3	Write complex SQL queries using JOINs, subqueries, window functions, and CTEs for advanced data analysis.
4	Create and execute stored procedures, user-defined functions, and triggers for database automation.
5	Implement indexing strategies and analyze query execution plans for performance tuning.
6	Explore NoSQL databases like MongoDB or Cassandra, perform CRUD operations, and query data using their specific query languages.
7	Implement data replication and partitioning techniques to ensure scalability and availability in distributed database systems.
8	Explore transaction management concepts like ACID properties and concurrency control mechanisms such as locking and timestamp ordering.
9	Design and implement a data warehouse schema and perform OLAP operations like roll-up, drill-down, and slice-and-dice using SQL-based tools.
10	Connect a database with applications developed in Java, Python, or .NET and perform operations programmatically.

UNIVERSITY OF TECHNOLOGY, JAIPUR
Scheme and Syllabus
M.Tech 2nd Year, III Sem: Computer Science

Teaching and Examination Scheme
ACADEMIC SESSION 2023-2024

S. No.	Course Name	Contact Hours per Week			Marks				Cr
		L	T	P	Exam Hrs	IA	ETE	Total	
1	Parallel and Distributed Computing	5	0	0	3	25	100	125	5
2	Network System Design	5	0	0	3	25	100	125	5
3	Seminar	0	0	0		90	60	150	6
4	Dissertation- I	0	0	0		100	0	100	4
	Total	**10**	**0**	**0**		**240**	**260**	**500**	**20**

L: *Lecture,* ***T:*** *Tutorial,* ***P:*** *Practical,* ***Cr:*** *Credits*
ETE: *End Term Exam,* ***IA:*** *Internal Assessment*
**: Not to be considered for award of division*

UNIVERSITY OF TECHNOLOGY, JAIPUR

2^{nd} Year - III Semester: M. Tech. (Computer Science)

Sub: Parallel and Distributed Computing

Credit: 5 **Max. Marks**: 125(IA:25, ETE:100)

5L+0T+0P **End Term Exam**: 3 Hours

S.No.	CONTENTS	CONTACT HOURS
1	**Introduction**: Overview of Parallel and Distributed Computing, Course objectives, and learning outcomes.	01
2	**Foundations of Parallel Computing**: Parallelism types (data, task, instruction-level), Flynn's taxonomy, Parallel architectures (shared memory, distributed memory, hybrid), Parallel algorithms, Parallel performance metrics (speedup, efficiency, scalability), Synchronization mechanisms (locks, barriers).	08
3	**Parallel Programming Models and Tools**: OpenMP, MPI, CUDA for GPU programming, TensorFlow for parallel machine learning, Parallel debugging and profiling tools, Shared vs. distributed memory programming, High-performance libraries (BLAS, MKL).	08
4	**Distributed Systems Concepts**: Characteristics of distributed systems, Middleware and messaging, Distributed algorithms (leader election, consensus, clock synchronization), Models of consistency (strong, eventual), Distributed file systems (HDFS, Ceph), Case studies of distributed systems (Google File System, Amazon Dynamo).	08
5	**Fault Tolerance and Reliability in Distributed Systems**: Fault models, Checkpointing and recovery, Replication techniques, Byzantine fault tolerance, Reliable multicast, Consensus algorithms (Paxos, Raft), Case studies of fault-tolerant systems.	08
6	**Distributed Databases and Cloud Integration**: Distributed database design, CAP theorem, Query optimization in distributed databases, Data replication and partitioning, Cloud platforms (AWS, Azure, GCP) for distributed computing, NoSQL databases for distributed storage.	08

Text/Reference Books

1. Andrew S. Tanenbaum, *Distributed Systems: Principles and Paradigms*, Pearson.
2. Ananth Grama et al., *Introduction to Parallel Computing*, Addison-Wesley.
3. Barry Wilkinson, *Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers*, Pearson.
4. Michael J. Quinn, *Parallel Programming in C with MPI and OpenMP*, McGraw-Hill.
5. Kai Hwang, *Distributed and Cloud Computing: From Parallel Processing to the Internet of Things*, Morgan Kaufmann.

Course Outcomes (COs)

CO	Description
CO1	Understand parallel computing models and evaluate performance metrics.
CO2	Design parallel programs using modern frameworks and tools.
CO3	Analyze distributed systems' architectures and their applications.
CO4	Develop fault-tolerant and scalable distributed systems.
CO5	Explore emerging technologies and their role in distributed computing.

Course Description

The course on Parallel and Distributed Computing explores the principles, techniques, and architectures of parallel and distributed systems. Students will learn how to design and implement efficient parallel programs, evaluate system performance, and explore fault tolerance in distributed systems. The course also covers emerging technologies shaping the future of parallel and distributed computing.

Course Objectives

1. To understand the fundamentals of parallel computing models and evaluate their performance metrics.
2. To design and develop parallel programs using modern frameworks and tools.
3. To analyse the architectures of distributed systems and their real-world applications.
4. To develop fault-tolerant and scalable distributed systems.
5. To explore emerging technologies and their impact on the future of distributed computing.

Mapping Table

The mapping table uses a scale of 1 to 3 to indicate the level of correlation between COs and POs/PSOs, where:

1. Slight (Low)
2. Moderate (Medium)
3. Substantial (High)

CO-PO Mapping

Course Outcomes (CO)	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	-	2	-	-	2	-	-	-	-	2
CO2	-	-	2	3	2	-	-	-	-	2	-	-
CO3	3	-	-	3	1	-	-	-	1	-	-	1
CO4	-	3	3	3	1	3	2	-	1	-	-	3
CO5	-	2	-	-	-	3	2	3	-	-	-	3

CO-PSO Mapping

Course Outcomes (CO)	PSO1	PSO2	PSO3
CO1	3	1	-
CO2	3	3	-
CO3	-	2	1
CO4	-	3	2
CO5	-	2	3

UNIVERSITY OF TECHNOLOGY, JAIPUR

2nd Year - III Semester: M. Tech. (Computer Science)

Sub: Network System Design

Credit: 5 **Max. Marks**: 125(IA:25, ETE:100)

5L+0T+0P **End Term Exam**: 3 Hours

S.No.	CONTENTS	CONTACT HOURS
1	**Introduction**: Scope, objectives, and learning outcomes of Network System Design.	01
2	**Fundamentals of Network Design**: Network design process, Requirements analysis, Network topology selection, LAN/WAN technologies, Protocol stack overview (OSI, TCP/IP), Network performance metrics (throughput, latency, jitter).	08
3	**Advanced Routing and Switching**: Interior and exterior routing protocols (OSPF, BGP, EIGRP), Multicast routing, VLANs and inter-VLAN routing, Software-defined networking (SDN), OpenFlow protocol, Traffic engineering.	08
4	**Network Security Design**: Security requirements, Threat modeling, Firewall design, Intrusion detection and prevention systems, VPNs and tunneling protocols, Zero Trust Architecture, Secure network design case studies.	08
5	**Wireless and Mobile Networks**: Cellular network architecture (4G, 5G), Wireless standards (802.11 family), Wireless sensor networks, MANETs, Network design for IoT, Challenges in mobile network design, Case studies on 5G network design.	08
6	**Network Virtualization and Cloud Integration**: Network Function Virtualization (NFV), Virtual network overlays (VXLAN, GRE), Cloud-based network design, Edge networking for cloud, Hybrid cloud networking solutions, Integration of SDN and NFV in cloud networks. Blockchain for secure networks, Quantum networks, High-speed networks (100GbE and beyond), Resilient networks for disaster recovery.	08

Text/Reference Books

1. James F. Kurose and Keith W. Ross, *Computer Networking: A Top-Down Approach*, Pearson.
2. Andrew S. Tanenbaum, *Computer Networks*, Pearson.

3. William Stallings, *Data and Computer Communications*, Pearson.
4. Paul Goransson, *Software Defined Networks: A Comprehensive Approach*, Morgan Kaufmann.
5. Behrouz A. Forouzan, *Data Communications and Networking*, McGraw-Hill.

Course Outcomes (COs)

CO	Description
CO1	Design efficient network architectures for diverse applications.
CO2	Analyze and implement advanced routing and switching technologies.
CO3	Develop secure network solutions addressing modern threats.
CO4	Integrate wireless and mobile technologies into network designs.
CO5	Utilize emerging trends and technologies for scalable and resilient networks.

Course Description

The course on Network System Design focuses on designing efficient and scalable network infrastructures. Students will learn advanced routing and switching technologies, secure network solutions to combat modern cybersecurity threats, and integration of wireless and mobile technologies into network systems. Additionally, the course explores emerging trends and technologies to build resilient, high-performance networks.

Course Objectives

1. To design efficient network architectures for a variety of applications.
2. To analyze and implement advanced routing and switching technologies.
3. To develop secure network solutions that address modern security threats.
4. To integrate wireless and mobile technologies into network designs.
5. To utilize emerging trends and technologies to create scalable and resilient networks.

Mapping Table

The mapping table uses a scale of 1 to 3 to indicate the level of correlation between COs and POs/PSOs, where:

1. Slight (Low)
2. Moderate (Medium)
3. Substantial (High)

CO-PO Mapping

Course Outcomes (CO)	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	2	1	1	-	-	-	-	2	-	-	-
CO2	2	3	2	2	-	-	-	-	-	-	-	-
CO3	2	3	2	3	2	-	-	-	1	-	-	-
CO4	-	2	-	3	-	-	-	-	-	-	-	-
CO5	2	2	2	2	3	2	2	-	-	-	-	1

CO-PSO Mapping

Course Outcomes (CO)	PSO1	PSO2	PSO3
CO1	3	2	-
CO2	-	3	2
CO3	2	3	-
CO4	3	2	3
CO5	3	3	-