GLOBAL INSTITUTE OF ENGINEERING AND TECHNOLOGY (AUTONOMOUS)
COURSE CATALOGUE
REGULATIONS B.TECH – GR - 24
COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE)
II YEAR II SEMESTER

Course Code	Course Name	Subject Area	Category	Periods Per Week			Credits	Scheme of Examination Max Marks		
				L	T	P		CIA	SEE	Total
THEORY										
DS401PC	Discrete Mathematics	PCC	CORE	3	0	0	3	40	60	100
DS402PC	Computer Organization and Architecture	PCC	CORE	3	0	0	3	40	60	100
DS403PC	Operating Systems	PCC	CORE	3	0	0	3	40	60	100
DS404PC	Computer Networks	PCC	CORE	3	0	0	3	40	60	100
DS405PC	Software Engineering	PCC	CORE	3	0	0	3	40	60	100
PRACTICAL										
DS406PC	Operating Systems Lab	PCC	CORE	0	0	2	1	40	60	100
DS407PC	Computer Networks Lab	PCC	CORE	0	0	2	1	40	60	100
DS408PC	Real-Time/Research Project/Societal Related Project	PROJ	PROJECT	0	0	4	2	40	60	100
DS409PC	Data Visualization R-Programming/Power BI	PCC	CORE	0	0	2	1	40	60	100
MANDATORY COURSE										
*MC410	Intellectual Property Rights	MC - IV	MC	3	0	0	0			
			Total Credits	**18**	**0**	**10**	**20**			

COURSE CONTENT

DISCRETE MATHEMATICS								
II Year - II Semester: CSE (DS)								
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
DS401PC	**Core**	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		3	**-**	**-**	**3**	**40**	**60**	**100**
Contact Classes: 48	Tutorial Classes: Nil	Practical Classes: Nil				**Total Classes: 48**		
Prerequisite: Mathematics courses of the first year of study.								

1. COURSE OVERVIEW

This course introduces students to the foundational concepts of **Discrete Mathematics**, which form the backbone of computer science and engineering. It equips students with essential tools to reason mathematically, model computational problems, and design efficient algorithms.

The course begins with **Mathematical Logic**, where students learn formal notation, propositional and predicate calculus, and methods of inference to construct precise proofs. It then covers **Set Theory, Relations, and Functions**, which are widely applied in database theory, automata, and programming languages.

Students also explore **Algebraic Structures** such as semigroups, monoids, lattices, and Boolean algebra, which play a vital role in digital logic design, cryptography, and theoretical computer science. The course further develops problem-solving skills through **Combinatorics and Counting Principles**, including permutations, combinations, binomial and multinomial theorems, and the principle of inclusion-exclusion—key tools for algorithm analysis and complexity.

Finally, students are introduced to **Graph Theory**, covering trees, spanning trees, binary trees, planar graphs, Euler and Hamiltonian circuits, chromatic numbers, and the Four-Colour problem, with applications in networking, data structures, and optimisation problems.

2. COURSE OBJECTIVE

The students will try to Learn:

1) Introduces elementary discrete mathematics for computer science and engineering.
2) Topics include formal logic notation, methods of proof, induction, sets, relations, algebraic structures, elementary graph theory, permutations and combinations, counting principles, recurrence relations and generating functions.

3. COURSE OUTCOMES

After successful completion of the course, students should be able to:

CO 1	Understand and construct precise mathematical proofs
CO 2	Apply logic and set theory to formulate precise statements
CO 3	Analyse and solve counting problems on finite and discrete structures
CO 4	Describe and manipulate sequences
CO 5	Apply graph theory in solving computing problems.

4. COURSE CONTENT

UNIT - I Mathematical logic: 10L

Introduction, Statements and Notation, Connectives, Normal Forms, Theory of Inference for the Statement Calculus, The Predicate Calculus, Inference Theory of the Predicate Calculus.

UNIT - II Set theory: 8L

Introduction, Basic Concepts of Set Theory, Representation of Discrete Structures, Relations and Ordering, Functions.

UNIT - III Algebraic Structures: 10L

Introduction, Algebraic Systems, Semigroups and Monoids, Lattices as Partially Ordered Sets, Boolean Algebra.

UNIT - IV Elementary Combinatorics: 10 L

Basics of Counting, Combinations and Permutations, Enumeration of Combinations and Permutations, Enumerating Combinations and Permutations with Repetitions, Enumerating Permutations with Constrained Repetitions, Binomial Coefficient, The Binomial and Multinomial Theorems, The Principle of Exclusion.

UNIT - V Graph Theory: 10L

Basic Concepts, Isomorphism and Subgraphs, Trees and their Properties, Spanning Trees, Directed Trees, Binary Trees, Planar Graphs, Euler's Formula, Multi-graphs and Euler Circuits, Hamiltonian Graphs, Chromatic Numbers, The Four-Colour Problem.

5. TEXT BOOKS

1) Discrete Mathematical Structures with Applications to Computer Science: J.P. Tremblay, R. Manohar, McGraw-Hill, 1st Edition.
2) Discrete Mathematics for Computer Scientists & Mathematicians: Joe L. Mott, Abraham Kandel, Theodore P. Baker, Prentis Hall of India, 2nd Edition.

6. REFERENCE BOOKS

1) Discrete and Combinatorial Mathematics - an applied introduction: Ralph P. Grimald, Pearson Education, 5th edition.
2) Discrete Mathematical Structures: Thomas Kosy, Tata McGraw-Hill Publishing Co.

CO-PO-PSO Mapping

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2
CO 1	**3**	**2**										**2**	**3**	**2**
CO 2	**2**	**3**										**2**	**2**	**3**
CO 3	**2**	**3**											**2**	**3**
CO 4	**3**	**-**	**1**		**2**								**3**	**-**
CO 5	**3**	**3**			**1**							**2**	**3**	**3**

COURSE CONTENT

COMPUTER ORGANIZATION AND ARCHITECTURE								
II Year - II Semester: CSE (DS)								
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
DS402PC	**Core**	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		3	**-**	**-**	**3**	**40**	**60**	**100**
Contact Classes: 48	Tutorial Classes: Nil	Practical Classes: Nil				**Total Classes: 48**		
Prerequisite: Nil								

1. COURSE OVERVIEW

This course provides a comprehensive introduction to the fundamental principles of computer organization and architecture. It covers the internal structure and operational behavior of digital computers, focusing on the interaction between hardware and software at a low level. Topics include data representation, instruction set architecture (ISA), processor design, memory hierarchy, input/output systems, and performance evaluation. Students will gain an understanding of how computers execute programs, how hardware supports software, and how performance is impacted by architectural choices.

2. COURSE OBJECTIVE

The students will try to Learn:

1) The purpose of the course is to introduce principles of computer organization and the basic architectural concepts.
2) It begins with basic organization, design, and programming of a simple digital computer and introduces simple register transfer language to specify various computer operations.
3) Topics include computer arithmetic, instruction set design, microprogrammed control unit, pipelining and vector processing, memory organization and I/O systems, and multiprocessors.

3. COURSE OUTCOMES

After successful completion of the course, students should be able to:

CO 1	Demonstrate an understanding of the design of the functional units of a digital computer system.
CO 2	Design of Control unit and Central processing unit.
CO 3	Recognize and manipulate representations of numbers stored in digital computers
CO 4	Understand Input - Output Organization and Memory Organization.
CO 5	Design a pipeline for consistent execution of instructions with minimum hazards.

4. COURSE CONTENT

UNIT - I **(10L)**

Digital Computers: Introduction, Block diagram of Digital Computer, Definition of Computer Organization, Computer Design and Computer Architecture.

Register Transfer Language and Micro operations: Register Transfer language, Register Transfer, Bus and memory transfers, Arithmetic Micro operations, logic micro operations, shift micro operations, Arithmetic logic shift unit.

Basic Computer Organization and Design: Instruction codes, Computer Registers Computer instructions, Timing and Control, Instruction cycle, Memory Reference Instructions, Input – Output and Interrupt.

UNIT - II **(10L)**

Microprogrammed Control: Control memory, Address sequencing, micro program example, design of control unit.

Central Processing Unit: General Register Organization, Instruction Formats, Addressing modes, Data Transfer and Manipulation, Program Control.

UNIT - III **(10L)**

Data Representation: Data types, Complements, Fixed Point Representation, Floating Point Representation.

Computer Arithmetic: Addition and subtraction, multiplication Algorithms, Division Algorithms, Floating – point Arithmetic operations. Decimal Arithmetic unit, Decimal Arithmetic operations.

UNIT - IV **(9L)**

Input-Output Organization: Input-Output Interface, Asynchronous data transfer, Modes of Transfer, Priority Interrupt Direct memory Access.

Memory Organization: Memory Hierarchy, Main Memory, Auxiliary memory, Associate Memory, Cache Memory.

UNIT - V **(9L)**

Reduced Instruction Set Computer: CISC Characteristics, RISC Characteristics.

Pipeline and Vector Processing: Parallel Processing, Pipelining, Arithmetic Pipeline, Instruction Pipeline, RISC Pipeline, Vector Processing, Array Processor.

Multi Processors: Characteristics of Multiprocessors, Interconnection Structures, Interprocessor arbitration, Interprocessor communication and synchronization, Cache Coherence.

5. TEXT BOOKS

1) Computer System Architecture – M. Morris Mano, Third Edition, Pearson/PHI.

6. REFERENCE BOOKS

1) Computer Organization – Carl Hamacher, Zvonks Vranesic, SafeaZaky, Vth Edition, McGraw Hill.
2) Computer Organization and Architecture – William Stallings Sixth Edition, Pearson/PHI.
3) Structured Computer Organization – Andrew S. Tanenbaum, 4 th Edition, PHI/Pearson.

CO-PO-PSO Mapping

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2
CO 1	**3**	**3**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**		
CO 2	**3**	**3**	**2**	**2**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**		
CO 3	**3**	**3**	**3**	**2**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**		
CO 4	**3**	**3**	**3**	**2**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**		
CO 5	**3**	**3**	**0**	**0**	**3**	**0**	**0**	**0**	**0**	**0**	**0**	**2**		

COURSE CONTENT

OPERATING SYSTEMS								
II Year - II Semester: CSE (DS)								
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
DS403PC	**Core**	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		3	**-**	**-**	**3**	**40**	**60**	**100**
Contact Classes: 48	Tutorial Classes: Nil	Practical Classes: Nil				**Total Classes: 48**		
Prerequisite: 1. A course on Computer Programming and Data Structures. 2. A course on Computer Organization and Architecture.								

1. COURSE OVERVIEW

This course introduces the fundamental principles of **Operating Systems (OS)**, which act as an intermediary between hardware and users. It covers concepts such as **processes, CPU scheduling, synchronization, deadlocks, memory management, file systems, and system calls**, with emphasis on both theoretical foundations and practical aspects of Unix/Linux.

2. COURSE OBJECTIVE

The students will try to Learn:

1) To understand **operating system concepts** including processes, threads, scheduling, synchronization, deadlocks, memory management, and file systems.
2) To explore the **design and development issues** of modern operating systems.
3) To learn and practice **Unix commands** and system call interfaces for process, communication, and I/O management.

3. COURSE OUTCOMES

After successful completion of the course, students should be able to:

CO 1	Control access to a computer and files, ensuring secure sharing and protection.
CO 2	Demonstrate knowledge of computer components and their respective roles in computing.
CO 3	Recognize and resolve user problems within standard operating environments.
CO 4	Gain practical knowledge of how programming languages, operating systems, and computer architecture interact effectively.
CO 5	Apply expert system development tools and frameworks to design domain-specific intelligent applications.

4. COURSE CONTENT

UNIT - I **(10L)**

Operating System - Introduction, Structures - Simple Batch, Multiprogrammed, Time-shared, Personal Computer, Parallel, Distributed Systems, Real-Time Systems, System components, Operating System services, System Calls.

Process - Process concepts and scheduling, Operations on processes, Cooperating Processes, Threads.

UNIT - II **(10L)**

CPU Scheduling - Scheduling Criteria, Scheduling Algorithms, Multiple -Processor Scheduling. System call interface for process management-fork, exit, wait, waitpid, exec.

Deadlocks - System Model, Deadlocks Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, and Recovery from Deadlock

UNIT – III **(10L)**

Process Management and Synchronization - The Critical Section Problem, Synchronization Hardware, Semaphores, and Classical Problems of Synchronization, Critical Regions, Monitors Interprocess Communication Mechanisms: IPC between processes on a single computer system, IPC between processes on different systems, using pipes, FIFOs, message queues, shared memory.

UNIT – IV **(9L)**

Memory Management and Virtual Memory - Logical versus Physical Address Space, Swapping, Contiguous Allocation, Paging, Segmentation, Segmentation with Paging, Demand Paging, Page Replacement, Page Replacement Algorithms.

UNIT – V **(9L)**

File System Interface and Operations -Access methods, Directory Structure, Protection, File System Structure, Allocation methods, Free-space Management. Usage of open, create, read, write, close, lseek, stat, ioctl system calls.

5. TEXT BOOKS

1) Abraham Silberchatz, Peter B. Galvin, Greg Gagne, *Operating System Principles*, 7th Edition, John Wiley.
2) W.R. Stevens, *Advanced Programming in the UNIX Environment*, Pearson Education.

6. REFERENCE BOOKS

1) William Stallings, *Operating Systems: Internals and Design Principles*, 5th Edition, Pearson/PHI.
2) Crowley, *Operating System: A Design Approach*, TMH.
3) Andrew S. Tanenbaum, *Modern Operating Systems*, 2nd Edition, Pearson/PHI.
4) Kernighan and Pike, *UNIX Programming Environment*, PHI/Pearson Education.
5) U. Vahalia, *UNIX Internals: The New Frontiers*, Pearson Education.

CO-PO-PSO Mapping

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2
CO 1	**3**	**2**	**3**	**2**	**2**	**1**	**2**	**1**	**1**	**2**	**2**	**2**		
CO 2	**3**	**2**	**2**	**2**	**3**	**1**	**1**	**1**	**1**	**2**	**2**	**2**		
CO 3	**2**	**2**	**3**	**3**	**2**	**2**	**2**	**1**	**1**	**2**	**2**	**3**		
CO 4	**3**	**2**	**3**	**3**	**3**	**2**	**2**	**1**	**1**	**2**	**2**	**3**		
CO 5	**3**	**2**	**3**	**3**	**3**	**2**	**2**	**1**	**2**	**2**	**2**	**3**		

COURSE CONTENT

COMPUTER NETWORKS								
II Year - II Semester: CSE (DS)								
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
DS404PC	**Core**	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		3	**-**	**-**	**3**	**40**	**60**	**100**
Contact Classes: 48	Tutorial Classes: Nil	Practical Classes: Nil				**Total Classes: 48**		
Prerequisite: 1. A course on *Programming for Problem Solving* 2. A course on *Data Structures*								

1. COURSE OVERVIEW

The **Computer Networks** course introduces fundamental concepts of networking, layered architectures, and protocols used in modern communication systems. Students will gain an in-depth understanding of the OSI and TCP/IP reference models, network topologies, subnetting, routing mechanisms, and essential protocols for reliable communication. Emphasis is placed on both theoretical knowledge and practical application of networking concepts in real-world scenarios.

2. COURSE OBJECTIVE

The students will try to Learn:

1) To understand the concepts and fundamentals of computer networks.
2) To familiarize with layered models (OSI and TCP/IP) for communication between machines.
3) To learn the functionalities of each network layer and its protocols.
4) To gain knowledge of routing, subnetting, congestion control, and quality of service.
5) To explore application-level protocols for web and multimedia communication.

3. COURSE OUTCOMES

After successful completion of the course, students should be able to:

CO 1	Explain the fundamental concepts of computer networks and communication models.
CO 2	Describe the functions of each layer in the OSI and TCP/IP reference models.
CO 3	Apply subnetting and routing techniques for efficient network design.
CO 4	Analyze various protocols of data link, network, transport, and application layers.
CO 5	Demonstrate the use of network protocols in real-world applications such as DNS, HTTP, email, and streaming.

4. COURSE CONTENT

UNIT – I **(10L)**

Introduction: Network hardware, Network software, OSI and TCP/IP Reference models, Example Networks: ARPANET, Internet.

Physical Layer: Guided Transmission media – twisted pairs, coaxial cable, fiber optics; Wireless Transmission.

Data Link Layer: Design issues, framing, Error detection and correction.

UNIT – II **(10L)**

Elementary Data Link Protocols: Simplex protocol, Stop-and-wait protocol for error-free and noisy channels.
Sliding Window Protocols: One-bit sliding window, Go-Back-N, Selective Repeat, Example data link protocols.
Medium Access Sublayer: Channel allocation problem, Multiple access protocols (ALOHA, CSMA, collision-free protocols), Wireless LANs, Data link layer switching.

UNIT – III **(10L)**

Network Layer: Design issues, Routing algorithms (shortest path, flooding, hierarchical, broadcast, multicast, distance vector).
Congestion Control Algorithms, Quality of Service (QoS), Internetworking, Network Layer in the Internet.

UNIT – IV **(9L)**

Transport Layer: Transport Services, Elements of Transport protocols, Connection management, TCP and UDP protocols.

UNIT – V **(9L)**

Application Layer: Domain Name System (DNS), SNMP, Electronic Mail; The World Wide Web (HTTP), Streaming audio and video.

5. TEXT BOOKS

1) *Computer Networks* – Andrew S. Tanenbaum, David J. Wetherall, 5th Edition, Pearson Education/PHI.

6. REFERENCE BOOKS

1) *An Engineering Approach to Computer Networks* – S. Keshav, 2nd Edition, Pearson Education.
2) *Data Communications and Networking* – Behrouz A. Forouzan, 3rd Edition, TMH.

CO-PO-PSO Mapping

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2
CO 1	**3**	**2**	**1**	**1**	**2**	**-**	**-**	**-**	**1**	**2**	**-**	**2**		
CO 2	**3**	**2**	**2**	**1**	**2**	**-**	**-**	**-**	**1**	**2**	**-**	**2**		
CO 3	**3**	**3**	**3**	**2**	**3**	**-**	**-**	**-**	**2**	**2**	**-**	**3**		
CO 4	**3**	**3**	**3**	**3**	**3**	**-**	**-**	**-**	**2**	**3**	**-**	**3**		
CO 5	**2**	**2**	**2**	**2**	**3**	**-**	**-**	**-**	**2**	**3**	**-**	**3**		

COURSE CONTENT

SOFTWARE ENGINEERING								
II Year - II Semester: CSE (DS)								
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
DS405PC	**Core**	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		3	**-**	**-**	**3**	**40**	**60**	**100**
Contact Classes: 48	Tutorial Classes: Nil	Practical Classes: Nil				**Total Classes: 48**		
Prerequisite: Nil								

1. COURSE OVERVIEW

The **Software Engineering** course provides students with the principles, practices, and methodologies used in the development of reliable, scalable, and maintainable software systems. Students will learn software process models, requirements engineering, software design concepts, UML modeling, testing strategies, risk management, and quality assurance. Emphasis is given to applying theoretical concepts to real-world software development projects, ensuring exposure to both classical and agile approaches to software engineering.

2. COURSE OBJECTIVE

The students will try to Learn:

1) To gain working knowledge of techniques for estimation, design, testing, and quality management of large-scale software projects.
2) To understand various process models and the role of process frameworks such as CMMI.
3) To acquire skills in requirements engineering and preparation of Software Requirement Documents (SRDs).
4) To design and model systems using UML diagrams and software architecture patterns.
5) To understand software testing methodologies, software metrics, risk management, and quality assurance practices.

3. COURSE OUTCOMES

After successful completion of the course, students should be able to:

CO 1	Translate end-user requirements into structured system and software requirements using UML, and prepare a Software Requirement Document (SRD).
CO 2	Apply suitable software architectures and patterns to design systems, and evaluate alternative design strategies.
CO 3	Develop UML diagrams such as class diagrams, sequence diagrams, and use case diagrams for effective system modelling.
CO 4	Demonstrate knowledge of software testing strategies, apply black-box and white-box techniques, and prepare a testing report.
CO 5	Analyze and apply risk management strategies and software quality assurance practices for reliable software development.

4. COURSE CONTENT

UNIT – I **(10L)**

Introduction to Software Engineering: The evolving role of software, changing nature of software, software myths.

A Generic View of Process: Software engineering as layered technology, process framework, capability maturity model integration (CMMI).

Process Models: The Waterfall model, Spiral model, Agile methodology.

UNIT – II **(10L)**

Software Requirements: Functional and non-functional requirements, user requirements, system requirements, interface specification, software requirements document.

Requirements Engineering Process: Feasibility studies, requirements elicitation and analysis, requirements validation, requirements management.

UNIT – III **(10L)**

Design Engineering: Design process and quality, design concepts, design model.

Architectural Design: Software architecture, data design, architectural styles and patterns.

UML Modeling: Conceptual model of UML, structural modeling, class diagrams, sequence diagrams, collaboration diagrams, use case diagrams, component diagrams.

UNIT – IV **(9L)**

Testing Strategies: A strategic approach to software testing, conventional software test strategies, black-box and white-box testing, validation testing, system testing, debugging techniques.

Metrics for Process and Products: Software measurement, metrics for software quality.

UNIT – V **(9L)**

Risk Management: Reactive vs. proactive risk strategies, software risks, risk identification, risk projection, risk refinement, RMMM.

Quality Management: Quality concepts, software quality assurance (SQA), reviews, formal technical reviews, statistical SQA, software reliability, ISO 9000 standards.

5. TEXT BOOKS

1) *Software Engineering: A Practitioner's Approach* – Roger S. Pressman, 6th Edition, McGraw Hill.
2) *Software Engineering* – Ian Sommerville, 7th Edition, Pearson Education.

6. REFERENCE BOOKS

1) *The Unified Modeling Language User Guide* – Grady Booch, James Rambaugh, Ivar Jacobson, Pearson Education.
2) *Software Engineering: An Engineering Approach* – James F. Peters, Witold Pedrycz, John Wiley.
3) *Software Engineering: Principles and Practice* – Waman S. Jawadekar, McGraw-Hill.
4) *Fundamentals of Object-Oriented Design using UML* – Meilir Page-Jones, Pearson Education.

CO-PO-PSO Mapping

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2
CO 1	**3**	**2**	**1**	**1**	**2**					**2**	**3**	**2**		
CO 2	**3**	**2**	**2**	**1**	**2**					**2**	**3**	**2**		
CO 3	**2**	**3**	**3**	**2**	**2**				**1**	**2**	**3**	**3**		
CO 4	**3**	**3**	**3**	**2**	**2**				**1**	**2**	**3**	**3**		
CO 5	**2**	**2**	**3**	**3**	**3**				**1**	**3**	**3**	**3**		

COURSE CONTENT

OPERATING SYSTEMS LAB								
II Year - II Semester: CSE (DS)								
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
DS406PC	**Core**	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		-	**-**	**2**	**1**	**40**	**60**	**100**
Contact Classes: Nil	Tutorial Classes: Nil	Practical Classes: 32				**Total Classes: 32**		
Prerequisite: Nil								

1. COURSE OVERVIEW

This course provides practical exposure to the core concepts of operating systems through programming and simulation. Students will gain hands-on experience in implementing process scheduling, deadlock handling, memory management, and inter-process communication. The lab also emphasizes the use of UNIX/LINUX system calls for process and file management.

2. COURSE OBJECTIVE

The students will try to Learn:

1) To provide an understanding of the design aspects of operating system concepts through simulation.
2) To introduce UNIX commands and system call interfaces for process, memory, and file management.

3. COURSE OUTCOMES

After successful completion of the course, students should be able to:

CO 1	Simulate and implement CPU scheduling algorithms.
CO 2	Apply UNIX/LINUX system calls for file operations and process management.
CO 3	Demonstrate deadlock avoidance, prevention, and recovery mechanisms.
CO 4	Implement inter-process communication using pipes, FIFOs, message queues, and shared memory.
CO 5	Simulate memory management and page replacement algorithms.

4. LIST OF EXPERIMENTS

1) Write C programs to simulate the following CPU Scheduling algorithms:
   a) FCFS b) SJF c) Round Robin d) Priority Scheduling
2) Write programs using the I/O system calls of UNIX/LINUX operating system:
   open, read, write, close, fcntl, seek, stat, opendir, readdir.
3) Write a C program to simulate Banker's Algorithm for Deadlock Avoidance and Prevention.
4) Write a C program to implement the Producer–Consumer problem using semaphores with UNIX/LINUX system calls.
5) Write C programs to illustrate IPC mechanisms
   a) Pipes b) FIFOs c) Message Queues d) Shared Memory
6) Write C programs to simulate the following memory management techniques
   a) Paging b) Segmentation
7) Write C programs to simulate Page Replacement policies:
   a) FCFS b) LRU c) Optimal

5. TEXT BOOKS

1) Operating System Principles – Abraham Silberchatz, Peter B. Galvin, Greg Gagne, 7th Edition, John Wiley.
2) Advanced Programming in the UNIX Environment – W.R. Stevens, Pearson Education.

6. REFERENCE BOOKS

1) Operating Systems: Internals and Design Principles – William Stallings, 5th Edition, Pearson Education/PHI.
2) Operating System: A Design Approach – Crowley, TMH.
3) Modern Operating Systems – Andrew S. Tanenbaum, 2nd Edition, Pearson/PHI.
4) UNIX Programming Environment – Kernighan and Pike, PHI/Pearson Education.
5) UNIX Internals: The New Frontiers – U. Vahalia, Pearson Education.

CO-PO-PSO Mapping

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2
CO 1	**3**	**2**	**3**		**2**					**2**		**3**		
CO 2	**3**	**3**	**2**		**2**					**2**		**2**		
CO 3	**3**	**2**	**3**		**2**					**2**		**3**		
CO 4	**3**	**2**	**3**	**2**	**2**					**2**		**2**		
CO 5	**3**	**2**	**3**	**2**	**2**					**2**		**3**		

COURSE CONTENT

COMPUTER NETWORKS LAB								
II Year - II Semester: CSE (DS)								
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
DS407PC	**Core**	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		-	**-**	**2**	**1**	**40**	**60**	**100**
Contact Classes: Nil	Tutorial Classes: Nil	Practical Classes: 32				**Total Classes: 32**		
Prerequisite: Nil								

1. COURSE OVERVIEW

This course provides practical exposure to computer networks and communication protocols. Students will learn to implement framing, error detection, routing, congestion control, and encryption techniques. They will also gain hands-on experience with Wireshark, Nmap, and NS2 simulators for traffic analysis and performance evaluation of network topologies.

2. COURSE OBJECTIVE

The students will try to Learn:

1) To understand the working principle of various communication protocols.
2) To understand the network simulator environment and visualize a network topology.
3) To analyze traffic flow and the contents of protocol frames.

3. COURSE OUTCOMES

After successful completion of the course, students should be able to:

CO 1	Implement data link layer framing methods.
CO 2	Analyze error detection and error correction codes.
CO 3	Implement and analyze routing and congestion issues in network design.
CO 4	Implement encoding and decoding techniques used in the presentation layer.
CO 5	Work with different network tools for packet analysis and network simulation.

4. LIST OF EXPERIMENTS

1) Implement the data link layer framing methods such as character, character-stuffing and bit stuffing.
2) Write a program to compute CRC code for the polynomials CRC-12, CRC-16 and CRC-CCIP.
3) Develop a simple data link layer that performs the flow control using the sliding window protocol, and loss recovery using the Go-Back-N mechanism.
4) Implement Dijkstra's algorithm to compute the shortest path through a network.
5) Take an example subnet of hosts and obtain a broadcast tree for the subnet.
6) Implement distance vector routing algorithm for obtaining routing tables at each node.
7) Implement data encryption and data decryption.
8) Write a program for congestion control using Leaky Bucket algorithm.
9) Write a program for frame sorting techniques used in buffers.

10) Wireshark:

i. Packet Capture using Wireshark

ii. Starting Wireshark

iii. Viewing Captured Traffic

iv. Analysis, Statistics & Filters

v. Running Nmap Scan

vi. Operating System Detection using Nmap

11) NS2 Simulator:

i. Introduction to NS2 Simulator

ii. Simulate to find the number of packets dropped

iii. Simulate to find the number of packets dropped by TCP/UDP

iv. Simulate to find the number of packets dropped due to congestion

v. Simulate to compare data rate & throughput

vi. Simulate to plot congestion for different source/destination

vii. Simulate to determine the performance with respect to transmission of packets

5. TEXT BOOKS

1) *Computer Networks*, Andrew S. Tanenbaum, David J. Wetherall, 5th Edition, Pearson Education/PHI.

6. REFERENCE BOOKS

1) *An Engineering Approach to Computer Networks*, S. Keshav, 2nd Edition, Pearson Education.
2) *Data Communications and Networking*, Behrouz A. Forouzan, 3rd Edition, TMH.

CO-PO-PSO Mapping

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2
CO 1	**3**	**2**	**2**		**2**					**1**		**2**		
CO 2	**3**	**3**	**2**		**2**					**1**		**2**		
CO 3	**3**	**2**	**3**	**2**	**2**					**2**		**2**		
CO 4	**3**	**2**	**3**	**2**	**2**					**2**		**2**		
CO 5	**3**	**2**	**2**	**3**	**3**					**2**		**3**		

COURSE CONTENT

DATA VISUALIZATION – R PROGRAMMING / POWER BI LAB								
II Year - II Semester: CSE (DS)								
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
DS409PC	**Core**	**L**	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
		-	**-**	**2**	**1**	**40**	**60**	**100**
Contact Classes: Nil	Tutorial Classes: Nil	Practical Classes: 32				**Total Classes: 32**		
Prerequisite: Nil								

1. COURSE OVERVIEW

This course introduces students to the principles and practices of data visualization using **R Programming and Power BI/Tableau**. Students will gain hands-on experience in preparing, analyzing, and visualizing data, with an emphasis on developing effective dashboards and storytelling applications. The course enhances decision-making skills through visualization of patterns, trends, and relationships in real-world data.

2. COURSE OBJECTIVE

The students will try to Learn:

1) Effective use of **Business Intelligence (BI) tools** such as Tableau/Power BI for visualization.
2) To discern patterns and relationships in data using visual techniques.
3) To build **interactive dashboards** for analytics.
4) To communicate results clearly and concisely through visualization.
5) To handle **different formats of datasets** and apply visualization techniques effectively.

3. COURSE OUTCOMES

After successful completion of the course, students should be able to:

CO 1	Understand and import data into Tableau/Power BI/R.
CO 2	Apply concepts of Dimensions, Measures, and Calculations for data analysis.
CO 3	Create visual layouts, charts, and graphs with formatting and customization.
CO 4	Build interactive dashboards and storytelling visualizations.
CO 5	Apply visualization techniques to real-world datasets and communicate solutions effectively.

4. LIST OF LAB PROBLEMS

1) Understanding Data: Identifying data sources, basics of building visualizations, creating the first visualization.
2) Connecting data in Tableau/Power BI: Handling data file formats, creating basic charts (line, bar, tree maps), using the **Show Me panel**.
3) Tableau Calculations: SUM, AVG, Aggregates, creating custom calculations and fields.
4) Applying calculations to visualizations, formatting tools and menus, formatting specific parts of the view.
5) Editing and formatting axes, manipulating and pivoting Tableau data.
6) Structuring data, sorting and filtering datasets, pivoting data for better visualization.
7) Advanced Visualization Tools: Using Filters, Detail Panel, Size Panels, customizing tooltips, formatting data with colors.
8) Creating Dashboards & Storytelling: Designing dashboards, interactivity, publishing, and distribution.
9) Tableau File Types: Publishing to Tableau Online, sharing, exporting visualizations.
10) Creating custom charts: Cyclical data visualizations, circular area charts, dual-axis charts.

5. REFERENCE BOOKS

1) *Microsoft Power BI Cookbook*, Brett Powell, 2nd Edition.
2) *R Programming for Data Science*, Roger D. Peng.
3) *The Art of R Programming*, Norman Matloff, Cengage Learning India.

CO-PO-PSO Mapping

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2
CO 1	**2**	**2**	**2**		**3**					**1**		**2**		
CO 2	**3**	**2**	**3**		**3**					**2**		**2**		
CO 3	**3**	**2**	**3**	**2**	**3**					**2**		**2**		
CO 4	**3**	**3**	**3**	**2**	**3**					**2**		**3**		
CO 5	**3**	**2**	**3**	**3**	**3**					**2**		**3**		

COURSE CONTENT

INTELLECTUAL PROPERTY RIGHTS								
II Year - II Semester: CSE (DS)								
Course Code	**Category**	**Hours/Week**			**Credits**	**Maximum Marks**		
		L	**T**	**P**	**C**	**CIA**	**SEE**	**Total**
***MC410**	**Mandatory**	**3**	**-**	**-**	**-**	**40**	**60**	**100**
Contact Classes: 48	Tutorial Classes: Nil	Practical Classes: Nil				**Total Classes: 48**		
Prerequisite:								

1. COURSE OVERVIEW

The Intellectual Property Rights (IPR) course is designed to provide students with a comprehensive understanding of the legal frameworks that protect creative and innovative works. It covers the principles, laws, and practices surrounding the protection of intellectual property such as: Patents, Trademarks, Copyrights, Designs and Trade Secrets.

2. COURSE OBJECTIVE

The students will try to Learn:

1) Significance of intellectual property and its protection.
2) Introduce various forms of intellectual property.

3. COURSE OUTCOMES

After successful completion of the course, students should be able to:

CO 1	Distinguish and Explain various forms of IPRs.
CO 2	Identify criteria to fit one's own intellectual work in particular form of IPRs.
CO 3	Apply statutory provisions to protect particular form of IPRs.
CO 4	Appraise new developments in IPR laws at national and international level
CO 5	Understand the new development of intellectual property.

4. COURSE CONTENT

UNIT – I **(10L)**

Introduction to Intellectual property: Introduction, types of intellectual property, international organizations, agencies and treaties, importance of intellectual property rights.

UNIT – II **(10L)**

Trade Marks: Purpose and function of trademarks, acquisition of trade mark rights, protectable matter, selecting, and evaluating trade mark, trade mark registration processes.

UNIT – III **(10L)**

Law of copyrights: Fundamental of copyright law, originality of material, rights of reproduction, rights to perform the work publicly, copyright ownership issues, copyright registration, notice of copyright, international copyright law.

Law of patents: Foundation of patent law, patent searching process, ownership rights and transfer

UNIT – IV **(9L)**

Trade Secrets: Trade secret law, determination of trade secret status, liability for misappropriations of trade secrets, protection for submission, trade secret litigation.

Unfair competition: Misappropriation right of publicity, false advertising.

UNIT – V **(9L)**

New development of intellectual property: new developments in trade mark law; copyright law, patent law, intellectual property audits.

International overview on intellectual property, international – trade mark law, copyright law, international patent law, and international development in trade secrets law.

5. TEXT BOOKS

1) Intellectual property right, Deborah. E. Bouchoux, Cengage learning.

6. REFERENCE BOOKS

1) Intellectual property right – Unleashing the knowledge economy, prabuddha ganguli, Tata McGraw Hill Publishing company ltd.

CO-PO-PSO Mapping

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO 1	PSO 2
CO 1	**3**	**2**	**2**		**2**					**2**	**2**	**2**		
CO 2	**3**	**3**	**2**		**2**					**2**	**2**	**2**		
CO 3	**3**	**2**	**3**	**2**	**2**					**2**	**2**	**2**		
CO 4	**3**	**2**	**3**	**2**	**2**					**2**	**2**	**2**		
CO 5	**3**	**2**	**2**	**3**	**3**					**2**	**2**	**2**		