For R25 Regulation

MUFFAKHAM JAH COLLEGE OF ENGINEERING AND TECHNOLOGY

(AUTONOMOUS)

ACCREDITED BY NBA & NAAC WITH 'A+' GRADE

Banjara Hills, Hyderabad-34

Approved by A.I.C.T.E., New Delhi and

Affiliated to Osmania University, Hyderabad.

B.E (CSE) SEMESTER-III

SEMESTER-III			Scheme of Instruction				Scheme of Examination			Credits
S. No	Course Code	Name of the Course	Lecture	Tutorial	Practical	Contact Hrs/Wk	CIE	SEE	Duration in Hrs	
1	25ES211EC	Integrated Electronics	3	-	-	3	40	60	3	3
2	25PC212CS	Data Structures	3	1	-	4	40	60	3	4
3	25PC213CS	Discrete Mathematics	3	-	-	3	40	60	3	3
4	25PC214CS	OOP using JAVA	3	-	-	3	40	60	3	3
5	25PC215CS	Computer Organization & Architecture	3	-	-	3	40	60	3	3
Practical / Laboratory Courses										
6	25ES251EC	Integrated Electronics Lab	-	-	2	2	40	60	3	1
7	25PC252CS	Data Structures Lab	-	-	2	2	40	60	3	1
8	25PC253CS	OOP using JAVA Lab	-	-	2	2	40	60	3	1
9	25ES254EE	Skill Development Course-I (Drone Technology)	-	-	2	2	50	-	-	1
		Total	**15**	**1**	**08**	**24**	**370**	**480**	**24**	**20**

MUFFAKHAM JAH COLLEGE OF ENGINEERING AND TECHNOLOGY

(AUTONOMOUS)

ACCREDITED BY NBA & NAAC WITH 'A+' GRADE

Banjara Hills, Hyderabad-34

Approved by A.I.C.T.E., New Delhi and

Affiliated to Osmania University, Hyderabad.

B.E (CSE) SEMESTER-IV

SEMESTER-IV			Scheme of Instruction				Scheme of Examination			Credits
S. No	Course Code	Name of the Course	Lecture	Tutorial	Practical	Contact Hrs/Wk	CIE	SEE	Duration in Hrs	
1	25BS211MT	Probability & Statistics	3	-	-	3	40	60	3	3
2	25PC222CS	Software Engineering	3	-	-	3	40	60	3	3
3	25PC223CS	Operating Systems	3	-	-	3	40	60	3	3
4	25PC224CS	Database Management Systems	3	1	-	4	40	60	3	4
5	25PC225CS	Web Technologies	3	-	-	3	40	60	3	3
Practical / Laboratory Courses										
6	25PC255CS	Operating Systems Lab	-	-	2	2	40	60	3	1
7	25PC256CS	Database Management Systems Lab	-	-	2	2	40	60	3	1
8	25PC257CS	Web Technologies Lab	-	-	2	2	40	60	3	1
9	25PW258CS	Mini Project	-	-	6	6	40	60	3	3
		Total	**15**	**1**	**12**	**28**	**360**	**540**	**27**	**22**

For R25 Regulation

MUFFAKHAM JAH COLLEGE OF ENGINEERING & TECHNOLOGY

(SULTAN-UL-ULOOM EDUCATION SOCIETY)

An Autonomous Institution

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

INTEGRATED ELECTRONICS

SYLLABUS FOR B.E. III-SEMESTER

L:T:P (Hrs./week):3:0:0	SEE Marks : 60	Course Code: 25ES211EC
Credits : 3	CIE Marks : 40	Duration of SEE : 3 Hours

COURSE OBJECTIVES	COURSE OUTCOMES
The course is taught with the objectives of enabling the student to:	On completion of the course, students will be able to
1. Understand the characteristics of diodes and its applications.	1. Study and analyze the rectifiers and regulator circuits.
2.Understand the design concepts of biasing of BJT and FET	2. Study and analyze the performance of BJTs,FETs on the basis of their operation and working.
3. Understand the fundamental principles of digital hardware design and number systems.	3. Explain the fundamentals of digital hardware design, logic gates, Boolean algebra, and logic synthesis.
4. Apply Boolean algebra and logic minimization techniques to implement combinational circuits efficiently.	4. Design and optimize combinational circuits using K-Map and Quine-Mc Cluskey methods.
5. Explore the architecture and design methodologies using programmable logic devices like PALs, PLAs, CPLDs, and FPGAs	5. Implement combinational logic using PLDs such as PALs, PLAs, CPLDs, and FPGAs.

CO-PO and CO-PSO mapping

CO/PO &PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	2	-	-	-	-	-	-	0	2	1	-	1
CO2	3	2	1	-	1	-	-	-	-	1	1	1	-	3
CO3	3	2	1	1	-	-	-	-	-	-	-	1	-	1
CO4	3	2	3	2	-	-	-	-	-	-	-	2	-	2
CO5	3	2	3	2	1	-	-	-	-	-	-	3	1	3

UNIT I

Introduction to Semiconductor Diodes: Formation of PN junction, V-I characteristics of PN junction diode. Half-wave rectifier, Full-wave rectifier, performance parameters—Peak Inverse Voltage (PIV), ripple factor, rectification efficiency, introduction to filter circuits (capacitor, inductor and π-filters). Applications of diodes. Voltage Regulation: Zener diode characteristics, Zener diode as a voltage regulator.

UNIT II

Bipolar Junction Transistor: BJT construction and working, modes of operation, configurations of BJT (CB, CE, CC). Junction Field Effect Transistor: Construction and working of JFET, V-I characteristics of JFET.

UNIT III

Number Systems and Combinational Logic Fundamentals: Number systems and codes. Introduction to logic circuits – Variables and functions, Logic gates and networks. Boolean algebra, Design Examples Using Logic Gate. Optimized implementation of logic functions using K-Map.

UNIT IV

Combinational Circuit Design and Flip-Flops: Adders, Subtractors, Multiplexers, Demultiplexers, Decoders, Encoders and Code converters: Binary to Gray, Gray to Binary and BCD to seven segments, Introduction to sequential circuit, Memory elements: Basic Latch, Gated SR Latch, gated D Latch, T Flip-flop and JK Flip-flop.

UNIT V

Design of combinational circuits using Programmable Logic Devices (PLDs): General structure of a Programmable Array Logic (PAL), Programmable Logic Arrays (PLAs), Structure of CPLDs and FPGAs, 2-input and 3-input lookup tables (LUTs).

Suggested Text Books:
1. Robert Boylestad L. and Louis Nashelsky, Electronic Devices and Circuit Theory, PHI,2007. 2nd Edition, Prentice Hall of India, 2006.
2. Morris Mano and Michael D. Ciletti, "Digital Design", Pearson, 6/e, 2021.
3. ZviKohavi, "Switching and Finite Automata Theory", 3/e, Cambridge University Press New Delhi, 2011
Reference Text Books:
1.John F. Wakerly, *Digital Design: Principles and Practices*, 5th Edition, Pearson Education, 2018.
2.Ronald J. Tocci, Neal S. Widmer, and Gregory L. Moss**,** *Digital Systems: Principles and Applications*, 12th Edition, Pearson Education, 2016.
3. Basic electronics by J B Gupta

For R25 Regulation

MUFFAKHAM JAH COLLEGE OF ENGINEERING & TECHNOLOGY

(SULTAN-UL-ULOOM EDUCATION SOCIETY)

ACCREDITED BY NBA and NAAC WITH 'A+' GRADE

(An Autonomous Institution)

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

DATA STRUCTURES

SYLLABUS FOR B.E. III-SEMESTER

L:T:P (Hrs./week):3:1:0	SEE Marks : 60	Course Code :25PC212CS
Credits : 4	CIE Marks : 40	Duration of SEE : 3 Hours

COURSE OBJECTIVES	COURSE OUTCOMES
1. Develop the ability to analyze algorithm efficiency and implement Abstract Data Types using appropriate Data Structures. 2. Understand and implement linear and non-linear data structures for solving computational problems 3. Apply tree-based and hashing techniques for efficient storage, retrieval and manipulation of data 4. Analyze and implement searching, sorting and graph based algorithms using appropriate data structures.	1. Analyze algorithm efficiency using asymptotic notations and implement array-based data structures and string processing techniques. 2. Implement linked list and hashing techniques for efficient storage, retrieval, and manipulation of data. 3. Apply stack and queue data structures to solve computational problems involving expression evaluation and scheduling applications. 4. Construct and analyze tree-based data structures including BSTs, AVL trees, and heaps for efficient searching and sorting. 5. Apply graph algorithms, sorting, and searching techniques to solve real-world computational problems.

CO-PO and CO-PSO mapping

CO/PO &PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	3	2	2	-	1	-	1	1	2	3	2	2
CO2	3	3	3	2	2	-	1	-	1	1	2	3	2	2
CO3	3	3	3	2	2	-	1	-	1	1	2	3	2	2
CO4	3	3	3	2	2	-	1	-	1	1	2	3	3	3
CO5	3	3	3	2	2	-	1	-	1	1	2	3	3	3

UNIT I

Algorithms: Introduction, Algorithm Specifications, Recursive Algorithms, Performance Analysis of an algorithm- Time and Space Complexity, Asymptotic Notations**.**

Arrays: Arrays ADT, Polynomials, Sparse matrices, Strings-ADT, Pattern Matching.

UNIT II

Linked Lists: Singly Linked Lists and Chains, Polynomials, Operations for Circularly linked lists, Equivalence Classes, Sparse matrices, Doubly Linked Lists.

UNIT III

Stacks and Queues: Stacks, Stacks using Arrays, and Stacks using dynamic arrays, Evaluation of Expressions, Evaluating Postfix Expression, Infix to Postfix. Linked Stacks.
Queues: Queues ADT, operations, Circular Queues, Linked Queues.
Hashing: Static Hashing, Hash Tables, Hash Functions, Overflow Handling

UNIT IV

Trees: Introduction, Binary Trees, Binary Tree Traversals, Heaps**.**
Binary Search trees (BST): Definition, Searching an element, Insertion into a BST, Deletion from a BST**.**
Efficient Binary Search Trees: AVL Trees: Definition, Searching an element, Insertion into a AVL.

UNIT V

Graphs: Graph Abstract Data Type, Elementary Graph operations (DFS and BFS), Minimum Cost Spanning Trees (Prim's and Kruskal's Algorithms).
Sorting and Searching: Insertion sort, Quick sort, Merge sort, Heap sort, Shell sort, Sorting on Several Keys, List and Table Sorts, Summary of Internal Sorting, Linear and Binary Search algorithms.

Textbooks:

1. Horowitz E, Sahni S and Susan Anderson-Freed, Fundamentals of Data structures in C, 2nd Edition (Reprint 2024), Universities Press.

References:

1. Tanenbaum A. M, Langsam Y. Augenstein M. J, Data Structures using C, Second Edition (2008), Pearson.
2. Gilberg R. F and Forouzan B. A, Data structures: A Pseudocode Approach with C, Second Edition (2007), Cengage Leaming.
3. Thomas H. Cormen, Charles E. Leiserson, Ronald L Rivest, Clifford Stein, Introduction to Algorithms, Fourth Edition (2022), MIT Press.
4. Kushwaha D. S and Misra A.K, Data structures A Programming Approach with C, Second Edition (2014), PHI.

For R25 Regulation

MUFFAKHAM JAH COLLEGE OF ENGINEERING & TECHNOLOGY

(SULTAN-UL-ULOOM EDUCATION SOCIETY)

ACCREDITED BY NBA and NAAC WITH 'A+' GRADE

(An Autonomous Institution)

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

DISCRETE MATHEMATICS

SYLLABUS FOR B.E. III-SEMESTER

L:T:P (Hrs./week):3:0:0	SEE Marks : 60	Course Code :25PC213CS
Credits : 3	CIE Marks : 40	Duration of SEE : 3 Hours

S. No.	Course Objectives	Course Outcomes
1.	**Understand** the fundamental concepts of mathematical logic, predicate logic, and proof techniques used in discrete mathematics.	Apply propositional and predicate logic, normal forms, quantifiers, and proof techniques to formulate and validate logical arguments.
2.	**Apply** the principles of set theory, relations, functions, lattices, and partial ordering to solve mathematical and computational problems.	Analyze and solve problems involving sets, relations, functions, lattices, equivalence relations, partial orders, and Hasse diagrams.
3.	**Analyze** graph-theoretic concepts, trees, and their applications in computer science, including graph traversal, graph coloring, and spanning trees.	Model and solve real-world problems using graph theory and trees, including graph traversal, connectivity, graph coloring, Euler and Hamilton paths, and minimum spanning trees.
4.	**Develop** problem-solving skills using combinatorial techniques, counting principles, permutations, combinations, binomial theorem, inclusion-exclusion principle, and pigeonhole principle.	Employ combinatorial methods, including counting principles, permutations, combinations, binomial theorem, inclusion-exclusion principle, and pigeonhole principle, to solve discrete mathematical problems.
5.	**Solve** recurrence relations and utilize generating functions to analyze algorithms and discrete structures.	Formulate and solve linear recurrence relations and apply generating functions and divide-and-conquer techniques in algorithm analysis.

COs	POs											PSOs		
	1	2	3	4	5	6	7	8	9	10	11	1	2	3
1	3	3	1	1	-	-	-	-	-	-	1	3	2	1
2	3	3	1	1	-	-	-	-	-	-	1	3	2	1
3	3	3	2	2	1	-	-	1	-	-	1	3	2	3
4	3	3	1	2	-	-	-	-	-	-	1	3	2	3
5	3	3	2	2	1	-	1	-	-	-	2	3	2	3

UNIT I

Mathematical Logic: Statements and Notations, Connectives, Well-Formed Formulas, Truth Tables, Tautology, Equivalence, Implication, Normal Forms, Quantifiers, Universal Quantifiers.

Predicates: Predicate Logic, Free & Bound Variables, Rules of Inference, Consistency, Proof of Contradiction.

UNIT II

Set Theory and Relations: Basic Concepts of Set Theory, Relations and Ordering, Properties of Binary Relations, Equivalence, Transitive Closure, Compatibility and Partial Ordering Relations, Hasse Diagram.

Functions: Composition of Functions, Inverse Functions, Recursive Functions, Lattice and Its Properties.

UNIT III

Graphs: Graphs and Graph Models, Graph Terminology and Special Types of Graphs, Subgraphs, Representing Graphs and Graph Isomorphism, Connectivity, Euler and Hamilton Paths, Euler Circuits, Planar Graphs, Graph Coloring, Chromatic Numbers.

Trees: Introduction to Trees, Properties, Applications of Trees, Tree Traversal, Minimal Spanning Trees.

UNIT IV

Elementary Combinatorics: Basics of counting, Combinations & Permutations with and without Repetitions and Constrained Repetitions.

Binomial Theorem: Binomial Coefficients, Binomial and Multinomial theorems, The Principle of Inclusion — Exclusion and Its Applications, Pigeon Hole Principle and Its Applications.

UNIT V

Recurrence Relations: Recurrence Relations, Solving Linear Homogenous Recurrence Relations, Solving Linear Non-Homogenous Recurrence Relations, Divide and Conquer Algorithms and Recurrence Relations, Generating Functions.

TEXTBOOKS

1. Discrete Mathematics and its Applications, Kenneth H. Rosen, VIII Edition, McGraw Hill Education, 2019.

REFERENCE BOOKS

1 Elements of Discrete Mathematics- A Computer Oriented Approach- C. L. Liu, D. P. Mohapatra, lll Edition, Tata Mc Graw Hill, 2017.

2 Discrete Mathematics for Computer Scientists & Mathematicians, Joe L. Mott, A. Kandel,T. P. Baker, II Edition, PHI, 2015.

3 Discrete Mathematical Structures Theory and Application- Malik & Sen, I Edition, Cengage Learning, 2012.

4 Discrete Mathematics with Applications, Thomas Koshy, I Edition, Elsevier, 2005.

For R25 Regulation

MUFFAKHAM JAH COLLEGE OF ENGINEERING & TECHNOLOGY

(SULTAN-UL-ULOOM EDUCATION SOCIETY)

ACCREDITED BY NBA and NAAC WITH 'A+' GRADE

(An Autonomous Institution)

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

OOP using Java

SYLLABUS FOR B.E. III-SEMESTER

L: T: P (Hrs. / Week) 3: 0 : 0	SEE Marks : 60	Course Code : 25PC214CS
Credits : 3	CIE Marks : 40	Duration of the SEE : 03 Hours

S. No.	Course Objectives	Course Outcomes
1	To understand object-oriented programming principles.	Explain OOP principles and Java program structure
2	To develop Java programs using core and advanced OOP concepts.	Develop Java programs using classes, inheritance, interfaces and exception handling
3	To implement data handling using Collections and file processing.	Implement data structures using Java Collections and Generics
4	To apply multithreading concepts in Java applications.	Demonstrate multithreading and file handling concepts
5	To develop simple database-driven applications using JDBC.	Build simple database applications using JDBC

COs \ POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	1	-	-	1	-	-	-	-	-	1	3	1	-
CO2	3	2	3	-	2	-	-	-	-	-	2	2	3	2
CO3	3	3	2	1	2	-	-	-	-	-	2	1	2	3
CO4	3	2	2	1	2	-	-	-	-	-	2	2	2	2
CO5	2	2	3	-	3	-	-	2	1	1	2	1	3	3

UNIT-I

Java Fundamentals & OOP

OOP principles (Abstraction, Encapsulation, Inheritance, Polymorphism), Java Architecture (JVM, JRE, JDK) , Structure of Java Program, Data Types, Variables, Type Casting, Operators & Control Statements, Arrays, Classes and Objects, Constructors, Methods & Overloading, this, static, final, String and StringBuilder classes.

UNIT-II

Advanced OOP & Exception Handling

Inheritance & Method overriding, super keyword, Abstract Classes, Interfaces (including Functional Interfaces), Packages & Access Modifiers, Exception Handling: try-catch-finally-throw-throws, Custom Exceptions, **Introduction to Lambda Expressions** (Basic syntax only)

UNIT-III

Collections & Generics

Generics, Overview of Java Collections Framework, List, Set, Map Interfaces, ArrayList, LinkedList, HashSet, TreeSet, HashMap classes, Comparable & Comparator, Iterators, **Basic Stream API** (**filter, map – intro level**)

UNIT-IV

Multithreading & File Handling

Java Thread Model, Creating Threads (Thread class & Runnable), Thread Lifecycle, Synchronization, Inter-thread Communication, AutoBoxing, Java Input/Output: exploring java.io, Java I/O classes and interfaces, File Handling: File class, Byte Streams, Character Streams, Serialization

UNIT-V

JDBC & Mini Application

JDBC Architecture, Types of JDBC Drivers, steps to Connect to Database, Statement vs PreparedStatement, CRUD Operations, Basic Transaction Handling, Mini **Case Study: Student Management / Employee Record System**

Recommended Text Books

Primary Text:

1. The Complete Reference Java, 12th Edition – Herbert Schildt
2. JDBC: Practical Guide for Java Programmers, 2nd Edition Gregory D. Speegle

Reference:

1. Head First Java, 3rd Edition – Kathy Sierra & Bert Bates
2. Core Java, Volume I: Fundamentals, 12th Edition - Cay S. Horstmann
3. Database Programming with JDBC and Java, 2nd Edition - George

E-learning resources:

1. https://docs.oracle.com/javase/tutorial/java
2. https://spoken-tutorial.org/tutorial-search/?search_foss=Java&search_language=English
3. https://docs.oracle.com/javase/tutorial/jdbc/basics/index.html

MUFFAKHAM JAH COLLEGE OF ENGINEERING & TECHNOLOGY

(SULTAN-UL-ULOOM EDUCATION SOCIETY)

ACCREDITED BY NBA and NAAC WITH 'A+' GRADE

(An Autonomous Institution)

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

COMPUTER ORGANIZATION AND ARCHITECTURE

SYLLABUS FOR B.E. III-SEMESTER

L:T:P (Hrs./week):3:0:0	SEE Marks : 60	Course Code :25PC215CS
Credits : 3	CIE Marks : 40	Duration of SEE : 3 Hours

S. No.	Course Objectives	Course Outcomes
1	Understand the fundamental concepts of computer organization, computer architecture, and data representation techniques.	**CO1:** Identify the fundamental concepts of computer organization, computer architecture, data representation, and register transfer operations.
2	Explain register transfer operations, microoperations, instruction execution, and control unit organization in a computer system	**CO2:** Explain microoperations, basic computer organization, instruction execution, and control unit organization.
3	Understand CPU organization, instruction formats, addressing modes, and techniques used for processor performance enhancement.	**CO3:** Apply CPU organization concepts, instruction formats, and addressing modes to explain instruction execution and program control.
4	Describe the organization and operation of memory hierarchy and input/output systems in modern computers.	**CO4: :** Explain the principles of pipeline processing, parallel processing, and RISC/CISC architectures used in computer systems.
5	Develop a strong foundation for advanced studies in computer architecture, embedded systems, and computer system design.	**CO5:** Analyze memory hierarchy and input/output organization to evaluate their impact on computer system performance.

Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	1									1	3	1	1
CO2	3	2			2						1	3	2	1
CO3	3	2	2		2							3	2	1
CO4	2	2	1	1	2						1	3	2	2
CO5	2	3	1	1	2						1	2	3	2

UNIT–I:

Fundamental Concepts and Data Representation

Introduction to computer systems, Functional units of a computer, Basic operational concepts, Computer organization, Computer design, Computer architecture. Data types, Number systems, Complements, Fixed-point representation, Floating-point representation, Error Detection.

Register Transfer Operations

Register transfer language, Register transfer , Bus and memory transfers.

UNIT–II:

Microoperations

Arithmetic microoperations, Logic microoperations, Shift microoperations, Arithmetic logic shift unit.

Basic Computer Organization

Instruction Codes, Computer registers , Common Bus system, Computer instructions, Timing and control, Instruction cycle.

UNIT–III:

Instruction Set

Register-reference instructions, Memory-reference instructions, Input-output and Interrupt .

Control Unit Design

Control memory, Address sequencing, Microprogram example, Design of control unit

UNIT–IV:

CPU Organization

General register organization, Stack organization, Instruction formats, Addressing modes, Data transfer and manipulation, program control.

Parallel Processing and Pipeline

Parallel Processing, Pipeline, Instruction pipeline, RISC,CISC

UNIT–V:

Memory Organization

Memory hierarchy, Main memory, Auxiliary memory, Associative memory, Cache memory, Virtual memory.

Input–Output Organization

I/O interface, Asynchronous serial transfer, asynchronous communication interface, Modes of Transfer, Priority interrupt, DMA,CPU-IOP.

Text Books:

1. Computer System Architecture, M. Morris Mano, 3rd Edition, Pearson Education.
2. Computer Organization and Architecture: Designing for Performance, William Stallings, 11th Edition, Pearson.

Reference Books:

1. Computer Organization and Embedded Systems, Carl Hamacher et al., 6th Edition, McGraw-Hill Education.
2. Structured Computer Organization, Andrew S. Tanenbaum and Todd Austin, 6th Edition, Pearson Education.

Online Resources:

1. NPTEL – Computer Organization and Architecture
   https://nptel.ac.in/courses/106106166
2. Computer Organization Tutorials – GeeksforGeeks
   https://www.geeksforgeeks.org/computer-organization-and-architecture-tutorials/

For R25 Regulation

MUFFAKHAM JAH COLLEGE OF ENGINEERING & TECHNOLOGY

(SULTAN-UL-ULOOM EDUCATION SOCIETY)

ACCREDITED BY NBA and NAAC WITH 'A+' GRADE

(An Autonomous Institution)

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

INTEGRATED ELECTRONICS LAB

SYLLABUS FOR B.E. III-SEMESTER

L:T:P (Hrs./week):0:0:2	SEE Marks : 60	Course Code : 25ES251EC
Credits : 1	CIE Marks : 40	Duration of SEE : 3 Hours

Course Objectives : The course is taught with the objectives of enabling the student to:	**Course Outcomes :** On completion of this course, the student will be able to :
1. To understand the characteristics of diodes and transistor configurations.	1. Ability to design diode circuits & understand the application of Zener diode.
2. To understand the design concepts of biasing of BJT and FET	2. Ability to analyze characteristics of BJTs &FETS.
3. To understand and verify the functionality of basic logic gates and standard combinational logic circuits.	3. Verify the truth tables of basic logic gates and implement simple code converters using logic gates.
4. To design and realize combinational circuits using universal gates, multiplexers, and decoders.	4. Design and implement half and full adders/subtractors using universal gates, MUX, and decoders.
5. To verify and analyze the operation of various flip-flops, conversions, design and simulate sequential circuits.	5. 5. Analyze the working of different flip-flops and perform flip-flop conversions

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO1**	**PSO2**	**PSO3**
CO1	3	2	2	2	3	-	-	-	-	-	-	3	2	1
CO2	3	3	3	2	3	-	-	-	-	-	-	3	2	1
CO3	3	3	2	3	3	-	-	-	-	-	-	2	3	2
CO4	2	3	2	3	3	-	-	-	-	-	-	2	3	2
CO5	3	2	3	2	3	-	-	-	-	-	-	2	3	2

List of Experiments: PART A
1. Characteristics of Semiconductor diode Silicon Germanium and Zener.
2. Full-wave rectifier with and without filter
3. Static Characteristics of BJT-Common Emitter
4. Static Characteristics of FET.
5. Common Emitter Amplifier
List of Experiments: PART B
1. Verification of truth tables of Logic gates and realization of Binary to Gray and Gray to Binary code converters.
2. Realization of Half adder/ subtractor using universal logic gates.
3. Realization of Full adder/ subtractor using universal logic gates.

4. Realization of Full adder/Sub using MUX and Decoder
5. Verification of truth tables of Flip Flops and Flip flop conversions form one form to the other.
6. Design and simulation of basic logic gates and verification of their truth tables.
7. Design and simulation of Combinational circuits such as adders, subtractors, multiplexers, and demultiplexers

Note: The Preferred Simulation Tools are LTspice – for analog circuit simulation, Multisim /PSpice – for analog and mixed-signal circuit simulation, Logisim – for basic digital logic circuits**.**

Suggested Text Books:
1. David A. Bell, Lab Manual for Electronic Devices and Circuits, 4th Edition, PHI.
2. Zbar, Malvino, Miller, Basic Electronics: A Text Lab Manual.
3. M. H. Rashid, SPICE for Circuits and Electronics Using PSpice, Pearson.
4. A. Sudhakar and Shyammohan S. Palli, Circuits and Networks: Analysis and Synthesis, McGraw Hill Education.

Note:-Minimum four experiments from Part A and Minimum Six from Part B must be covered.

For R25 Regulation

MUFFAKHAM JAH COLLEGE OF ENGINEERING & TECHNOLOGY

(SULTAN-UL-ULOOM EDUCATION SOCIETY)

ACCREDITED BY NBA and NAAC WITH 'A+' GRADE

(An Autonomous Institution)

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

DATA STRUCTURES LAB

SYLLABUS FOR B.E. III-SEMESTER

L:T:P (Hrs./week):0:0:2	SEE Marks : 60	Course Code : 25PC252CS
Credits : 1	CIE Marks : 40	Duration of SEE : 3 Hours

COURSE OBJECTIVES	COURSE OUTCOMES On completion of the course, students will be able to
1. To develop skills to design and analyse simple linear and nonlinear data structures. 2. To gain programming skills to implement sorting and searching algorithms 3. To Strengthen the ability to identify and apply the suitable data structures for the given real world problem 4. To Gain knowledge in practical applications of data structures	1. Implement linear data structures such as arrays, stacks, queues, and their variations to perform basic operations like insertion, deletion, and traversal. 2. Apply linked lists (singly, doubly, and circular) for dynamic memory management and use them to implement linear data structures such as stacks and queues. 3. Use standard searching (linear and binary) and sorting techniques (selection, insertion, merge, quick, heap) to process and organize data efficiently. 4. Construct tree structures such as binary trees, binary search trees, and AVL trees, and apply recursive traversal techniques. 5. Apply graph traversal (DFS and BFS), hashing techniques, and minimum spanning tree algorithms (Prim's and Kruskal's) for solving computing problems.

CO-PO and CO-PSO mapping

CO/PO &PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	3	3	2	2	-	1	2	1	1	2	3	2	2
CO2	3	3	3	2	2	-	1	2	1	1	2	3	2	2
CO3	3	3	3	2	2	-	1	2	1	1	2	3	2	2
CO4	3	3	3	2	2	-	1	2	1	1	2	3	3	3
CO5	3	3	3	2	2	-	1	2	1	1	2	3	3	3

List of Experiments:

1. Write a program using arrays:
   a. To check whether the given matrix is sparse or not and display triplet representation.
   b. To add two given polynomials.

2. Write a program to implement the following linear data structures using arrays:
    a. Stacks
    b. Queues
    c. Circular Queue
3. Write a program to implement the following stack applications:
    a. String Reversal.
    b. Infix to Postfix Conversion.
    c. Postfix Expression Evaluation.
    d. Balanced Parenthesis
4. Write a program to implement insertion, deletion and search operation on Singly Linked List
5. Write a program to implement insertion, deletion and search operation on Doubly Linked List.
6. Write a program to implement insertion, deletion and search operation on Circular Linked List.
7. Write a program to implement the following using singly linked list:
    a. Stacks
    b. Queues
8. Write a program to implement the following searching techniques using arrays:
    a. Linear search
    b. Binary search
9. Write a program to build a hash table using linear probing and search for a given element.
10. Write a program to implement inorder, preorder, and postorder traversals in a Binary Tree.
11. Write a program to implement insertion, deletion and search operation in a Binary Search Tree.
12. Write a program to implement insertion, deletion and search operation in an AVL Tree.
13. Write a program to implement DFS and BFS graph traversal techniques.
14. Write a program to find the minimum cost spanning tree for a given connected graph using the following algorithms:
    a. Prims Algorithm
    b. Kruskal's Algorithm.
15. Write a program to implement the following sorting techniques:
    a. Selection sort
    b. Insertion sort
    c. Merge sort
    d. Quick sort
    e. Heap sort

For R25 Regulation

MUFFAKHAM JAH COLLEGE OF ENGINEERING & TECHNOLOGY

(SULTAN-UL-ULOOM EDUCATION SOCIETY)

ACCREDITED BY NBA and NAAC WITH 'A+' GRADE

(An Autonomous Institution)

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

OOP using Java Lab

SYLLABUS FOR B.E. III-SEMESTER

L: T: P (Hrs. / Week) 0: 0 : 2	**SEE Marks** : 60	**Course Code** : 25PC253CS
Credits : 2	**CIE Marks** : 40	**Duration of the SEE** : 03 Hours

S. No.	Course Objectives	Course Outcomes
1	Develop Java programs using core object-oriented principles.	Apply object-oriented programming concepts such as classes, inheritance, polymorphism, abstract classes, and interfaces in Java programs.
2	Handle exceptions and implement basic multithreaded applications.	Develop programs using exception handling and multithreading with proper synchronization mechanisms.
3	Perform file operations including reading, writing, and serialization.	Perform file handling operations using Java I/O streams, including serialization and deserialization.
4	Use collections effectively for data storage and manipulation.	Utilize Java Collection Framework components (List, Set, Map) and implement object comparison techniques.
5	Apply advanced Java features for efficient problem-solving.	Apply modern Java features such as lambda expressions, Stream API, and regular expressions for efficient data processing.

CO \ PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	2	3	-	3	-	-	-	-	-	2	3	2	1
CO2	3	3	3	1	3	-	-	-	-	-	2	3	3	2
CO3	2	2	3	-	3	-	-	-	-	-	1	2	2	2
CO4	2	3	3	-	3	-	-	-	-	-	2	1	2	3
CO5	2	2	3	-	3	-	-	-	1	-	3	2	3	3

List of Experiments:

1. Write a Java program to illustrate Command Line Arguments.
2. Write a Java program to illustrate java.util.Scanner class for input.

3. Write a Java program to illustrate the concept of classes with methods (Box class).
4. Write a Java program to illustrate the concept of constructor Overloading (Box class).
5. Write a Java program to illustrate the concept of method Overloading (Test class).
6. Write a Java Program that reads a line of integers, and then displays each integer, and the sum of all the integers Use java.util.StringTokenizer class
7. Write a Java program to illustrate the concept of Single level and Multi level Inheritance.
8. Write a Java program to illustrate the concept of Method Overriding (Figure, Triangle, Rectangle class). (demonstrate Runtime polymorphism – Dynamic method dispatch).
9. Write a Java program to illustrate the concept of Abstract Classes (Method Overriding).
10. Write a Java program to demonstrate the Interfaces.
11. Write a Java Program to implement Lambda Functions.
12. Write a Java Program to implement Regular Expressions.
13. Write a Java program to demonstrate checked and unchecked Exceptions.
14. Write a Java Program to demonstrate user defined Exceptions.
15. Write a Java program to illustrate collection classes like Array List, Linked List, using Iterator interface and List Iterator interface.
16. Write a Java program to illustrate the concept of Hashmap (key-value).
17. Write a Java Program to implement Stream API.
18. Write a Java program to illustrate the concept of threading using Thread Class and runnable Interface.
19. Write a Java program to illustrate the concept of Thread synchronization.
20. Write a Java program that reads a file name from the user, and then displays information about whether the file exists, whether the file is readable, whether the file is writable, the type of file and the length of the file in bytes.
21. Write a Java program to illustrate the concept of I/O Streams (FileInputStream, FileReader, FileOutputStream, FileWriter).
22. A program to demonstrate the usage of Filter Stream (DataInputStream and DataOutputStream) and Buffered I/O streams.
23. Write a Java program to implement Serialization / De-serialization concept.
24. Write a Java program to illustrate CRUD Operations on a database using JDBC.
25. Write a Java program to illustrate PreparedStatement with ResultSet.

For R25 Regulation

MUFFAKHAM JAH COLLEGE OF ENGINEERING & TECHNOLOGY

(SULTAN-UL-ULOOM EDUCATION SOCIETY)

ACCREDITED BY NBA and NAAC WITH 'A+' GRADE

(An Autonomous Institution)

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

SKILL DEVELOPMENT COURSE: DRONE TECHNOLOGY

SYLLABUS FOR B.E. III-SEMESTER

L:T:P (Hrs./week):0:0:2	SEE Marks : -	Course Code :25ES254EE
Credits : 1	CIE Marks : 50	Duration of SEE : -

COURSE OBJECTIVES	COURSE OUTCOMES
1) Understand the fundamentals of drone technology, flight principles, aerodynamics, and DGCA safety regulations. 2) Develop practical skills in configuring, calibrating, and maintaining drone hardware and onboard systems. 3) Operate drones using manual and autonomous flight modes through mission planning and telemetry systems. 4) Analyze drone performance using calibration, thrust measurement, flight stability, and communication experiments. 5) Apply drone technology for aerial imaging, inspection and payload-based applications while following SOPs	CO1: Identify and explain the functions of major drone components and subsystems. CO2: Configure and calibrate drone hardware, including ESCs, flight controllers, batteries, and sensors. CO3: Perform safe manual and autonomous drone flight operations. CO4: Analyze drone performance through thrust measurement, PID tuning, and flight data analysis. CO5: Demonstrate the use of drones for aerial imaging, inspection, and payload delivery applications.

CO-PO and CO-PSO MAPPING

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	2	-	-	2	-	-	-	-	-	1	3	2	1
CO2	3	3	2	2	3	1	1	2	-	-	1	3	3	2
CO3	3	3	3	2	3	2	1	2	1	-	2	3	3	2
CO4	3	3	2	3	2	1	-	-	1	-	2	3	2	3
CO5	3	3	3	3	3	2	1	2	2	1	3	2	3	3

1) Fundamentals of Flight and DGCA Regulations
2) Identification and Analysis of Drone Components
3) Electronic Speed Controller (ESC) Calibration and Motor Testing
4) Li-Po Battery Management and Safety
5) Flight Controller Configuration and Setup

6) IMU and Compass Calibration for Drone Flight Operations
7) Autonomous Waypoint Mission Planning and Execution
8) Payload Integration and Release Mechanism Testing
9) PID Tuning and Flight Stability Analysis
10) Aerial Imaging and Drone-Based Inspection
11) Propeller Thrust and Performance Evaluation
12) Assembling of mechanical and Electrical Subsystems of Drone

Outcome: Students will assemble a drone and demonstrate a drone-based solution for a real-world application by integrating drone hardware, planning, telemetry, flight control and suitable payloads.

Reference Books

1. **Terry Kilby and Belinda Kilby**, *Getting Started with Drones*, First Edition, Maker Media, Inc., San Francisco, USA, 2015.
2. **Garvit Pandya**, *Basics of Unmanned Aerial Vehicles: Time to Start Working on Drone*, First Edition, BPB Publications, 2020.
3. **John Baichtal**, *Building Your Own Drones: A Beginners' Guide to Drones, UAVs, and ROVs*, Que Publishing, First Edition, ISBN: 978-0789755988, 2016.

Reference Manuals / Online Resources

1. **DGCA**, *Drone Rules, 2021* and latest Civil Aviation Requirements (CAR) for Remotely Piloted Aircraft Systems (RPAS).
2. **ArduPilot Documentation** – https://ardupilot.org/
3. **PX4 User Guide** – https://docs.px4.io/
4. **Mission Planner Documentation** – https://ardupilot.org/planner/

MUFFAKHAM JAH COLLEGE OF ENGINEERING AND TECHNOLOGY

(AUTONOMOUS)

ACCREDITED BY NBA & NAAC WITH 'A+' GRADE

Banjara Hills, Hyderabad-34

Approved by A.I.C.T.E., New Delhi and

Affiliated to Osmania University, Hyderabad.

B.E (CSE) SEMESTER-IV

SEMESTER-IV			Scheme of Instruction				Scheme of Examination			Credits
S. No	Course Code	Name of the Course	Lecture	Tutorial	Practical	Contact Hrs/Wk	CIE	SEE	Duration in Hrs	
1	25BS211MT	Probability & Statistics	3	-	-	3	40	60	3	3
2	25PC222CS	Software Engineering	3	-	-	3	40	60	3	3
3	25PC223CS	Operating Systems	3	-	-	3	40	60	3	3
4	25PC224CS	Database Management Systems	3	1	-	4	40	60	3	4
5	25PC225CS	Web Technologies	3	-	-	3	40	60	3	3
Practical / Laboratory Courses										
6	25PC255CS	Operating Systems Lab	-	-	2	2	40	60	3	1
7	25PC256CS	Database Management Systems Lab	-	-	2	2	40	60	3	1
8	25PC257CS	Web Technologies Lab	-	-	2	2	40	60	3	1
9	25PW258CS	Mini Project			6	6	40	60	3	3
		Total	15	1	12	28	360	540	27	22

For R25 Regulation

MUFFAKHAM JAH COLLEGE OF ENGINEERING & TECHNOLOGY

(SULTAN-UL-ULOOM EDUCATION SOCIETY)

ACCREDITED BY NBA and NAAC WITH 'A+' GRADE

(An Autonomous Institution)

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

PROBABILITY AND STATISTICS

SYLLABUS FOR B.E. IV-SEMESTER

L:T:P (Hrs./week):3:0:0	SEE Marks : 60	Course Code : 25BS211MT
Credits : 3	CIE Marks : 40	Duration of SEE : 3 Hours

Course Objectives: To learn	**Course Outcomes**: After learning the contents of this course, the student must be able to
1. Understand fundamental concepts of probability theory and random variables.	1. Solve problems involving probability axioms, conditional probability and Bayes' theorem
2. Apply discrete and continuous probability distributions in engineering problems.	2. Evaluate statistical parameters of discrete probability distribution.
3. Analyze statistical data using sampling distributions and estimation techniques.	3. Evaluate statistical parameters of continuous probability distribution.
4. Perform hypothesis testing for means, proportions and variances.	4. Perform regression analysis to compute the coefficient of correlation to interpret data.
5. Apply correlation, regression and basic statistical tools in real-life engineering applications.	5. Testing of hypothesis of few unknown statistical parameters using types of sampling, Sampling distribution of means, Sampling distribution of variance, Estimations of statistical parameters.

CO/PO &PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1:	3	3	–	2	–	–	–	–	–	–	–	3	2	–
CO2:	3	3	–	2	1	–	–	–	–	–	–	3	2	–
CO3:	3	3	–	2	1	–	–	–	–	–	–	3	2	–
CO4:	3	3	2	3	2	–	–	–	–	1	–	3	3	1
CO5:	3	3	2	3	2	–	–	–	–	1	–	3	3	1

Unit-I:

Probability: Introduction, Conditional Probability, Baye's Theorem (Without proof), Random variable, Discrete Probability Distribution, Continuous Probability Distribution, Probability Mass Function, Probability Density Function, Expectation, Variance.

Unit-II:

Discrete Probability Distributions: Binomial Distribution and Poisson Distributions- Evaluation of Statistical Parameters for these Distributions- Mean, Variance and Moment Generating Function.

Unit-III:
Continuous Probability Distributions: Uniform Distribution, Exponential Distribution and Normal Distribution- Evaluation of Statistical Parameters for these Distributions-Mean, Variance and Moment Generating Function.

Unit-IV:
Curve Fitting by the Method of Least Squares-Fitting of Straight Line, Fitting of Parabola, Fitting of other Curves ($y = ae^{bx}, y = ax^{b}, y = ab^{x}$), Coefficient of Correlation, Regression Coefficient, Lines of Regression, Rank Correlation Coefficient.

Unit-V:
Test of Significance -Large samples- Test for single mean, difference of means, single proportion, difference of proportions -Test of significance for small samples- chi-square test for goodness of fit and independence of attributes, student's t-test, F -test for ratio of variances,

Suggested Reading:

1. R. K Jain S.R.K Iyengar, Advanced Engineering Mathematics, Narosa Publication. 4^{Th} Edition, 2014.
2. B. S. Grewal, Higher Engineering Mathematics, Khanna Publication 43^{rd} Edition, 2014.
3. S.C Gupta and V.K Kapoor, Fundamental of Mathematical Statistics, Sultan Chand & sons, New Delhi, 2014.

For R25 Regulation

MUFFAKHAM JAH COLLEGE OF ENGINEERING & TECHNOLOGY

(SULTAN-UL-ULOOM EDUCATION SOCIETY)

ACCREDITED BY NBA and NAAC WITH 'A+' GRADE

(An Autonomous Institution)

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

SOFTWARE ENGINEERING

SYLLABUS FOR B.E. IV-SEMESTER

L: T: P (Hrs. / Week) 3: 0 : 0	SEE Marks : 60	Course Code : 25PC222CS
Credits : 3	CIE Marks : 40	Duration of the Exam : 03 Hours

S. No.	Course Objectives	Course Outcomes
1	Understand the principles, processes, and methodologies of Software Engineering.	**CO1**: Explain software engineering principles, software processes, and process models.
2	Apply software process models and requirements engineering techniques in software development.	**CO2**: Analyze software requirements and prepare Software Requirements Specification (SRS) documents.
3	Design software systems using analysis and design models, architectural patterns, and user interface principles.	**CO3**: Develop analysis and design models using appropriate software design methodologies.
4	Apply software testing, quality assurance, and software metrics for quality software development.	**CO4**: Apply software testing strategies, quality assurance techniques, and software metrics.
5	Understand project management, risk management, and software process improvement techniques.	**CO5**: Estimate project effort, manage software risks, and evaluate software process maturity.

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	2	1	1	2	-	-	-	-	1	-	3	2	1
CO2	3	3	2	2	2	-	-	-	1	1	-	3	3	1
CO3	3	3	3	2	3	-	-	-	1	1	-	3	3	2
CO4	3	3	2	2	3	-	-	-	1	1	-	3	3	2
CO5	3	3	2	2	2	-	-	-	2	1	3	2	3	3

UNIT-I

Introduction to Software Engineering and Process Models:

Software Engineering Fundamentals: Nature of Software, Software Engineering as a Layered Technology, Software Process Framework, Umbrella Activities, Software Myths.

Process Models: Generic Process Model, Waterfall Model, Incremental Model, RAD Model, Evolutionary Models, Prototyping, Spiral Model, Unified Process, Personal Software Process (PSP), Team Software Process (TSP).

Agile Development: Agility and Agile Process, Agile Manifesto, Scrum, Extreme Programming (XP), Agile Modeling.

UNIT-II

Requirements Engineering:

Requirements Engineering, Functional and Non-functional Requirements, Requirements Engineering Process, Feasibility Study, Requirements Elicitation Techniques, Requirements Analysis.

Requirements Modeling: Use Case Modeling, Scenario-Based Modeling, Data Modeling Concepts, Flow-Oriented Modeling, Class-Based Modeling, Behavioral Modeling.

Software Requirements Specification: Characteristics of Good SRS, IEEE SRS Format, Requirements Validation, Requirements Management.

UNIT-III
Software Design Engineering:
Design Concepts: Design Process, Design Quality, Abstraction, Refinement, Modularity, Functional Independence, Information Hiding, Design Patterns.
Architectural Design: Software Architecture, Architectural Styles and Patterns, Data Design, Architectural Design Process.
User Interface Design: Golden Rules of User Interface Design, User Interface Analysis, Interface Design Process, Design Evaluation.
Component-Level Design: Designing Class-Based Components, Component-Level Design Principles, Design Guidelines, Refactoring

UNIT-IV
Software Testing and Quality Assurance
Software Testing Fundamentals: Verification and Validation, Test Planning, Test Case Design.
Testing Strategies: Unit Testing, Integration Testing, Validation Testing, System Testing, Regression Testing.
Testing Techniques: Black Box Testing, White Box Testing, Basis Path Testing, Control Structure Testing, Object-Oriented Testing.
Debugging: Debugging Process, Debugging Strategies, The Art of Debugging.
Software Quality Assurance: SQA Framework, Formal Technical Reviews, Software Reliability, ISO 9000 Standards, The SQA Plan.

UNIT-V
Software Risk Management, Metrics and Process Improvement
Risk Management: Software Risks, Risk Identification, Risk Projection, Risk Mitigation, RMMM Plan.
Software Metrics: Product Metrics, Metrics for Requirements, Design, Testing and Maintenance.
Software Process Improvement: SPI Framework, Process Assessment, CMMI Levels.
Case Study: Case Study – ATM System

Recommended Text Books
Primary Text

1. Roger S. Pressman and Bruce R. Maxim, *Software Engineering: A Practitioner's Approach*, 8th Edition, McGraw-Hill, 2015.
2. Ian Sommerville, *Software Engineering*, 10th Edition, Pearson Education.

References

1. Pankaj Jalote, *An Integrated Approach to Software Engineering*, 3rd Edition.
2. Rajib Mall, "Fundamentals of Software Engineering", 5th Edition, PHI, 2018.
3. Shari Lawrence Pfleeger, *Software Engineering: Theory and Practice*, 4Th edition, Pearson, 2011.
4. Ali Behforooz and Frederick J. Hudson, *Software Engineering Fundamentals*, Oxford University Press.

E-Learning Resources

1. NPTEL – Software Engineering by Prof. Rajib Mall (IIT Kharagpur) https://nptel.ac.in/courses/106105182
2. MIT OpenCourseWare – Software Engineering https://ocw.mit.edu/courses/16-355j-software-engineering-concepts-fall-2005/
3. SWAYAM Software Engineering Courses https://onlinecourses.swayam2.ac.in/e-learning/preview/cec26_cs06

For R25 Regulation

MUFFAKHAM JAH COLLEGE OF ENGINEERING & TECHNOLOGY

(SULTAN-UL-ULOOM EDUCATION SOCIETY)

ACCREDITED BY NBA and NAAC WITH 'A+' GRADE

(An Autonomous Institution)

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

OPERATING SYSTEMS

SYLLABUS FOR B.E. IV-SEMESTER

L: T: P (Hrs. / Week) 3: 0 : 0	SEE Marks : 60	Course Code : 25PC223CS
Credits : 3	CIE Marks : 40	Duration of the Exam : 03 Hours

S. No.	Course Objectives	Course Outcomes
1	To learn the fundamentals of Operating Systems.	Describe the evolution, types, and structure of operating systems including system calls and Virtual Machine.
2	To learn the mechanisms of OS to handle processes and threads and their communication.	Identify and explain the lifecycle of processes and threads along with scheduling techniques
3	To learn the mechanisms involved in memory management in contemporary OS.	Apply various synchronization mechanisms to resolve concurrency problems in operating systems and Deadlock conditions
4	To gain knowledge on distributed operating system concepts that includes architecture, Mutual exclusion algorithms, deadlock detection.	Demonstrate different memory management schemes including paging and virtual memory concepts.
5	To understand the concepts of file management, directory structures, storage allocation methods, and secondary storage organization.	Analyze and compare file management techniques, disk scheduling algorithms, and secondary storage structures, and evaluate their impact on system efficiency and performance.

CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	2	1	1	2	-	-	-	-	-	-	3	2	1
CO2	3	3	2	2	2	-	-	-	-	-	-	3	2	2
CO3	3	3	2	2	2	-	-	-	-	-	-	3	3	2
CO4	3	3	2	2	2	-	-	-	-	-	-	3	3	2
CO5	3	3	2	2	3	-	-	-	-	-	-	3	3	3

UNIT-I

Introduction: Concept of Operating Systems, Computer-System Organization & Architecture, Dual Mode of operation, Generations of Operating systems, Types of Operating Systems, OS Services, System Calls, Structure of an OS - Layered, Monolithic, Microkernel Operating Systems, Concept of Virtual Machine.

UNIT-II

Processes: Definition, Process Relationship, Process State transitions, Process Control Block (PCB), Operations on Processes, Context switching.

Thread: Definition, Various states, Benefits of threads, Types of threads, Concept of multithreads

Process Scheduling: Foundation and Scheduling objectives, Types of Schedulers, Scheduling Criteria, Scheduling algorithms, multiprocessor scheduling.

UNIT-III

Process Synchronization: Inter-process Communication: Pipes, Message Passing and Shared memory, Concept of Synchronization: Critical Section, Race Conditions, Peterson's Solution, Semaphores, Classical problems of synchronization: The Producer/Consumer problem, Reader's writer problem, Dining philosopher's problem, Monitors.

Deadlocks: Definition, Necessary and sufficient conditions for Deadlock, Methods for Handling Deadlocks: Deadlock prevention, Deadlock Avoidance: Banker's algorithm, Deadlock detection and Recovery.

UNIT-IV

Memory Management: Basic concept, Logical and Physical Address Space, Swapping. Memory allocation: Contiguous Memory Allocation, fragmentation, and Compaction; Paging, Structure of Page table, Segmentation.

Virtual Memory: Basics of Virtual Memory, locality of reference, Page fault, Demand paging, Page Replacement algorithms, Trashing.

UNIT-V

File Management: Concept of File-Attributes, Operations, File types, Access methods, Directory structure, File System structure, Allocation methods, Free-space management, directory implementation, Efficiency and performance.

Secondary-Storage Structure: Disk structure, Disk scheduling algorithms, Disk Management, RAID Structure

Recommended Text Books

Primary Text:

1. Avi Silberschatz, Peter Galvin, Greg Gagne, Operating System Concepts Essentials, 9th Edition, Wiley Asia Student Edition, 2017.
2. William Stallings, Operating Systems: Internals and Design Principles, 5th Edition, Prentice Hall of India, 2016.

References:

1. A.Tanenbaum – Modern Operation Systems. Third edition,PearsonEducation,2008.
2. Maurice Bach, Design of the Unix Operating Systems, 8th Edition, Prentice-Hall of India, 2009.
3. Daniel P. Bovet, Marco Cesati, Understanding the Linux Kernel, 3rd Edition,, O'Reilly and Associates.

E-learning resources:

1. https://nptel.ac.in/courses/106105214
2. https://ocw.mit.edu/courses/6-1810-operating-system-engineering-fall-2023/

For R25 Regulation

MUFFAKHAM JAH COLLEGE OF ENGINEERING & TECHNOLOGY

(SULTAN-UL-ULOOM EDUCATION SOCIETY)

ACCREDITED BY NBA and NAAC WITH 'A+' GRADE

(An Autonomous Institution)

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

DATABASE MANAGEMENT SYSTEMS

SYLLABUS FOR B.E. IV-SEMESTER

L:T:P (Hrs./week):3:1:0	SEE Marks : 60	Course Code :25PC224CS
Credits : 4	CIE Marks : 40	Duration of SEE : 3 Hours

Course Objectives:	**Course Outcomes:**
Comprehend the fundamental concepts of database systems and design efficient databases using Entity-Relationship (ER) modelling, including constraints, specialization, generalization, and aggregation.	**CO1**: Describe the fundamental concepts of database systems and analyze database requirements to develop conceptual database designs using the Entity-Relationship (ER) model
Recognize SQL and relational algebra concepts to retrieve, manipulate, and manage data while ensuring database integrity and security.	**CO2**: Apply SQL and relational algebra to query, manipulate, and manage relational databases using joins, views, transactions, integrity constraints, authorization, and relational operations.
Apply SQL queries and relational algebra for database creation, retrieval, and manipulation.	**CO3**: Normalize database schemas to improve efficiency and eliminate anomalies in database design.
Analyze database normalization techniques to reduce redundancy and eliminate anomalies.	**CO4**: Analyze indexing concepts, including balancing trees, and demonstrate the use of transaction management.
Apply the concepts of transaction management, indexing, concurrency control, and recovery mechanisms in DBMS.	**CO5**: Demonstrate concurrency control techniques to improve database performance and maintain consistency.

CO/ PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PS O1	PS O2	PS O3
CO1	3	2	3	1	2	-	-	-	-	-	2	3	2	2
CO2	3	3	2	1	3	-	-	-	-	-	2	2	3	2
CO3	3	3	3	-	2	-	-	-	-	-	2	2	3	2
CO4	3	3	2	1	3	-	-	-	-	-	2	3	3	2
CO5	3	3	2	2	3	1	-	-	-	1	2	2	3	3

Unit-I

Introduction: Database System Application, Purpose of Database Systems, View of Data, Database Languages, Data Storage and Querying, Database Architecture, Database Users and Administrators.

Database Design and the E-R Model: Overview of the Design Process, Entity Relationship Model, Constraints, Entity-Relationship Diagrams, Specialization, Generalization, Aggregation.

Unit-II

Introduction to SQL: Basic structure of SQL Queries, Additional Basic Operations, Set Operations, Null Values, Aggregate Functions, Nested Subqueries.

Intermediate SQL: Joins Expressions, Views, Transactions, Integrity Constraints, Authorization.

Relational Algebra: Fundamental Operations, Additional Relational Algebra Operations, Extended Relational Algebra Operations

Unit III

Advance SQL: Cursors, Functions, Procedures, Packages, Triggers, Exception Handling.

Normalization: Database Anomalies, First Normal Form, Second Normal Form, Third Normal form and Boyce Codd's Normal Form.

Unit IV

Indexing: Basic Concepts, Ordered indices, B+ Trees, B Trees, Bitmap Indices.

Transactions: ACID Properties, Transaction States, Storage Structure, Conflict Serializability, View Serializability, Recoverability.

Unit V

Concurrency Control: Lock Based Protocols, Deadlock Handling, Multiple Granularity, Timestamp Based Protocols, Validation Based Protocols

TEXT BOOKS:

1. Abraham Silberchatz, Henry F Korth and Sudarshan S, "Database System Concepts", TataMcGraw- Hill, New Delhi,2010.
2. RamezElmasri and Shamkant B Navathe, "Fundamentals of Database Systems",Addison Wesley,USA,2010.
3. Raghu Ramakrishnan and Johannes Gehrke, "Database Management Systems", Tata McGraw-Hill, New Delhi,2008.
4. Gupta G K, "Database Management System", Tata McGraw-Hill, New Delhi,2011.
5. AtulKahate, "Introduction to Database Management Systems", Pearson Education, New Delhi,2009

For R25 Regulation

MUFFAKHAM JAH COLLEGE OF ENGINEERING & TECHNOLOGY

(SULTAN-UL-ULOOM EDUCATION SOCIETY)
ACCREDITED BY NBA and NAAC WITH 'A+' GRADE
(An Autonomous Institution)

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

WEB TECHNOLOGIES

SYLLABUS FOR B.E. IV-SEMESTER

L:T:P (Hrs./week):3:0:0	SEE Marks : 60	Course Code : 25PC225CS
Credits : 3	CIE Marks : 40	Duration of SEE : 3 Hours

COURSE OBJECTIVES: The course aims to	**COURSE OUTCOMES: On completion of the course, students will be able to**
1. Introduce HTML and CSS for building and styling web pages.	1. Create structured and presentable web pages using HTML and CSS.
2. Describe the working of Dynamic web pages using JavaScript.	2. Create dynamic web pages using JavaScript for event handling and validations.
3. Introduce the concept of executing JavaScript outside the web browser using Node.js	3. Demonstrate the use of JavaScript outside the web browser using Node.js
4. Describe the concept of NoSQL databases using MongoDB	4. Apply NoSQL concepts in web application development using Node.js
5. Develop a Web Application using React.js and Express.js	5. Develop an Interactive Web App using React.js and Express.js

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	1	2	-	3	-	-	-	-	1	1	3	2	1
CO2	3	2	3	1	3	-	-	-	-	1	2	3	3	1
CO3	3	2	2	1	3	-	-	-	-	1	2	2	3	2
CO4	3	3	3	2	3	-	-	-	-	1	2	2	3	2
CO5	3	3	3	1	3	-	-	-	2	1	3	3	3	3

UNIT-I

HTML5: Fundamentals of HTML, Working with Text, Organizing Text in HTML, Working with Links, Creating Tables, Working with Images, Working with Forms.

CSS3: Overview of CSS, Background and Colors, Fonts and Text Styles, Creating Boxes and Columns, Displaying, Positioning and Floating Elements, List Styles, Table Layouts.

UNIT-II

JavaScript: Overview of JavaScript, JavaScript Functions and Events, JavaScript Objects, Working with Browser Objects, Working with Document Object, Document Object Model, Validation, Errors, Exception Handling.

JSON: What is JSON, JSON Syntax, JSON Datatypes, JSON Schema.

UNIT-III

Node.js: What is Node.js, Getting Started with Node.js, Using Events, Listeners, Timers, Handling Data I/O, Accessing File System, Implementing HTTP Services, Implementing Socket Services, Using OS Module, Using Util Module.

UNIT-IV

MongoDB: What is MongoDB, Understanding NoSQL and MongoDB, Getting Started with MongoDB and Node.js, Manipulating MongoDB, Accessing MongoDB.

UNIT-V

Express.js: Getting Started with Express, Configuring Routes, Using Request Objects, Using Response Objects, Implementing Template Engine.

React.js: What is React, Understanding React Components and JSX, Components and Props, Working with Events and State, Rendering Lists and Conditional Content.

Suggested Textbooks:

1. "HTML5 Black Book", 2nd Ed., 2016, DT Editorial Services, Dreamtech Press (Unit-1 & 2)
2. "Introduction to JavaScript Object Notation", 1st Ed, 2015, Lindsay Bassett, O'Reilly. (Unit-2)
3. "Node.js, MongoDB and Angular Web Development: The Definitive Guide to Using the MEAN Stack to Build Web Applications", 2nd Ed., 2017, Brad Dayley, Brendan Dayley, Calen Dayley, Addison-Wesley Professional. (Unit-3, Unit-4 & Unit-5)
4. "React Key Concepts: An in-depth guide to React's core features", 2nd Edition, 2025, Maximilian Schwarzmuller, Packt Publishing (Unit-5)

Reference Books:

1. "Pro HTML5 with CSS, JavaScript and Multimedia" 2017, Mark J. Collins, Apress.
2. "Web Programming with HTML5, CSS and JavaScript", 2019, John Dean, Jones & Bartlett Learning
3. "Learning React: Modern Patterns for Developing React Apps", 2nd Ed. 2020, Alex Banks & Eve Porcello, O'Reilly
4. "Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React and Node", 2017, Vasan Subramanian, Apress.

For R25 Regulation

MUFFAKHAM JAH COLLEGE OF ENGINEERING & TECHNOLOGY

(SULTAN-UL-ULOOM EDUCATION SOCIETY)

ACCREDITED BY NBA and NAAC WITH 'A+' GRADE

(An Autonomous Institution)

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

OPERATING SYSTEMS LAB

SYLLABUS FOR B.E. IV-SEMESTER

L: T: P (Hrs. / Week) 0: 0 : 2	SEE Marks : 60	Course Code : 25PC255CS
Credits : 1	CIE Marks : 40	Duration of the Exam : 03 Hours

S. No.	Course Objectives	Course Outcomes
1	To enable students to apply UNIX system calls for file management and process control using C.	Apply UNIX system calls for file management and process control in C programs.
2	To develop the ability to demonstrate process creation, synchronization, and inter-process communication using system calls and IPC mechanisms.	Demonstrate process creation, synchronization, and communication using system calls and IPC mechanisms.
3	To provide knowledge of multithreading concepts and develop applications using POSIX threads with proper synchronization.	Develop multithreaded applications using POSIX threads and implement synchronization techniques.
4	To equip students with the ability to analyze and simulate CPU scheduling and memory management algorithms.	Analyze and simulate CPU scheduling and memory management algorithms.
5	To enable students to implement disk scheduling algorithms and perform basic shell scripting for system-level tasks.	Implement disk scheduling algorithms and basic shell scripts for system-level tasks.

CO \ PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO1**	**PSO2**	**PSO3**
CO1	3	2	3	-	3	-	-	-	-	-	-	3	2	1
CO2	3	3	3	2	3	-	-	-	-	-	-	3	3	2
CO3	3	3	3	2	3	-	-	-	-	-	-	2	3	2
CO4	3	3	2	3	3	-	-	-	-	-	-	3	2	1
CO5	2	2	3	2	3	-	-	-	-	-	-	2	3	2

CASE STUDY

Perform a case study by installing and exploring various types of operating systems on a physical or logical (virtual) machine.

List of Experiments (preferred programming language is C)

1. Write C programs to demonstrate the usage of UNIX system calls for file management. (open, read, write, close, lseek, stat).
2. Write C programs to demonstrate process-related concepts using UNIX system calls (fork, exec, wait, getpid), including orphan and zombie processes.
3. Write C programs to demonstrate thread-related concepts using POSIX threads (pthreads).
4. Write C programs to simulate CPU scheduling algorithms: FCFS, SJF, and Round Robin.
5. Write C programs to simulate Intra & Inter-Process Communication (IPC) techniques: Pipes, Message Queues, and Shared Memory.
6. Write C programs to simulate solutions to Classical Process Synchronization Problems: Producer-Consumer, Readers-Writers, Dining Philosophers
7. Write C programs to simulate Page Replacement Algorithms: FIFO, LRU
8. Write C programs to simulate implementation of Disk Scheduling Algorithms: FCFS, SSTF
9. Write shell scripts to demonstrate basic shell programming concepts.

For R25 Regulation

MUFFAKHAM JAH COLLEGE OF ENGINEERING & TECHNOLOGY

(SULTAN-UL-ULOOM EDUCATION SOCIETY)

ACCREDITED BY NBA and NAAC WITH 'A+' GRADE

(An Autonomous Institution)

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

DATABASE MANAGEMENT SYSTEMS LAB

SYLLABUS FOR B.E. IV-SEMESTER

L:T:P (Hrs./week):0:0:2	SEE Marks : 60	Course Code : 25PC256CS
Credits : 1	CIE Marks : 40	Duration of SEE : 3 Hours

Course Objectives: The course will help to:	**Course Outcomes:**
1. To develop skills in SQL for database definition, manipulation, and query processing.	1. Apply DDL, DML, TCL, and DCL commands for database creation and management.
2. To learn advanced database concepts such as joins, subqueries, integrity constraints, and views.	2. Construct few complex SQL queries.
3. To gain practical knowledge of PL/SQL programming including procedures, functions, triggers, and exception handling	3. Design relational database tables with appropriate integrity constraints.
4. To understand transaction management, locking mechanisms, and real-time database applications through case studies	4. Develop PL/SQL programs to demonstrate the basic construct of programming.
5. Get acquainted with GUI development tools and design	5. Design a case studies for practical database use cases such as student information, library systems, and payroll management.

CO / PO / PSO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO1**	**PSO2**	**PSO3**
CO1:	3	2	2	–	3	–	–	–	–	–	–	3	3	1
CO2:	3	3	2	1	3	–	–	–	–	–	–	3	3	2
CO3:	3	3	3	2	2	–	–	–	–	–	1	3	3	2
CO4:	2	2	3	2	3	–	–	–	1	1	2	2	3	2
CO5:	2	3	3	2	3	1	–	–	2	2	2	2	3	3

List of Programs

1. Practice DDL Commands.
2. Practice DML Commands.
3. Practice Different Operators.
4. Practice Aggregate Functions and Sub Queries.
5. Practice Group by, Having and Order by Clauses.
6. Practice Various Types of Joins and Built in Functions.
7. Create tables with Complex Integrity Constrains.
8. Practice TCL and DCL Commands and Create Different Views

9. PL/SQL Programs on Control Structures, and Triggers.
10. PL/SQL Programs on Cursors and Exception Handling.
11. PL/SQL Programs on Functions, Procedures, Packages.
12. Any one Case Study (University or Hospital or Bank)

Suggested Readings:

1. Nilesh Shah, Database System Using Oracle, PHI, 2007.
2. Rick F Vander Lans, Introduction to SQL, Fourth edition, Pearson Education, 2007.
3. Benjamin Rosenzweig, Elena Silvestrova, Oracle PL/SQL by Example, Third edition, Pearson Education, 2004.
4. Ivan Bayross, Sql Pl/SQL the Programming language of Oracle , 2003
5. Albert Lulushi, Oracle Forms Developer's Handbook, Pearson Education, 2006.

For R25 Regulation

MUFFAKHAM JAH COLLEGE OF ENGINEERING & TECHNOLOGY

(SULTAN-UL-ULOOM EDUCATION SOCIETY)

ACCREDITED BY NBA and NAAC WITH 'A+' GRADE

(An Autonomous Institution)

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

WEB TECHNOLOGIES LAB

SYLLABUS FOR B.E. IV-SEMESTER

L:T:P (Hrs./week):0:0:2	SEE Marks : 60	Course Code : 25PC257CS
Credits : 1	CIE Marks : 40	Duration of SEE : 3 Hours

Course Objectives: The course will help to:	**Course Outcomes:**
1. Introduce HTML and CSS for building and styling web pages.	1. Develop structured and presentable web pages using HTML and CSS.
2. Describe the working of Dynamic web pages using JavaScript.	2. Create dynamic web pages using JavaScript for event handling and validations.
3. Introduce the concept of executing JavaScript outside the web browser using Node.js	3. Demonstrate the working of Node.js built-in modules.
4. Describe the concept of NoSQL databases using MongoDB	4. Demonstrate NoSQL concepts in web application development using Node.js
5. Develop a Web Application using React.js and Express.js.	5. Develop an Interactive Web App using React.js and Express.js

CO / PO / PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1:	3	1	2	–	3	–	–	–	–	1	1	3	2	1
CO2:	3	2	3	1	3	–	–	–	–	1	2	3	3	1
CO3:	3	2	2	1	3	–	–	–	–	1	2	2	3	2
CO4:	3	3	3	2	3	–	–	–	–	1	2	2	3	2
CO5:	3	3	3	2	3	1	–	–	2	2	3	3	3	3

List of Experiments:

1. Develop a static website using HTML Tables
2. Develop a static website using DIV and CSS
3. Develop a registration page using HTML forms
4. Develop a dynamic web page using JavaScript
5. Write a JavaScript program to validate the registration form
6. Create a dynamic web page using JavaScript's Event Handling and DOM manipulation.
7. Demonstrate working of built-in node.js modules and file system.
8. Demonstrate routes and parameter handling in Express.js
9. Write a program to implement MVC architecture

10. Demonstrate rendering HTML and JSX using React.js
11. Demonstrate the use of props, events, lists, forms using React.js
12. Create a Single Page Application (SPA) using REST service.
13. Write a Node.js program to create DB and Collections in MongoDB
14. Write a React.js program to retrieve data from MongoDB
15. Write a React.js program to insert, update and delete data in MongoDB

Suggested Textbooks:

1. "HTML5 Black Book", 2nd Ed., 2016, DT Editorial Services, Dreamtech Press
2. "Node.js, MongoDB and Angular Web Development", 2nd Ed., 2017, Brad Dayley, Brendan Dayley, Calen Dayley, Addison-Wesley Professional.
3. "React Key Concepts: An in-depth guide to React's core features", 2nd Edition, 2025, Maximilian Schwarzmuller, Packt Publishing

For R25 Regulation

MUFFAKHAM JAH COLLEGE OF ENGINEERING & TECHNOLOGY

(SULTAN-UL-ULOOM EDUCATION SOCIETY)

ACCREDITED BY NBA and NAAC WITH 'A+' GRADE

(An Autonomous Institution)

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

MINI PROJECT

SYLLABUS FOR B.E. IV-SEMESTER

L:T:P (Hrs./week):0:0:6	SEE Marks : 60	Course Code : 25PW258CS
Credits : 3	CIE Marks : 40	Duration of SEE : 3 Hours

Course Objectives:	Course Outcomes: Student will be able to
1. To enhance practical and professional skills.	1. Demonstrate the ability to synthesize and apply the knowledge and skills acquired in the academic program to the real-world problems.
2. To familiarize tools and techniques of systematic literature survey and documentation.	2. Evaluate different solutions based on economic and technical feasibility.
3. To expose the students to industry practices and teamwork.	3. Effectively plan a project and confidently perform all aspects of project management.
4. To encourage students to work with innovative and entrepreneurial ideas.	4. Demonstrate effective coding, written, presentation and oral communication skills.

COs \ POs\PSO's	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO1**	**PSO2**	**PSO3**
CO1	3	3	2	2	3	–	–	–	1	–	1	3	3	2
CO2	2	3	3	2	2	1	1	–	1	–	2	3	2	2
CO3	1	2	3	2	2	1	–	1	3	2	3	2	3	3
CO4	1	–	2	–	3	–	–	1	2	3	2	2	3	2

I. CASE STUDY

The students are required to carry out mini projects in any of the areas such as Data Structures, Database Management Systems, Operating Systems, Software Engineering, OOP using Java, Python, Matlab and C programming.
Problems Statements are suggested to be taken from Smart India Hackathon (SIH) Portal invited from the Ministries / PSUs / MNCs / NGOs to be worked out through.

The project could be classified as hardware, software, modeling, simulation etc. The project should involve one or many elements of techniques such as analysis, design, and synthesis.

The department will appoint a project coordinator who will coordinate the following:
1. Grouping of students (maximum of 3 students in a group)
2. Allotment of projects and project guides.

3. All projects allotment is to be completed by the 4th week of the semester so that the students get sufficient time for completion of the project.
4. Disseminate guidelines given by monitoring committee comprising of senior faculty members to the students and their guides.

Sessional marks are to be awarded by the monitoring committee. Common norms will be established for the final presentation and documentation of the project report by the respective departments.

Students are required to submit a presentation and report on the mini project at the end of the semester.