Kaveri University
B.Tech. Computer Science & Engineering
Course Structure
Applicable for 2025-2026 Admitted Batch

II-Year

SEMESTER-III						
Sl. No.	**Course Code**	**Course**	**L**	**T**	**P**	**Credits**
1	25MA301BS	Mathematics–III (Numerical Methods)	3	1	0	4
2	25CS302PC	Computer Organization &Architecture	3	0	0	3
3	25EC303ES	Digital Logic Design	3	0	0	3
4	25CS304PC	Data Structures	3	0	0	3
5	25CS305PC	Object Oriented Programming using Java	3	0	0	3
6	25EC306ES	Digital Logic Design Laboratory	0	0	2	1
7	25CS307PC	Data Structures Laboratory	0	0	2	1
8	25CS308PC	Object Oriented Programming Laboratory using Java	0	0	2	1
9	25EN309HS	Soft Skills and Inter Personal Skills	0	0	2	1
10	25CS310ES	Drone Technology Workshop	1	0	2	2
		Total	**16**	**1**	**10**	**22**

SEMESTER-IV						
S. No.	**Course Code**	**Course**	**L**	**T**	**P**	**Credits**
1	25MA401BS	Mathematics–IV (Probability and Statistics)	3	1	0	4
2	25CS402PC	Design & Analysis of Algorithms	3	0	0	3
3	25CS403PC	Full Stack Web Development	3	0	0	3
4	25CS404PC	Operating Systems	3	0	0	3
5	25CS405PC	Data Base Management System	3	0	0	3
6	25CS406PC	Design & Analysis of Algorithms Laboratory	0	0	2	1
7	25CS407PC	Operating Systems Laboratory	0	0	2	1
8	25CS408PC	Data Base Management System Laboratory	0	0	2	1
9	25CS409PC	Full Stack Web Development Laboratory	0	0	2	1
10	25CS410EEC	Minor Project	0	0	6	3
11	25CH411AU	Environmental Science	3^	0	0	0
		Total	**15+3^**	**1**	**14**	**23**

^represents related to Audit Course.

NUMERICAL METHODS (MATHEMATICS III)

Sl. No.	Course Code	Course	L	T	P	Credits
1	25MA301BS	Numerical Methods (Mathematics III)	3	1	0	4

Course Objectives:

- Apply numerical techniques for solving algebraic and transcendental equations.
- Apply numerical techniques for solving systems of linear algebraic equations.
- Understand the concept of interpolation for estimating intermediate values from given data
- Interpret numerical differentiation and integration in solving physical problems.
- Apply numerical techniques for solving ordinary differential equations.

Course Outcomes:

- Solve algebraic and transcendental equations using numerical methods.
- Solve systems of linear equations and analyze eigenvalues and eigenvectors.
- Apply finite difference concepts and interpolation methods.
- Apply numerical methods for differentiation and integration in engineering problems.
- Solve ordinary differential equations using numerical techniques.

UNIT – 1 :
Numerical Solution of Transcendental Equations: Solution of nonlinear equations using Bisection method, False Position method, Fixed Point Iteration, and Newton-Raphson method.

UNIT – 2:
Linear Systems & Eigen Values: Gaussian elimination, Gauss-Jacobi, and Gauss-Seidel methods. Eigen values using the Power Method.

UNIT-3:
Interpolation : Newton's forward and backward difference methods, Newton's divided difference method, Lagrange interpolation, and inverse interpolation.

Unit-4:
Numerical Differentiation and Integration: Numerical differentiation using Newton's forward, backward, and divided difference methods. Numerical integration using Trapezoidal rule and Simpson's 1/3 and 3/8 rules.

Unit-5:
Numerical Solution of ODEs: Single-step methods: Taylor series method, Euler method, Improved Euler method, and fourth-order Runge-Kutta method. Multi-step methods: Milne's predictor-corrector method.

REFERENCE BOOKS:

1. M.K. Jain, S.R.K. Iyengar, and R.L. Jain, *Numerical Methods for Scientific and Engineering Computation*, Wiley Eastern Ltd., 4th Edition, 2003.
2. B.S. Grewal, *Numerical Methods in Engineering and Science*, Khanna Publishers, 42nd Edition, 2012.
3. S.S. Sastry, *Introductory Methods of Numerical Analysis*, Prentice Hall of India, 2003.
4. F.B. Hildebrand, *Introduction to Numerical Analysis*, 2nd Edition, Dover Publications, 2013.
5. Steven C. Chapra and Raymond P. Canale, *Numerical Methods for Engineers*, McGraw-Hill, 2004.

COMPUTER ORGANIZATION AND ARCHITECTURE

Sl. No.	Course Code	Course	L	T	P	Credits
2	25CS302PC	Computer Organization & Architecture	3	0	0	3

Course Objectives:

- Introduce principles of computer organization and the basic architectural concepts.
- Explore the basic organization, design, and programming of a simple digital computer.
- Introduces simple register transfer language to specify various computer operations.
- Describing memory organization and I/O systems.
- Introduce pipelining and vector processing.

Course Outcomes:

- Identity of computer organization architecture.
- Analyze the basics of instruction sets and their functionality.
- Evaluate arithmetical operations by using data.
- Demonstrate the functional units of the computer.
- Design a pipeline for consistent execution of instructions.

UNIT–I :

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

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

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.

UNIT–II :

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

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

UNIT–III :

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

Computer Arithmetic: Addition and subtraction, multiplication Algorithms, Division Algorithms, floating – point Arithmetic operations.

UNIT–IV :

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

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

UNIT–V :
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, Cache Coherence.

TEXT BOOKS:

1. M. Moris Mano, Computer System Architecture, *Third Edition, Pearson/PHI, 2016.*
2. William Stallings, Computer Organization and Architecture, *6th ed, Pearson/PHI, 2010.*

REFERENCE BOOKS:

1. Car Hamacher, ZvonksVranesic, SafeaZaky, Computer Organization, Vth Edition, McGraw Hill, 2002.
2. Andrew S.Tanenbaum, Structured Computer Organization, 4th Edition, PHI/Pearson, 2003.
3. B. Ram, Computer Fundamentals Architecture and Organization, 5th ed., New Age International Publications, 2000.

WEB LINKS:

1. https://nptel.ac.in/courses/106105163.
2. https://onlinecourses.nptel.ac.in/noc20_cs64/preview.
3. https://www.udemy.com/topic/computer-architecture/

DIGITAL LOGIC DESIGN

Sl. No.	Course Code	Course	L	T	P	Credits
3	25EC303ES	Digital Logic Design	3	0	0	3

Course Objectives:

- To understand common forms of number representation in logic circuits.
- To learn basic techniques for the design of digital circuits and fundamental concepts used in thedesign of digital systems.
- To understand the concepts of combinational logic circuits and sequential circuits.
- To understand the Realization of Logic Gates Using Diodes & Transistors.
- To understand the concept of FSM and ASM charts.

Course Outcomes:

- Acquire the knowledge on numerical information in different forms and Boolean algebra theorems.
- Define Postulates of Boolean algebra and to minimize combinational functions, and design the combinational circuits.
- Design and analyze sequential circuits for various cyclic functions.
- Characterize logic families and analyze them for the purpose of AC and DC Parameters.
- Understand the concept of FSM and ASM charts.

UNIT – I:

Number Systems: Binary, Octal, Decimal, Hexadecimal, Fixed-point and Floating-point Number Representations, Complements of Numbers: 1's and 2's Complement, Error Detection and Correction Codes: Parity Check, Hamming Code.

Boolean Algebra and Logic Gates: Axiomatic definitions, basic theorems and properties, Boolean Functions: Canonical and standard forms, Digital Logic Gates Overview.

UNIT - II:

Gate-Level Minimization Techniques: Karnaugh maps: 2, 3, and 4 variables, Sum-of-products (SOP) and product-of-sums (POS) simplification, don't care conditions, Implementation using NAND and NOR gates.

UNIT - III:

Combinational Logic Circuits: Analysis and design procedures, Binary adder-subtractor and BCD adder, magnitude comparator, decoders, encoders, multiplexers and de multiplexers.

UNIT - IV:

Sequential Logic Circuits: Gated latches, Flip-flops: Clocked S-R, D, T, JK, Master-Slave JK, Design of synchronous and asynchronous counters, Shift registers: types and applications.

UNIT - V:

Synchronous Sequential Logic Moore and Mealy state machines, State diagrams, state tables, and state reduction, Case studies: sequence detector.

Programmable Logic Devices: Memory devices - RAM, ROM, Programmable Logic Arrays (PLA), Programmable Array Logic (PAL)

TEXT BOOK:

1. M. Morris Mano, Michael D. Ciletti, *Digital Design with an Introduction to the Verilog HDL*, 6th Edition, Pearson Education/PHI, 2017.

REFERENCE BOOKS:

2. Ronald J. Tocci, Neal S. Widmer, Gregory L. Moss, *Digital Systems: Principles and Applications*, 10th Edition, Pearson Education.
3. Charles H. Roth Jr., Larry L. Kinney, *Fundamentals of Logic Design*, 6th Edition, Cengage Learning.

DATA STRUCTURES

Sl. No.	Course Code	Course	L	T	P	Credits
4	25CS304PC	Data Structures	3	0	0	3

Course Objectives:

- Exploringbasic data structures such as stacks and queues.
- Introduces a varietyof data structures such as hash tables.
- Discussion of search trees.
- Understand the sorting algorithms.
- Introduces pattern matching algorithms

Course Outcomes:

- Explain the basic concepts such as Abstract Data Types, Linear and Non-Linear Data structures.
- Discuss hashing and different collision resolve techniques.
- Design programs using a varietyof data structures including binarysearch trees, heaps trees and AVL-trees.
- Design programs on sorting and graphs.
- Apply different searching techniques on Non linear data structure.

UNIT–I :

Introduction to Data Structures, abstract data types, Linear list – singly linked list implementation, insertion, deletion and searching operations on linear list, Stacks-Operations, array and link edre presentations of stacks, stack applications, Queues-operations, array and linked representations.

UNIT–II :

Dictionaries: linear list representation, skip list representation, operations - insertion, deletion and searching.

Hash Table Representation: hash functions, collision resolution-separate chaining, open addressing-linear probing, quadratic probing, double hashing, rehashing, extendible hashing.

UNIT–III :

Search Trees: Binary Search Trees, Definition, Implementation, Operations- Searching, Insertion and Deletion, AVL Trees, Definition, Height of an AVL Tree, Operations–Insertion, Deletion and Searching, Red –Black, Splay Trees.

UNIT–IV :

Graphs: Graph Implementation Methods. Graph Traversal Methods.

Sorting: Heap Sort, External Sorting- Model for external sorting, Merge Sort.

UNIT–V :

Pattern Matching and Tries: Pattern matching Algorithms-Brute force, the Boyer –Moore algorithm, the Knuth-Morris-Pratt algorithm, Standard Tries, Compressed Tries, Suffix tries.

TEXTBOOKS:

1. Fundamentals of Data Structures in C, 2nd Edition, E. Horowitz, S. Sahni and Susan AndersonFreed, *UniversitiesPress*.
2. DataStructuresusingC–A.S.Tanenbaum,Y.Langsam,andM.J.Augenstein,*PHI/Pearson Education*.

REFERENCE BOOKS:

1. Data Structures: A Pseudocode Approach with C, 2nd Edition, R. F. Gilberg and B.A.Forouzan, Cengage Learning.

OBJECT ORIENTED PROGRAMMINGS USING JAVA

Sl. No.	Course Code	Course	L	T	P	Credits
5	25CS305PC	Object Oriented Programming using Java	3	0	0	3

Course Objectives:

The students will try to learn:

1. The Fundamental concepts of Object-oriented approach for solving real-time problems.
2. The basic and advanced constructs of Java programming for developing object oriented concepts.
3. The design concepts for developing user interface of real time applications.

Course Outcomes:

1. Understand OOP concepts to apply basic Java constructs.
2. Analyze different forms of inheritance and usage of Exception Handling
3. Understand the different kinds of file I/O and Multithreading in complex Java programs, and usage of Container classes
4. Contrast different GUI layouts and design GUI applications
5. Construct a full-fledged Java GUI application and Applet with database connectivity.

UNIT – I :

Java Basics History of Java, Java buzzwords, data types, variables, scope and life time of variables, arrays, operators, expressions, control statements, type conversion and casting, simple java program

Fundamentals of Object Oriented Programming: Object-Oriented Paradigm, Basic Concepts of Object Oriented Programming, Applications of OOP. Concepts of classes, objects, constructors, methods, access control, this keyword, garbage collection, overloading methods and constructors, parameter passing, recursion, static keyword, nested and inner classes, Strings, Object class.

UNIT – II :

Inheritance & Polymorphism: Introduction, Forms of Inheritance specialization, specification, construction, extension, limitation, combination, Member access rules, super keyword, polymorphism-method overriding, abstract classes, final keyword.

Interfaces and Packages: Introduction to Interfaces, differences between abstract classes and interfaces, multiple inheritance through interfaces, Creating and accessing a package, Understanding CLASSPATH, importing packages.

Exception handling Concepts of exception handling, exception hierarchy, built in exceptions, usage of try, catch, finally, throw, and throws, creating own exception sub classes.

UNIT – III :

Files: Introduction to I/O Streams: Byte Streams, Character Streams. File me /O. Multi-threading: Differences between multi-threading and multitasking, thread life cycle, creating threads, thread priorities, synchronizing threads, inter thread communication. Java.util package- Collection Interfaces: List, Map, Set. The Collection classes: Linked List, HashMap, TreeSet, StringTokenizer, Date, Random, Scanner.

UNIT-IV :
AWT: Class hierarchy, Component, Container, Panel, Window, Frame, Graphics.
AWT controls: Labels, Button, Scrollbar, Text Components, Checkbox, Checkbox Group, Choice, List, Panes ScrollPane, Dialog and MenuBar.
Event Handling: Events, Event sources, Event classes, Event Listeners, Delegation event model, handling mouse and keyboard events, Adapterclasses.

UNIT – V :
Layout Manager: Border, Grid, Flow, Card and Gridbag**.**
Applets Concepts of Applets, life cycle of an applet, creating applets, passing parameters toapplets.
JDBC Connectivity: JDBC Type 1 to 4 Drivers, connection establishment, Query Execution.

TEXT BOOKS:
1. Java- the complete reference, Seventh edition, Herbert Schildt, Tata McGraw Hill.
2. Database Programming with JDBC&JAVA, Second Edition, George Reese, O'Reilly Media.

REFERENCE BOOKS:
3. Thinking in Java Fourth Edition, Bruce Ecke.
4. Introduction to Java programming, Y. Daniel Liang, Pearson Education.
5. Understanding OOP with Java, updated edition, T. Budd, Pearson Education.

DIGITAL LOGIC DESIGN LABORATORY

Sl. No.	Course Code	Course	L	T	P	Credits
6	25EC306ES	Digital Logic Design Laboratory	0	0	2	1

Course Objectives

- To understand the operation of basic logic gates and Boolean algebra principles.
- To develop skills in designing and implementing combinational logic circuits using standard digital ICs.
- To provide practical knowledge of sequential circuits using flip-flops, registers, and counters.
- To analyze the behavior of digital circuits through experimental verification and timing diagrams.
- To enhance problem-solving skills in the realization of digital systems and state machines.

Course Objectives

- Apply Boolean algebra concepts for the realization of digital logic circuits.
- Implement combinational circuits using logic gates and digital building blocks.
- Analyze the operation of sequential circuits using flip-flops, registers, and counters.
- Apply digital design principles to realize specified logic functions and counting sequences.
- Analyze and verify the functionality of digital systems through experimental implementation and testing.

List of Experiment:

1. Realization of Logic circuit to generator's Compliment using Logic Gates.
2. Realization of given Boolean function using universal gates and minimizing the same. Compare the gate count before and after minimization.
3. Design and realize Full Adder circuit using gates/universal gates. Implement Full Subtractor using full adder.
4. Designing a 2 – bit Comparator using AND, OR and NOT gates. Realize 4 – bit Comparator using 2 – bit Comparators.
5. Realize 2:1 MUX using the given gates and Design 8:1 using 2:1 MUX.
6. Implement the given Boolean function using the given MUX (ex: code converters).
7. Realize a 2x4 Decoder using logic gates and implement 3x8 Decoder using 2x4 Decoder.
8. Implement the given Boolean function using given Decoders.
9. Convert De-multiplexer to Decoder and vice versa.
10. Verification of truth tables of flip-flops using different clocks (level triggering, positive and negative edge triggering) also converts the given flip-flop from one type to other.
11. Designing of Universal n-bit shift register using flip-flops and Multiplexers. Draw the timing diagram of the Shift Register.
12. Design a Synchronous binary counter using D-flip-flop/given flip-flop.
13. Design a asynchronous counter for the given sequence using given flip-flops.
14. Designing of MOD 8 Counter using JK flip-flops.
15. Designing of sequence detecting State Machine with minimal states using the given flip-flops.
16. Designing of Parity Bit(even/odd) generator using the given flip-flops.

Major Equipment required for Laboratories:

1. 5 V Fixed Regulated Power Supply/ 0-5V or more Regulated Power Supply.
2. 20 MHz Oscilloscope with Dual Channel.
3. Bread board and components/ Trainer Kit.
4. Multimeter.

Essential ICs :

- Logic gates: 7400, 7402, 7408, 7432, 7486
- Flip-flops: 7474, 7476
- MUX/DEMUX: 74151, 74153
- Decoder: 74138
- Counters: 7490, 7493

Passive Components :

- Resistors (220Ω for LEDs)
- LEDs
- Connecting wires
- Push buttons / switches
- Clock generator (or function generator)

DATA STRUCTURES LAB

Sl. No.	Course Code	Course	L	T	P	Credits
7	25CS307PC	Data Structures Laboratory	0	0	2	1

Course Objectives:

- It provides an understanding of linear data structures such as stacks and queues.
- It provides an understanding of non-linear data structures like tress and graphs.
- It provides an understanding of linear and binarysearch algorithms.
- It provides an understanding of sorting algorithms.
- It provides an understanding of searching algorithms.

Course Outcomes:

- After completion of this course, the students will be able to:
- Implement various linear data structures.
- Implement various non-linear data structures.
- Compare various searching and sorting algorithms.
- Ability to implement trees and graphs traversals.

List of Experiments

1. Writeaprogramthatusesfunctionstoperformthefollowingoperationsonsinglylinked list.:
i) Creation ii) Insertion iii) Deletion iv) Traversal

2. Writeaprogramthatusesfunctionstoperformthefollowingoperationsondoublylinked list.:
i) Creation ii) Insertion iii) Deletion iv) Traversal

3. Write a program that uses functions to perform the following operations on circular linked list.:
i) Creation ii) Insertion iii) Deletion iv) Traversal

4. Write a program that implement stack (its operations) using
i) Arrays ii) Pointers

5. Write a program that implement Queue (its operations) using
i) Arrays ii) Pointers
6. Write a program that implements the following sorting methods to sort a given list of integers in ascending order
i) Bubble sort ii) Selection sort iii) Insertion sort

7. Write a program that use both recursive and non-recursive functions to perform the following searching operations for a Key value in a given list of integers:
i) Linear search ii) Binary search

8. Write a program to implement the tree traversal methods.

9. Write a program to implement the graph traversal methods.

TEXTBOOKS:

1. Fundamentals of Data Structures in C, 2nd Edition, E. Horowitz, S. Sahni and Susan Anderson Freed, *UniversitiesPress*.
2. Data Structures using C – A. S. Tanenbaum, Y. Langsam, and M. J.Augenstein, *PHI/Pearson Education*.

REFERENCE:

1. Data Structures: A Pseudocode Approach with C, 2nd Edition, R. F. Gilberg and B. A. Forouzan, Cengage *Learning*.

OBJECT ORIENTED PROGRAMMING USING JAVA LABORATORY

Sl. No.	Course Code	Course	L	T	P	Credits
8	25CS308PC	Object Oriented Programming Laboratory using Java	0	0	2	1

Course Objective:

1. Explain the fundamentals of Java programming, program structure, data types, operators, and control statements.
2. Apply object-oriented programming concepts such as classes, objects, inheritance, polymorphism, abstraction, and encapsulation in Java programs.
3. Develop Java applications using arrays, strings, methods, packages, interfaces, and exception handling mechanisms.
4. Demonstrate the use of file handling, collections framework, and multithreading concepts in solving programming problems.
5. Analyze programming requirements and select appropriate object-oriented design approaches for developing Java-based solutions.

Course Outcomes:

At the end of the course, student will be able to:

1. Apply basic Java constructs and OOP to solve mathematical problems.
2. Apply Inheritance in Java programs and Analyze Exception Handling code
3. Implement File input/output and multithreading concepts in advanced Java programs.
4. Design different GUI applications using GUI layouts.
5. Apply Applet development and Database connectivity to build GUI applications

Week 1 & 2:

1. Write a program to find total, average of given two numbers by using function with command-line arguments, static data members.
2. Write a program to illustrate class and objects.
3. Write a program to illustrate method & constructor overloading.
4. Write a program to illustrate parameter passing using objects.
5. Write a program to illustrate Array Manipulation.

Week 3:

1. Write a program to illustrate different types of inheritances.
2. Write a java program to illustrate Method overriding.
3. Write a java program to demonstrate the concept of polymorphism (Dynamic Method Dispatch).
4. Write a program to demonstrate final keyword.

Week 4 & 5:

1. Write a program to illustrate the use of creation of packages.
2. Write a java program to handle the situation of exception handling using multiple catch blocks.
3. Write a program to implement the concept of User defined Exceptions.

Week 6 & 7:

1. Write a program to illustrate Multithreading and Multitasking.
2. Write a program to illustrate thread priorities.
3. Write a program to illustrate Synchronization

Week 8 & 9:

1. Write a program to implement String Tokenizer.
2. Write a program to read one line at a time, and write it to another file.

Week 10 & 11:

1. Write a program to illustrate Event Handling (keyboard, Mouse events)
2. Write a program to illustrate applet life cycle and parameter passing.

Week 12:

Write a program to develop a calculator application using AWT.

Week 13:

Write a program to illustrate JDBC.

SOFT SKILLS AND INTERPERSONAL SKILLS

Sl. No.	Course Code	Course	L	T	P	Credits
9	25EN309HS	Soft Skills And Interpersonal Skills	0	0	2	1

Course Objectives:

The learner will be able to:

- Know the importance of soft skills.
- Identify good leadership skills /qualities.
- Recognize the importance of interpersonal skills.
- Demonstrate the significance of confidence building.
- Define and differentiate between a report and a proposal.

Course Outcomes:

After the end of the course the learners will be able to:

- Develop soft skills, communication skills, leadership skills etc.
- Implement goal setting techniques to build a promising career.
- Design formal reports and proposals with appropriate formal expressions.
- Describe team dynamics and exchange ideas about the elements of positive teamwork.
- Create a healthy workplace environment by treating others with respect and dignity.

List of Experiments

1. Introduction to soft skills, Exercise on soft skills
2. Differentiating between verbal and nonverbal communication
3. Presentations
4. Simulating English in Business/Corporate scenario
5. Grooming leadership qualities
6. Solving strategic and crisis management
7. Exercise on problem solving and decision making
8. Building self-esteem through self-analysis
9. Team Building Activities
10. Exercise on goal setting and SWOC analysis
11. Practice on different types of report writing
12. Building networking and understanding professional relationships
13. Mock Interviews

REFERENCE BOOKS:

1. Soft skills for Everyone - Jeff Butterfield, CENAGE Learning
2. Soft skills for Interpersonal Communication - S.Balasubramaniam (ORIENT BLACKSWAN)
3. *Technical Communication: Principles and Practice* - Meenakshi Raman and Sangeeta Sharma

Online Sources

1. https://owl.purdue.edu/index.html

DRONE TECHNOLOGY WORKSHOP

Sl. No.	Course Code	Course	L	T	P	Credits
10	25CS310ES	Drone Technology Workshop	1	0	2	2

Course Objectives:

1. To introduce the regulatory frameworks, including Drone Rules 2021, and foundational aviation principles.
2. To provide comprehensive technical knowledge on the aerodynamics and subsystems of various UAVs, including fixed-wing, rotorcraft, and hybrid models.
3. To impart vital skills in risk assessment, flight safety management, and rigorous fleet maintenance protocols.
4. To develop preliminary flight proficiency, emergency response, and circuit flying techniques within a simulated environment.
5. To equip learners with practical, hands-on command of RPAS through extensive field exercises, auto-missions, and night flying.

Course Outcomes:

1. Understand and comply with aviation regulations, meteorology, and air traffic services as required for commercial drone operations.
2. Analyze UAV payloads, ground control stations, and technical data to ensure efficient operation and regular servicing.
3. Apply standard operating procedures for mission planning, drone assembly, and handling of flight emergencies.
4. Demonstrate accurate flight maneuvers, orientation awareness, and recovery strategies using a flight simulator.
5. Execute complex real-world multirotor missions, including level turns, figure-8s, and automated flight paths, concluding with a final practical assessment.

UNIT - 1:

Regulatory Framework & Aviation Basics Stakeholders & Their Laws (Basic) , Drone Rules 2021, Principles of Flight , Air Traffic Services , Need for Drone Pilots to know about ATS , Radio Telephony , Meteorology.

UNIT - 2:

UAV Systems & Technical Knowledge Fixed Wing UAV Operations & Aerodynamics, Rotorcraft UAV Operations &Aerodynamics , Hybrid UAV Operations & Aerodynamics , Ground Control Station , Drone Equipment Maintenance , Regular Maintenance Cycles & Servicing , Payloads , Drone Data and Analysis , Integration of Sub-sections/Modules.

UNIT - 3:

Safety, Maintenance & Mission Planning Risk Assessment and Safety Management , Role of Flight Safety in Drone Operations , Handling of Flight Emergencies , Mission Planning , Assembly & Maintenance , De-assembling of Drone , Abnormal / Emergency Procedures.

UNIT - 4:

Simulator Training (Practical) Introduction to Flight Simulator , Simulator Familiarization & Controls Check , Pre-Flight Checks, Take-off & Cruise , Approach, Go-Around, Landing & Post-Flight Checks , Cruise and Turns, Climbing and Climbing Turns , Descend and Descending Turns , Disorientation and Recovery , Circuit Flying - Rectangular/Square / Circle / Figure-8 , Abnormal / Emergency Procedures.

UNIT - 5:

Practical Flight Training (Hands-on UAV Flying, Practical) RPAS Familiarization & Safety Briefing , Introductory Flight: Control Sensitivity & Orientation Awareness , Take-off, Climbing, Descending & Maintaining Height , Basic Controls: Pitch, Roll and Yaw , Disorientation & Recovery Progress Check - Multirotor , Level Turns in Both Directions, Left and Right Square Circuits , Flying in Circles , Auto Mission & Flight

TEXT BOOK:

1. *Basics of Unmanned Aerial Vehicles* Garvit Pandya

REFERENCE BOOKS:

1. *Introduction to Unmanned Aircraft Systems* by R. Kurt Barnhart, John M. Robbins, and Eric Shappee.*Fundamentals of Drone Technology: Drones- The future of 21st century.* - Mr. I.V.S.Yeswanth, Dr.A.V.S.Sridhar Kumar

MATHEMATICS–IV (PROBABILITY AND STATISTICS)

Sl. No.	Course Code	Course	L	T	P	Credits
1	25MA401BS	Probability and Statistics (Mathematics–IV)	3	1	0	4

Course Objectives:

- Apply fundamental rules and theorems of probability and evaluate expectation and variance of random variables
- Understand and apply standard theoretical probability distributions.
- Formulate and test statistical hypotheses using appropriate methods.
- Analyze relationships between variables using correlation and regression techniques.
- Apply the method of least squares to fit curves to given data.

Course Outcomes:

- Apply concepts of probability, conditional probability, and random variables to solve problems.
- Analyze and model data using standard theoretical distributions such as Binomial, Poisson, and Normal distributions.
- Apply hypothesis testing techniques for large and small samples to draw statistical inferences.
- Analyze correlation and regression to interpret relationships between variables.
- Apply the method of least squares to fit curves and perform regression analysis.

UNIT-1:

Probability and One-dimensional random variable : Probability concepts, Types of Events, Axioms and theorems - Conditional probability, Baye's theorem – without proof- Applications of Baye's Theorem. Random variables – Discrete case and continuous case- Mathematical expectation, Variance – discrete case and continuous case.

UNIT-2:

Theoretical Distributions : Discrete distributions – Introduction- Mean and Variance of Binomial Distribution- Fitting a Binomial distribution- M.G.F of Binomial Distribution- Poisson Distribution- Mean and Variance of Poisson Distribution- Fitting a Poisson distribution- MGF of Poisson distribution, Continuous distributions – Introduction, Mean and Variance - Normal distribution.

UNIT-3:

Testing of Hypothesis : Sampling Distributions – Type I and Type II errors- large sample test-Test of significance for single proportion- Test of significance for difference of proportions- Test of significance for single mean- Test of significance for difference of means- Small sample tests- Student's t- test for single mean- t- test for the difference of means-Fisher's F-test- Test of significance for two sample variances- Chi -square test- for the goodness of fit.

UNIT-4:

Correlation and Regression : Correlation and its Properties- *Karl Pearson's coefficient of correlation* - Spearman's rank correlation coefficient for repeated and non-repeated ranks- Linear Regression lines and Properties- Relation between correlation and regression coefficient.

UNIT-5:

Curve fitting and Regression: Curve fitting by the method of least squares- fitting of straight lines, second degree parabolas and more general curves.

TEXT BOOKS:

1.S.C. Gupta and V.K. Kapoor, *Fundamentals of Mathematical Statistics*, Sultan Chand & Sons, 11th Edition, 2015.

1. Ramana B.V. Ramana, *Higher Engineering Mathematics*, Tata McGraw-Hill, New Delhi, 2010.
2. R.A. Johnson, *Miller & Freund's Probability and Statistics for Engineers*, 8th Edition, Prentice Hall India, 2011.
3. T. Veerarajan, *Probability and Statistics*, Tata McGraw-Hill, 2010.
4. J.L. Devore, *Probability and Statistics for Engineering and the Sciences*, 8th Edition, Cengage Learning, 2012.

DESIGN AND ANALYSIS OF ALGORITHMS

Sl. No.	Course Code	Course	L	T	P	Credits
2	25CS402PC	Design And Analysis Of Algorithms	3	0	0	3

Course Objectives:

- Develop proficiency in evaluating algorithms using asymptotic notations, including best average-, and worst-case time/space complexities, and solving related recurrence relations.
- Master various algorithmic strategies—divide-and-conquer, greedy, dynamic programming, backtracking, and branch-and-bound—identifying suitable use cases and demonstrating their application.
- Critically assess and contrast different algorithms in terms of efficiency, scalability, and correctness through rigorous analytical reasoning and empirical evaluation.
- Differentiate between tractable (polynomial-time) and intractable (super-polynomial or exponential-time) problems.
- Identify and classify problems as P, NP, NP-hard, or NP-complete, and assess theirrelationships through polynomial-time reductions and Cook's theorem.

Course Outcomes:

- Able to Apply space and time complexity analysis using asymptotic notations.
- Able to Design divide-and-conquer algorithms and critically assess their runtime and space trade-offs.
- Able to Device backtracking and dynamic programming solutions.
- Able to Apply greedy methods and graph traversal algorithm.
- Able to Analyses and Design branch-and-bound algorithms for NP-hard problems

UNIT - I:

Introduction: Algorithm, Pseudo code for expressing algorithms, Performance Analysis-Space complexity, Time complexity, Asymptotic Notation- Big oh notation, Omega notation, Theta notation and Little oh notation, Disjoint Sets- disjoint set operations, union and find operations.

Divide and conquer: General method, applications - Binary search, Quick sort, Merge sort, Stassen's matrix multiplication.

UNIT - II:

Graphs: Breadth First Search, Depth First Search, spanning trees, connected And bi-connected components.

Greedy method: General method, Applications- Optimal storage on Tapes, Job sequencing with deadlines, knapsack problem, Minimum cost spanning trees, Single source shortest path problem.

UNIT - III:

Dynamic Programming: General method, applications-Matrix chain multiplication, Optimal binary search trees, 0/1 knapsack problem, All pairs shortest path problem, Travelling sales person problem, Reliability design.

UNIT - IV:

Backtracking: General method, applications-n-queen problem, sum of subsets problem, graph colouring, Hamiltonian cycles.

Branch and Bound: General method, applications - Travelling sales person problem, 0/1 knapsack problem- LC Branch and Bound solution, FIFO Branch and Bound solution.

UNIT - V:

Lower Bound Theory: Comparison Trees, NP-Hard and NP-Complete problems: Basic concepts, non-deterministic algorithms, NP - Hard and NP Complete classes, Satisfiability problem, Clique Decision Problem (CDP), Node cover decision problem.

TEXT BOOKS:

1. Fundamentals of Computer Algorithms, Ellis Horowitz, SatrajSahni and Rajasekharan, Galgotia publications Pvt.Ltd.
2. Design and Analysis of algorithms, Aho, Ullman and Hopcroft, Pearsoneducation.

REFERENCES:

1. Introduction to Design and Analysis of Algorithms A strategic approach, R.C.T.Lee, S.S.Tseng, R.C.Chang and T.Tsai, McGrawHill.
2. Data structures and Algorithm Analysis in C++, Allen Weiss, Second edition, Pearsoneducation.

FULL STACK DEVELOPMENT

Sl. No.	Course Code	Course	L	T	P	Credits
3	25CS403PC	Full Stack Development	3	0	0	3

Course Objectives:

- To learn the core concepts of Full Stack Development.
- To understand the front-end programming like Data, Files, Http Requests and Responses usingnode.js.
- To know the storing and processing huge data and connects with NodeJS application usingMangoDB.
- To provide an in-depth study of Angular Services in Web Applications.
- To explore interactive user interfaces with react structure.

Course Outcomes:

- Develop Full stack components for developing web application.
- Apply packages of NodeJS to work with Data, Files, Http Requests and Responses.
- Contrast MongoDB database for storing and processing huge data and connect with NodeJS application.
- Design faster and effective single page applications using Express and Angular.
- Create interactive user interfaces with react components.

UNIT –I :

Introduction to Full Stack Development: Understanding the Basic Web Development Framework- User, Browser, Webserver, Backend Services, Full Stack Components - Node.js,MongoDB, Express, React, Angular. Java Script Fundamentals, NodeJS- Understanding Node.js, Installing Node.js, Working with Node Packages, creating a Node.js Application, Understanding the Node.js Event Model, Adding Work to the Event Queue, Implementing Callbacks.

UNIT –II :

Node.js: Working with JSON, Using the Buffer Module to Buffer Data, Using the Stream Module to Stream Data, Accessing the File System from Node.js- Opening, Closing, Writing, Reading Files and other File System Tasks. Implementing HTTP Services in Node.js- Processing URLs, Processing Query Strings and Form Parameters, Understanding Request, Response, and Server Objects, Implementing HTTP Clients and Servers in Node.js, Implementing HTTPS Servers and Clients. Using Additional Node.js Modules-Using the os Module, Using the util Module, Using the dns Module, Using the crypto Module.

UNIT – III :

MongoDB: Need of NoSQL, Understanding MongoDB, MongoDB Data Types, Planning Your Data Model, Building the MongoDB Environment, Administering User Accounts, Configuring Access Control, Administering Databases, Managing Collections, Adding the MongoDB Driver to Node.js, Connecting to MongoDB from Node.js, Understanding the Objects Used in the MongoDB Node.js Driver, Accessing and Manipulating Databases, Accessing and Manipulating Collections.

UNIT –IV :

Express and Angular: Getting Started with Express, Configuring Routes, Using Requests Objects, Using Response Objects. Angular: importance of Angular, Understanding Angular, creating a Basic Angular Application, Angular Components, Expressions, Data Binding, Built-in Directives, Custom Directives, Implementing Angular Services in Web Applications.

UNIT – V:

React: Need of React, Simple React Structure, The Virtual DOM, React Components, Introducing React Components, Creating Components in React, Data and Data Flow in React, Rendering and Life Cycle Methods in React, Working with forms in React, integrating third party libraries, Routing in React.

TEXT BOOKS:

1. Brad Dayley, Brendan Dayley, Caleb Dayley., Node.js, MongoDB and Angular Web Development, 2nd Edition, Addison-Wesley, 2019.
2. Mark Tielens Thomas, React in Action, 1st Edition, Manning Publications, 2018.
3. Jon Duckett, Web Design with HTML, CSS, JavaScript and Jquery Set, Wiley, Pck edition,2014
4. Robin Nixon, Learning PHP, MySQL & JavaScript: A Step-by-Step Guide to CreatingDynamic Websites (Learning PHP, MYSQL, Javascript, CSS & HTML5), O'Reilly Media, 6th edition, 2021.
5. AzatMardan, Full Stack JavaScript: Learn Backbone.js, Node.js, and MongoDB, APress, 2nd ed.edition, 2018.

REFERENCE BOOKS:

1. Vasan Subramanian, Pro MERN Stack, Full Stack Web App Development with Mongo, Express, React, and Node, 2nd Edition, Apress, 2019.
2. Chris Northwood, The Full Stack Developer: Your Essential Guide to the Everyday SkillsExpected of a Modern Full Stack Web Developer', 1st edition, Apress, 2018.
3. KirupaChinnathambi, Learning React: A Hands-On Guide to Building Web Applications Using React and Redux, 2nd edition, Addison-Wesley Professional, 2018.
4. Eric Bush, Full-Stack JavaScript Development: Develop, Test and Deploy with Mongodb,Express, Angular and Node on Aws, Red Sky, 2016.
5. Tomasz Dyl, KamilPrzeorski, MaciejCzarnecki, Mastering Full Stack React Web Development, Packt Publishing Limited, 2016.

WEB LINKS:

1. https://www.fullstacklabs.co/
2. https://www.researchgate.net/publication/372345794_Full_Stack_Web_Development_with_Hands-On_Lab
3. https://www.udemy.com/course/the-practical-introduction-to-web-developmen.
4. https://www.coursera.org/learn/the-full-stack

OPERATING SYSTEMS

Sl. No.	Course Code	Course	L	T	P	Credits
4	25CS404PC	Operating Systems	3	0	0	3

Course Objectives:

- To understand the OS role in the overall computer system
- To study the operations performed by OS as a resource manager
- To understand the scheduling policies of OS
- To understand the different memory management techniques
- To understand process concurrency and synchronization
- To understand the concepts of input/output, storage and file management
- To understand the goals and principles of protection
- Introduce system call interface for file and process management
- To study different OS and compare their features.

Course Outcomes:

- Apply optimization techniques for the improvement of system performance.
- Ability to design and solve synchronization problems.
- Learn about minimization of turnaround time, waiting time and response time and
- also maximization of throughput by keeping CPU as busy as possible.
- Ability to change access controls to protect files.
- Ability to compare the different operating systems.

UNIT–I :

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

UNIT–II :

Process and CPU Scheduling - Process concepts and scheduling, Operations on processes, Cooperating Processes, Threads, and Interposes Communication, Scheduling Criteria, Scheduling Algorithms, Multiple -Processor Scheduling.

System call interface for process management-fork, exit, wait, wait pid, exec

UNIT–III :

Deadlocks - System Model, Deadlocks Characterization, Methods for Handling Deadlocks, DeadlockPrevention,DeadlockAvoidance,DeadlockDetection,andRecoveryfromDeadlock

Process Management and Synchronization - The Critical Section Problem, Synchronization Hardware,Semaphores,andClassicalProblemsofSynchronization,CriticalRegions,Monitors

Inter process Communication Mechanisms: IPC between processes on a single computer system,IPCbetweenprocessesondifferentsystems,usingpipes,FIFOs,messagequeues,shared memory.

UNIT–IV :

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 :

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.

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, Pearsoneducation.
3.

REFERENCE BOOKS:

1. Operating Systems – Internals and Design Principles Stallings, Fifth Edition–2005, Pearson Education/PHI
2. Operating System A Design Approach- Crowley,TMH.
3. Modern Operating Systems, AndrewS.Tanenbaum2ndedition, Pearson/PHI
4. UNIX programming environment, Kernighan and Pike,PHI/Pearson Education
5. UNIX Internals -The New Frontiers, U. Vahalia, Pearson Education.

DATABASE MANAGEMENT SYSTEMS

Sl. No.	Course Code	Course	L	T	P	Credits
5	25CS405PC	Database Management Systems	3	0	0	3

Course Objectives:

- To understand the basic concepts and the applications of database systems.
- To master the basics of SQL and construct queries using SQL.
- To understand the relational database design principles.
- To become familiar with the basic issues of transaction processing and concurrencycontrol.
- To become familiar with database storage structures and access techniques.

Course Outcomes:

- Demonstrate the basic elements of a relational database management system.
- Ability to identify the data models for relevant problems.
- Ability to design entity relationship model and convert entity relationship diagramsinto RDBMS and formulate SQL queries on the data.
- Apply normalization for the development of application software

UNIT – I:

Introduction to Database System Concepts: Database-System Applications, Purpose of Database Systems, View, Database Language, Database Architecture, Database Users and Administrators.

Introduction to the Relation Models and Database Design using ER Model: Overview of the Design Process, The Entity-Relationship Model, Constraints, Entity-Relationship Diagrams, Reduction to Relational Schemas, Entity-Relationship Design Issues, Extended E-R Feature, Structure of relational databases , database schema , keys, schema diagrams.

UNIT – II:

Introduction to SQL: Overview of the SQL Query Language, SQL Data Definition, Basic Structure of SQL Queries, Additional Basic Operations, Set Operations, Null Values, Aggregate Functions Nested Sub queries, Modification of theDatabase.

Intermediate and Advanced SQL: Join Expressions, Views, Integrity Constraints, SQL Data Types, Authorization. Functions and Procedures, Triggers.

UNIT – III:

Formal Relational Query Languages: The Relational operations, The Tuple Relational Calculus, The Domain Relational Calculus.

Relational Database Design: Features of Good Relational Designs, Atomic Domains and First Normal Form, Decomposition Using Functional Dependencies, Decomposition Using Multi valued Dependencies, BCNF.

UNIT – IV:

Transactions: Transaction Concept, a Simple Transaction Model, Storage Structure, Transaction Atomicity and Durability, Transaction Isolation, Serializability, Transaction Isolation and Atomicity.

Concurrency Control: Lock-Based Protocols, Deadlock Handling, Timestamp- Based Protocols.

UNIT-V:

Recovery System: Failure Classification, Recovery and Atomicity, Recovery Algorithm, Buffer Management, Failure with Loss of Nonvolatile Storage, ARIES, Remote Backup Systems.

TEXT BOOKS:

1. Abraham Silberschatz, Henry F. Korth,S.Sudarshan, Database System Concepts‖,6thEdition, Tata McGraw-Hill.
2. RaghuRamaKirshna,JohannesGehrk,DatabaseManagementSystem‖TataMcGrawHill 3rd Edition.

REFERENCE BOOKS:

1. PeterRob&CarlosCoronel―DatabaseSystemConceptsCengageLearning.
2. RamezElmasri, ShamkanthB. Navrate―Fundamentals of DatabaseSystems7th Edition, Pearson Education.
3. C.J. Date Introduction to Database Systems PearsonEducation

DESIGN AND ANALYSIS OF ALGORITHMS LABORATORY

Sl. No.	Course Code	Course	L	T	P	Credits
6	25CS406PC	Design And Analysis Of Algorithms Laboratory	0	0	2	1

Course Objectives:

- To write programs in java to solve problems using divide and conquer strategy.
- To write programs in java to solve problems using backtracking strategy.
- To write programs in java to solve problems using greedy and dynamic programming techniques.

Course Outcomes:

- Ability to write programs in java to solve problems using algorithm design techniquessuch as Divide and Conquer, Greedy, Dynamic programming, and Backtracking.

List of Experiments:

1. Write a java program to implement Quick sort algorithm for sorting a list of integersin ascending order
2. Write a java program to implement Merge sort algorithm for sorting a list of integersin ascending order.
3. i) Write a java program to implement the dfs algorithm for a graph.
4. ii) Write a. java program to implement the bfs algorithm for a graph.
5. Write a java programs to implement backtracking algorithm for the N-queensproblem.
6. Write a java program to implement the backtracking algorithm for the sum of subsetsproblem.
7. Write a java program to implement the backtracking algorithm for the HamiltonianCircuits problem.
8. Write a java program to implement greedy algorithm for job sequencing withdeadlines.
9. Write a java program to implement Dijkstra's algorithm for the Single source shortestpath problem.
10. Write a java program that implements Prim's algorithm to generate minimum costspanning tree.
11. Write a java program that implements Kruskal's algorithm to generate minimum costspanning tree
12. Write a java program to implement Floyd's algorithm for the all pairs shortest pathproblem.
13. Write a java program to implement Dynamic Programming algorithm for the 0/1Knapsack problem.
14. Write a java program to implement Dynamic Programming algorithm for the OptimalBinary Search Tree Problem.

REFERENCE BOOKS

1. Data structures, Algorithms and Applications in java, 2nd Edition, S. Sahani,Universities Press.
2. Data structures and Algorithms in java, 3rd edition, A. Drozdek, Cengage Learning.
3. Data structures with Java, J. R. Hubbard, 2nd edition, Schaum's Outlines, TMH.
4. Data structures and algorithms in Java, 2nd Edition, R. Lafore, Pearson Education.
5. Data Structures using Java, D. S. Malik and P.S. Nair, Cengage Learning.

OPERATING SYSTEMS LABORATORY

Sl. No.	Course Code	Course	L	T	P	Credits
7	25CS407PC	Operating Systems Laboratory	0	0	2	1

Course Objectives:

- To implement the scheduling algorithms.
- To implement page replacement algorithms
- To implement file allocation methods.
- To understand and implement ipc mechanism using named and unnamed pipes.
- To develop solutions for synchronization problems using semaphores.

Course Outcomes:

- Ability to implement interprocess communication between two processes.
- Ability to design and solve synchronization problems.
- Ability to simulate and implement operating system concepts such as scheduling,deadlock management, file management, and memory management.

LIST OF EXPERIMENTS:

1. Write C programs to simulate the following CPU Scheduling algorithms
a) FCFS b) SJF c) Round Robin d) priority
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 Bankers Algorithm for Deadlock Avoidance and Prevention.
4. Write a C program to implement the Producer – Consumer problem using semaphores using UNIX/LINUX system calls.
5. Write C programs to illustrate the following 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

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.

REFERENCE BOOKS:

1. Operating Systems – Internals and Design Principles, William Stallings, Fifth Edition– 2005, Pearson Education/PHI
2. Operating System - A Design Approach-Crowley,TMH.
3. Modern Operating Systems, Andrew S Tanenbaum,2ndedition, Pearson/PHI
4. UNIX Programming Environment, Kernighan and Pike,PHI/Pearson Education
5. UNIX Internals: The New Frontiers, U. Vahalia, Pearson Education

DATABASE MANAGEMENT SYSTEMS LABORATORY

Sl. No.	Course Code	Course	L	T	P	Credits
8	25CS408PC	Database Management Systems Laboratory	0	0	2	1

Course Objectives:

- Explain database concepts, SQL commands, and constraints.
- Create database schemas and perform database operations
- Write and execute complex SQL queries
- Perform normalization and analyze database structures
- Assess database designs and query optimization techniques

Course Outcomes:

- Ability to design and implement a database schema for given problem.
- Apply the normalization techniques for development of application software to realistic problems.
- Ability to formulate queries using SQL DML/DDL/DCL commands.

1. Database Schema for a customer-sale scenario

Customer(Cust id : integer, cust_name: string)
Item(item_id: integer,item_name: string, price:integer)
Sale(bill_no: integer, bill_data: date, cust_id: integer, item_id: integer, qty sold: integer)
For the above schema, perform the following
a. Create the tables with the appropriate integrityconstraints
b. Insert around 10 records in each of thetables
c. List all the bills for the current date with the customer names and itemnumbers
d. List the total Bill details with the quantity sold, price of the item and the finalamount
e. List the details of the customer who have bought a product which has aprice>200
f. Give a count of how many products have been bought by eachcustomer
g. Give a list of products bought by a customer having cust_id as5
h. List the item details which are sold as oftoday
i.Create a view which lists out the bill_no, bill_date, cust_id, item_id, price, qty_sold,amount Create a view which lists the daily sales date wise for the last oneweek

2. Database Schema for a Student Libraryscenario

Student(Stud_no : integer,Stud_name: string)
Membership(Mem_no: integer,Stud_no: integer)
Book(book_no: integer, book_name:string, author: string)
Iss_rec(iss_no:integer, iss_date: date, Mem_no: integer, book_no: integer)
For the above schema, perform the following
a. Create the tables with the appropriate integrityconstraints
b. Insert around 10 records in each of thetables

c. List all the student names with their membershipnumbers
d. List all the issues for the current date with student and Booknames
e. List the details of students who borrowed book whose author isCJDATE
f. Give a count of how many books have been bought by eachstudent
g. Give a list of books taken by student with stud_no as5
h. List the book details which are issued as oftoday
i. Create a view which lists out the iss_no, iss _date, stud_name, bookname
j. Create a view which lists the daily issues-date wise for the last oneweek

3. Database Schema for a Employee-payscenario
employee(emp_id:integer,emp_name:string)
department(dept_id:integer,dept_name:string)
paydetails(emp_id : integer,dept_id: integer, basic: integer, deductions: integer, additions: integer, DOJ: date)
payroll(emp_id : integer, pay_date: date)
For the above schema, perform the following
a. Create the tables with the appropriate integrityconstraints
b. Insert around 10 records in each of thetables
c. List the employee details departmentwise
d. List all the employee names who joined after particulardate
e. List the details of employees whose basic salary is between 10,000 and20,000
f. Give a count of how many employees are working in eachdepartment
g. Give a names of the employees whosenetsalary>10,000
h. List the details for an employee_id=5
i. Create a view which lists out the emp_name, department, basic, dedeuctions,netsalary
j. Create a view which lists the emp_name and his net salary

4. Database Schema for a Video Library scenario
Customer(cust_no: integer,cust_name: string)
Membership(Mem_no: integer, cust_no: integer)
Cassette(cass_no:integer, cass_name:string, Language:String)
Iss_rec(iss_no: integer, iss_date: date, mem_no: integer, cass_no: integer)
For the above schema, perform the following
a. Create the tables with the appropriate integrityconstraints
b. Insert around 10 records in each of thetables
c. List all the customer names with their membershipnumbers
d. List all the issues for the current date with the customer names and cassettenames
e. List the details of the customer who has borrowed the cassette whose title is ―The Legend‖
f. Give a count of how many cassettes have been borrowed by eachcustomer
g. Give a list of book which has been taken by the student with mem_no as5
h. List the cassettes issues fortoday
i. Create a view which lists outs the iss_no, iss_date, cust_name,cass_name
j. Create a view which lists issues-date wise for the last oneweek

5. Database Schema for a student-Lab scenario

Student(stud_no: integer, stud_name: string, class: string)
Class(class: string,descrip:string)
Lab(mach_no: integer, Lab no: integer, description: String)
Allotment(Stud_no: Integer, mach_no: integer, day of week: string)
For the above schema, perform the following
a. Create the tables with the appropriate integrityconstraints
b. Insert around 10 records in each of thetables
c. List all the machine allotments with the student names, lab and machinenumbers
d. List the total number of lab allotments daywise
e. Give a count of how many machines have been allocated to the 'IT'class
f. Give a machine allotment details of the stud_no 5 with his personal and classdetails
g. Count for how many machines have been allocated in Lab_no 1 for the day of the week as ―Monday‖
h. How many students class wise have allocated machines in thelabs
i. Create a view which lists out the stud_no, stud_name, mach_no, lab_no,dayofweek
j. Create a view which lists the machine allotmentdetailsfor―Thursday.

6**. Create a cursor, which displays all employee numbers and names from the EMP table.**
7. Create a cursor, which update the salaries of all employees as per the given data.
8. Create a cursor, which displays names of employees having salary > 50000.
9. Create a procedure to find reverse of a given number.
10. Create a procedure to update the salaries of all employees as per the given data.
11. Create a procedure to demonstrate IN, OUT and INOUT parameters.
12. Create a function to check whether given string is palindrome or not.
13. Create a function to find sum of salaries of all employees working in depart number 10.
14. Create a trigger before/after update on employee table for each row/statement.
15. Create a trigger before/after delete on employee table for each row/statement.
16. Create a trigger before/after insert on employee table for each row/statement.

FULL STACK DEVELOPMENT LABORATORY

Sl. No.	Course Code	Course	L	T	P	Credits
9	25CS409PC	Full Stack Development Laboratory	0	0	2	1

Course Objectives:

- To learn the core concepts of Full Stack Development.
- To understand the front-end programming like Data, Files, Http Requests and Responses usingnode.js.
- To know the storing and processing huge data and connects with NodeJS application using MangoDB.
- To provide an in-depth study of Angular Services in Web Applications.
- To explore interactive user interfaces with react structure.

Course Outcomes:

- Develop Full stack components for developing web application.
- Apply packages of NodeJS to work with Data, Files, Http Requests and Responses.
- Contrast MongoDB database for storing and processing huge data and connect with NodeJS application.
- Design faster and effective single page applications using Express and Angular.
- Create interactive user interfaces with react components.

List of Experiments:

1. Create an application to setup node JS environment and display “Hello World”.
2. Create a Node JS application for user login system.
3. Write a Node JS program to perform read, write and other operations on a file.
4. Write a Node JS program to read form data from query string and generate response usingNodeJS
5. Create a food delivery website where users can order food from a particular restaurant listed in the website for handling http requests and responses using NodeJS.
6. Implement a program with basic commands on databases and collections using MongoDB.
7. Implement CRUD operations on the given dataset using MongoDB.
8. Perform Count, Limit, Sort, and Skip operations on the given collections using MongoDB.
9. Develop an angular JS form to apply CSS and Events.
10. Develop a Job Registration form and validate it using angular JS.
11. Write an angular JS application to access JSON file data of an employee from a server using $http service.
12. Develop a web application to manage student information using Express and Angular JS.
13. Write a program to create a simple calculator Application using React JS.
14. Write a program to create a voting application using React JS
15. Develop a leave management system for an organization where users can apply different types of leaves such as casual leave and medical leave. They also can view the available number of days using react application.
16. Build a music store application using react components and provide routing among the web pages.

17. Create a react application for an online store which consist of registration, login, product information pages and implement routing to navigate through these pages.

TEXT BOOKS:

1. Brad Dayley, Brendan Dayley, Caleb Dayley., Node.js, MongoDB and Angular WebDevelopment, 2nd Edition, Addison-Wesley,2019.
2. Mark Tielens Thomas., React in Action, 1st Edition, Manning Publications, 2018.
3. Chris Northwood, The Full Stack Developer: Your Essential Guide to the Everyday SkillsExpected of a Modern Full Stack Web Developer, APress, 1st ed. edition, 2018.
4. JuhaHinkula, Hands-On Full Stack Development with Spring Boot 2.0 and React: Buildmodern and scalable full stack applications using the Java-based Spring Framework 5.0 andReact, Ingram short title, 2018.

REFERENCE BOOKS:

1. Vasan Subramanian, Pro MERN Stack, Full Stack Web App Development with Mongo,Express,React, and Node, 2nd Edition, Apress,2019.
2. Brad Green&Seshadri. Angular JS. 1st Edition. O'Reilly Media, 2013.
3. KirupaChinnathambi, Learning React: A Hands-On Guide to Building Web Applications UsingReact and Redux, 2nd edition, Addison-Wesley Professional, 2018.
4. Frank Zammetti, Modern Full-Stack Development: Using TypeScript, React, Node.js, Webpack,and Docker, Apress, 1st ed. edition, 2020.
5. Riaz Ahmed, Full Stack Web Development For Beginners: Learn Ecommerce WebDevelopment Using HTML5, CSS3, Bootstrap, JavaScript, MySQL, and PHP, IndependentlyPublished, 2021.

WEB LINKS:

1. https://www.fullstacklabs.co/
2. https://www.udemy.com/course/the-practical-introduction-to-web-development/
3. https://www.coursera.org/learn/the-full-stack
4. https://www.udemy.com/course/ultimate-web/

ENVIRONMENTAL SCIENCE

Sl. No.	Course Code	Course	L	T	P	Credits
11	25CH411AU	Environmental Science	3^	0	0	0

Course Objectives:

- Understanding the importance of ecological balance for sustainable development.
- Understanding the impacts of developmental activities and mitigation measures.
- Understanding the environmental policies and regulations
- Understand the impact of biodiversity and biotic resources
- Understand the impact of environmental pollution and control technologies.

Course Outcomes:

- Appreciate concepts and methods from ecological and physical sciences and their application inenvironmental problem solving.
- Analyze and synthesize scientific data to characterize and evaluate the status of atleast one type of ecological system and apply skills of measurement, spatial orientation, sampling, and dataanalysis to characterize natural resource phenomena.
- Create awareness on the basic philosophyof science, concepts and scope.
- Evaluate consequences of human exposure to pollution and its impacts to environmental quality.
- Comprehending the statutoryand regulatorymechanisms pertaining to environment in India andunderstanding judicial response to environmental issues in India.

UNIT – I:

Ecosystem: Definition, Scope and Importance of ecosystem, Structure and Functions of ecosystem: Food chains, Food Web and Ecological Pyramids, Flow of energy; Bio-magnification.

Biodiversity and Biotic Resources: Introduction, Definition, levels of Biodiversity, Values of biodiversity, Hot spots of biodiversity, Threats to biodiversity, conservation of biodiversity: In-Situ and Ex-situ conservation.

UNIT – II:

Natural Resources: Classification of Resources, **Water resources**: use and over utilization of surface and ground water, Dams: benefits and problems, Rain water harvesting; **Energy resources**: growing energy needs, Renewable and Non Renewable Energy resources. **Land resources**: land degradation – Landslide and Soil Erosion; **Forest Resources –** Uses and Exploitation.

UNIT – III:

Environmental Pollution And Control: Types of Pollution, Sources, Effects and Control measures of Air Pollution, WaterPollution, Soil Pollution and Noise Pollution.

UNIT – IV:

Global Environmental Problems and Global Efforts: Green house effect, Global Warming, climate change and their impacts on human environment; Ozone depletion and Ozone depleting substances (ODS); Aid Rains.

Environmental Impact Assessment (EIA): Scope of EIA, EIA methods, scope of Environmental audit and Environmental Management Plan.

UNIT - V:

Environmental Policy, Legislation, Rules And Regulations: Salient features of Environmental Protection act, Air (Prevention and Control of pollution) Act- 1981, Water (Prevention and Control of pollution) Act-1974, Forest Conservation Act, Municipal solid waste, Hazardous waste, E-waste, Bio-medical waste and Radioactive waste Rules.

Towards Sustainable Future: Concept of Sustainable Development, Sustainable goals defined by UN, Threats to Sustainability, Environmental Education, Role of IT in Environment, Smart Cities, Concept of Green Building, Low Carbon Lifestyle, Life cycle assessment and Ecological Foot Print.

TEXT BOOKS:

1. Text Book of Environmental Studies by Anubha Kaushik (4th Edition), New age International Publishers.
2. Environmental studies by ErachBharucha 2005, University Grants Commission, University Press.

REFERENCE BOOKS:

1. Text book of Environmental Science and Technology by M.Anji Reddy 2007
2. Environmental Science: Towards a Sustainable Future by Richard T. Wright. 2008 PHL Learning Private Ltd. New Delhi.
3. Environmental studies, From crisis to cure by R.Rajagopalan, 2005.