Kaveri University
B.Tech. Computer Science and Engineering (AIML)
Course Structure
Applicable for 2025-2026 Admitted Batch

II Year

SEMESTER-III						
Sl.No.	**Course Code**	**CourseTitle**	**L**	**T**	**P**	**Credits**
1	25AM301PC	Machine Learning	3	0	0	3
2	25MA302BS	Mathematics−III (Mathematical foundation of Data Science)	3	1	0	4
3	25AM303PC	Digital Logic and Computer Organization	3	0	0	3
4	25CS304PC	Data Structures	3	0	0	3
5	25CS305PC	Object Oriented Programming using Java	3	0	0	3
6	25AM306PC	Machine Learning Laboratory	0	0	2	1
7	25CS307PC	Data Structures Laboratory	0	0	2	1
8	25CS308PC	Object Oriented Programming using Java Laboratory	0	0	2	1
9	25EN309HS	Soft Skills and Inter Personal Skills Laboratory	0	0	2	1
10	25CS310ES	Drone Technology Workshop	1	0	2	2
		Total	**16**	**1**	**10+2 ^**	**22**

^represents related to Audit Course.

SEMESTER-IV						
Sl. No.	**Course Code**	**CourseTitle**	**L**	**T**	**P**	**Credits**
1	25AM401PC	Artificial Intelligence	3	0	0	3
2	25CS402PC	Design & Analysis of Algorithms	3	0	0	3
3	25AM403PC	Software Engineering	3	0	0	3
4	25CS404PC	Operating Systems	3	0	0	3
5	25CS405PC	Data Base Management Systems	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 Systems Laboratory	0	0	2	1
9	25AM409PC	Artificial Intelligence Laboratory	0	0	2	1
10	25AM410PC	Software Engineering Laboratory	0	0	2	1
11	25CH411AU	Environmental Science	3^	0	0	0
12	25AM412EEC	Minor Project	0	0	6	3
		Total	**15+3^**	**0**	**16**	**23**

^representsrelated to Audit Course.

MACHINE LEARNING

Sl. No.	Course Code	Course	L	T	P	Credits
1	25AM301PC	MACHINE LEARNING	3	0	0	3

Course Objectives:

- To introduce fundamental concepts of machine learning, including learning types, processes, and real-world applications.
- To understand data preprocessing, feature engineering, and model evaluation techniques for effective learning.
- To study supervised learning algorithms for regression and classification tasks.
- To explore unsupervised learning techniques such as clustering and their applications.
- To understand the basics of artificial neural networks and emerging paradigms like deep learning and reinforcement learning.

Course Outcomes:

- Explain machine learning concepts, types, and the complete learning process including data preparation and evaluation.
- Apply feature engineering techniques and probabilistic models such as Bayes classifiers.
- Implement and evaluate supervised learning algorithms for regression and classification problems.
- Analyse and apply unsupervised learning methods like clustering for data analysis.
- Understand and apply neural networks and identify applications of deep learning and reinforcement learning in real-world scenarios.

UNIT - I

Introduction to Machine Learning: Types of Human learning, machine learning process, Well-posed learning problem, Types of machine learning and comparison, applications of machine learning. Model Preparation

Evaluation and feature engineering: Machine learning activities, Types of data in machine learning, dataset understanding, data pre-processing, selecting a model, predictive and descriptive models, supervised learning model training, cross-validation and boot strapping, lazy vs eager learner, interpreting the model- underfitting, overfitting.

UNIT – II

Feature Engineering: Feature transformation - feature construction, feature extraction by PCA, SVD, LDA. Feature subset selection – feature relevancy and redundancy measures. Feature selection process and approaches.

Review of Probability concepts: joint probability, conditional probability, Bayes rule, Common discrete and continuous distributions, dealing with multiple random variables, central limit theorem. Bayes classifier, Multi-class Classification, Naïve Bayes classifier, Bayesian belief network.

UNIT – III

Supervised Learning - Introduction to supervised learning, Regression: Introduction of regression, Regression algorithms: Simple linear regression, Multiple linear regression, Polynomial regression model, Logistic regression, Maximum likelihood estimation.

Classification: Classification model and learning steps, Classification algorithms: Distance measures, k-Nearest Neighbor (KNN), Decision tree, Support vector machines, Kernel trick, Random Forest.

UNIT - IV

Unsupervised Learning: Introduction to unsupervised learning, unsupervised vs supervised learning, Application of unsupervised learning, Clustering and its types,

Partitioning method: K-Means and K-Medoids, Hierarchical clustering, Density-based methods – DBSCAN.

UNIT - V

Artificial Neural Network: Biological neuron, artificial neuron, Activation functions, neural network architecture, perceptron, learning process in ANN, Back propagation.

Introduction to deep learning: overview of reinforcement learning, Representation learning, Evolutionary learning. Case-study of ML applications: Image recognition, Email spam filtering, Online fraud detection.

TEXT BOOKS:

1. SaikatDutt, S. Chjandramouli, Das – Machine Learning, Frist Edition, Pearson
2. M N Murty, Anathanarayana V S – Machine Learning, First Edition, University Press
3. Tom M Mitchell, ―Machine Learning, First Edition, McGraw Hill Education, 2013.

REFERENCE BOOKS:

1. Stephen Marsland, ―Machine Learning – An Algorithmic Perspective, Second Edition,
2. Chapman and Hall/CRC Machine Learning and Pattern Recognition Series, 2014.

MATHEMATICS–III (MATHEMATICAL FOUNDATION OF DATA SCIENCE)

Sl. No.	Course Code	Course	L	T	P	Credits
2	25MA302BS	MATHEMATICS–III (Mathematical Foundation for Data Science)	3	1	0	4

Course Objectives

- To develop understanding of linear algebra concepts relevant to data analysis.
- To provide knowledge of probability theory and random variables for modeling uncertainty in data.
- To introduce statistical inference and hypothesis testing techniques for data-driven decision making.
- To apply regression and least-square methods for curve fitting and predictive analytics. To understand optimization principles and their applications in machine learning models such as regression and classification.

Course Outcomes

- Apply linear algebra concepts such as matrices, eigenvalues, eigenvectors, projections, and matrix factorizations in data analysis problems.
- Analyze random phenomena using probability theory, conditional probability, Bayes' theorem, and random variables.
- Perform statistical inference using sampling distributions and hypothesis testing techniques.
- Apply least-square techniques and regression models to fit curves and analyze data.
- Interpret optimization concepts and develop basic machine learning models for regression and classification problems.

UNIT-1: Linear Algebra

Matrices and their properties (determinants, traces, rank, nullity, etc.); Eigenvalues and eigenvectors; Matrix factorizations; Inner products; Distance measures; Projections; Notion of hyper planes; half-planes.

UNIT-II: 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-III: Testing of Hypothesis

Sampling Distributions – 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- Chi-square test- for the independence of attributes.

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

UNIT - V
Optimization view of machine learning. Introduction to Data Science Methods: Linear regression as an exemplar function approximation problem; Linear classification problems.

Text Books:

1. Gilbert Strang, *Introduction to Linear Algebra*, 5th Edition, Wellesley-Cambridge Press, 2016.
2. S. C. Gupta and V. K. Kapoor, *Fundamentals of Mathematical Statistics*, 11th Edition, Sultan Chand & Sons, 2015.
3. J. S. Bendat and A. G. Piersol, *Random Data: Analysis and Measurement Procedures*, 4th Edition, Wiley, 2010.

Reference Books:

1. Montgomery, D. C. and G. C. Runger. Applied Statistics and Probability for Engineers. 5th Edition. John Wiley & Sons, Inc., NY, USA, 2011.
2. David G. Luenberger . Optimization by Vector Space Methods, John Wiley & Sons (NY), 1969. 5. Cathy O'Neil and Rachel Schutt . Doing Data Science, O'Reilly Media, 2013.

DIGITAL LOGIC AND COMPUTER ORGANIZATION

Sl. No.	Course Code	Course	L	T	P	Credits
3	25AM303PC	Digital Logic and Computer Organization	3	0	0	3

Course Objectives

- To understand number systems, codes, and fundamentals of Boolean Algebra used in digital systems.
- To apply logic minimization techniques for designing efficient digital circuits.
- To explain the basic structure and operation of digital computers and data representation methods.
- To analyze register transfer operations, micro-operations, and control unit design.
- To understand CPU organization and perform arithmetic operations used in computer systems.

Course Outcomes

- Apply number systems, complements, and error detection techniques in digital systems.
- Simplify Boolean expressions and design combinational circuits using minimization techniques.
- Explain digital computer organization, data representation, and register transfer operations.
- Analyse instruction execution, control unit design, and input-output organization.
- Evaluate CPU organization and perform arithmetic operations including multiplication and division algorithms.

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

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–IV

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–V

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.

TEXT BOOKS:

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

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 andAVL-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 linkedre 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 PROGRAMMING 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:

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

Course Outcomes:

- Understand OOP concepts to apply basic Java constructs.
- Analyse different forms of inheritance and usage of Exception Handling
- Understand the different kinds of file I/O and Multithreading in complex Java programs, and usage of Container classes
- Contrast different GUI layouts and design GUI applications
- 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:

1. Thinking in Java Fourth Edition, Bruce Eckel
2. Introduction to Java programming, Y. Daniel Liang, Pearson Education.
3. Understanding OOP with Java, updated edition, T. Budd, Pearson Education.

MACHINE LEARNING LABORATORY

Sl. No.	Course Code	Course	L	T	P	Credits
6	25AM306PC	Machine Learning Laboratory	0	0	2	1

Course Objectives

- To understand and implement basic statistical measures such as central tendency and dispersion using Python.
- To familiarize with Python libraries including Statistics, Math, NumPy, SciPy, Pandas, and Matplotlib for data analysis.
- To develop skills in data preprocessing, visualization, and handling datasets for machine learning tasks.
- To implement and analyse supervised learning algorithms such as Linear Regression, Decision Tree, KNN, and Logistic Regression.
- To explore unsupervised learning techniques like K-Means clustering and evaluate model performance on real datasets.

Course Outcomes

- Compute statistical measures such as mean, median, mode, variance, and standard deviation using Python.
- Use Python libraries like NumPy, Pandas, and Matplotlib for data manipulation and visualization.
- Implement regression models including simple and multiple linear regression using Python and sklearn.
- Apply classification algorithms such as Decision Tree, KNN, and Logistic Regression and tune their parameters.
- Perform clustering and evaluate machine learning models through a mini project using real-world datasets.

List of Experiments:

1. Write a python program to compute Central Tendency Measures: Mean, Median, Mode, Measure of Dispersion: Variance, Standard Deviation.
2. Study of Python Basic Libraries such as Statistics, Math, Numpy and Scipy.
3. Study of Python Libraries for ML application such as Pandas and Matplotlib.
4. Write a Python program to implement Simple Linear Regression.
5. Implementation of Multiple Linear Regression for House Price Prediction using sklearn.
6. Implementation of Decision tree using sklearn and its parameter tuning
7. Implementation of KNN using sklearn
8. Implementation of Logistic Regression using sklearn
9. Implementation of K-Means Clustering
10. Performance analysis of Classification Algorithms on a specific dataset (Mini Project)

TEXT BOOK:

1. Machine Learning – Tom M. Mitchell, - MGH.

REFERENCE BOOK:

1. Machine Learning: An Algorithmic Perspective, Stephen Marshland, Taylor & Francis.

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 Analyse 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.

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:

1. Thinking in Java Fourth Edition, Bruce Eckel
2. Introduction to Java programming, Y. Daniel Liang, Pearson Education.
3. Understanding OOP with Java, updated edition, T. Budd, Pearson Education.

SOFT SKILLS AND INTERPERSONAL SKILLS

Sl. No.	Course Code	Course	L	T	P	Credits
9	25EN309HS	Soft Skills and Interpersonal Skills Lab	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:

- Soft skills for Everyone - Jeff Butterfield, CENAGE Learning
- Soft skills for Interpersonal Communication – Balasubramanian (ORIENT BLACKSWAN)
- *Technical Communication: Principles and Practice* - Meenakshi Raman and Sangeeta Sharma

Online Sources

- 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:

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

Course Outcomes:

- Understand and comply with aviation regulations, meteorology, and air traffic services as required for commercial drone operations.
- Analyse UAV payloads, ground control stations, and technical data to ensure efficient operation and regular servicing.
- Apply standard operating procedures for mission planning, drone assembly, and handling of flight emergencies.
- Demonstrate accurate flight manoeuvres, orientation awareness, and recovery strategies using a flight simulator.
- 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.
2. *Fundamentals of Drone Technology: Drones- The future of 21st century.* - Mr. I.V.S.Yeswanth, Dr.A.V.S.Sridhar Kumar

ARTIFICIAL INTELLIGENCE

Sl. No.	Course Code	Course	L	T	P	Credits
1	25AM401PC	Artificial Intelligence	3	0	0	3

Course Objectives

- To introduce the fundamentals of Artificial Intelligence and intelligent agents, including problem-solving and search techniques.
- To understand uninformed and informed search strategies and their applications in AI problems.
- To study knowledge representation using propositional and first-order logic and apply inference mechanisms.
- To explore planning, constraint satisfaction problems, and reasoning techniques in AI systems.
- To understand uncertainty handling and probabilistic reasoning using concepts like Bayes' rule and Bayesian networks.

Course Outcomes

- Explain AI concepts, intelligent agents, and apply search strategies to solve problems.
- Analyse adversarial search problems, constraint satisfaction problems, and apply logical reasoning methods.
- Represent knowledge using propositional and first-order logic and perform inference.
- Apply planning algorithms and reasoning techniques for solving complex AI problems.
- Analyse uncertainty in AI systems and apply probabilistic reasoning using Bayesian methods.

UNIT - I

Introduction to AI, Intelligent Agents, problem-Solving Agents, Searching for Solutions, Uninformed Search Strategies: Breadth-first search, Uniform cost search, Depth-first search, Iterative deepening Depth-first search, Bidirectional search, Informed (Heuristic) Search Strategies: Greedy best-first search, A* search, Heuristic Functions, Beyond Classical Search: Hill-climbing search, Simulated annealing search, Local Search in Continuous Spaces

UNIT - II

Problem Solving by Search-II and Propositional Logic Adversarial Search: Games, Optimal Decisions in Games, Alpha–Beta Pruning, Imperfect Real-Time Decisions. Constraint Satisfaction Problems: Defining Constraint Satisfaction Problems, Constraint Propagation, Backtracking Search for CSPs, Local Search for CSPs, The Structure of Problems.

Propositional Logic: Knowledge-Based Agents, The Wumpus World, Logic, Propositional Logic

Propositional Theorem Proving: Inference and proofs, Proof by resolution, Horn clauses and definite clauses, Forward and backward chaining, Effective Propositional Model Checking, Agents Based on Propositional Logic.

UNIT - III

Logic and Knowledge Representation First-Order Logic: Representation, Syntax and Semantics of First-Order Logic, Using First-Order Logic, Knowledge Engineering in First-Order Logic. Inference in First-Order Logic: Propositional vs. First-Order Inference, Unification and Lifting, Forward Chaining, Backward Chaining, Resolution.

UNIT - IV

Knowledge Representation: Ontological Engineering, Categories and Objects, Events. Mental Events and Mental Objects, Reasoning Systems for Categories, Reasoning with Default Information. **Classical Planning:** Definition of Classical Planning, Algorithms for Planning with State-Space Search, Planning Graphs, other Classical Planning Approaches, Analysis of Planning approaches.

UNIT - V

Uncertain knowledge and Learning Uncertainty: Acting under Uncertainty, Basic Probability Notation, Inference Using Full Joint Distributions, Independence, Bayes' Rule and Its Use **Probabilistic Reasoning:** Representing Knowledge in an Uncertain Domain, The Semantics of Bayesian Networks, Efficient Representation of Conditional Distributions, Approximate Inference in Bayesian Networks, Relational and First-Order Probability, Other Approaches to Uncertain Reasoning; Dempster-Shafer theory.

TEXT BOOK:

1. Artificial Intelligence: A Modern Approach, Third Edition, Stuart Russell and Peter Norvig, Pearson Education.

REFERENCE BOOKS:

1. Artificial Intelligence, 3rd Edn, E. Rich and K. Knight (TMH)
2. Artificial Intelligence, 3rd Edn., Patrick Henry Winston, Pearson Education.
3. Artificial Intelligence, ShivaniGoel, Pearson Education.
4. Artificial Intelligence and Expert systems – Patterson, Pearson Education

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 theirapplication.
- Critically assess and contrast different algorithms in terms of efficiency, scalability, andcorrectness 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 their relationships 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 spacetrade-offs.
- Able to Device backtracking and dynamic programming solutions.
- Able to Apply greedy methods and graph traversal algorithms
- Able to Analyse 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.

SOFTWARE ENGINEERING

Sl. No.	Course Code	Course	L	T	P	Credits
3	25AM403PC	Software Engineering	3	0	0	3

Course Objective

- To understand fundamental concepts of software engineering, process models, and the evolution of software systems.
- To learn techniques for software requirements analysis, specification, and management.
- To gain knowledge of software design principles, architectural design, and UML modelling.
- To understand software testing strategies, debugging techniques, and quality metrics.
- To study risk management practices and software quality assurance standards such as ISO 9000.

Course Outcomes

- Apply appropriate software process models for developing software systems.
- Analyse and document functional and non-functional requirements effectively.
- Design software architecture and represent it using UML diagrams.
- Evaluate software quality using testing techniques and relevant metrics.
- Identify, analyse, and manage risks while ensuring software quality using standard practices.

UNIT - I

Introduction to Software Engineering: The evolving role of software, changing nature of software, software myths. A Generic view of process: Software engineering- a layered technology, a process framework, the capability maturity model integration (CMMI). Process models: The waterfall model, Spiral model, Incremental Process Models, Concurrent Models, Component based development and Agile Development.

UNIT - II

Software Requirements: Functional and non-functional requirements, user requirements, system requirements, interface specification, the software requirements document. Requirements engineering process: Feasibility studies, requirements elicitation and analysis, requirements validation, requirements management.

UNIT - III

Design Engineering: Design process and design quality, design concepts, the design model. Creating an architectural design: software architecture, data design, architectural styles and patterns, architectural design, conceptual model of UML, basic structural modeling, use case diagrams, class diagrams, sequence diagrams, collaboration diagrams, activity diagrams and component diagrams.

UNIT – IV

Testing Strategies: A strategic approach to software testing, test strategies for conventional software, black-box and white-box testing, validation testing, system testing, the art of debugging. Metrics for Process and Products: Software measurement, metrics for software quality.

UNIT - V

Risk management: Reactive Vs proactive risk strategies, software risks, risk identification, risk projection, risk refinement, RMMM. Quality Management: Quality concepts, software quality assurance, software reviews, formal technical reviews, statistical software quality assurance, software reliability, the ISO 9000 quality standards

TEXT BOOKS:

1. Software Engineering, A practitioner's Approach- Roger S. Pressman, 6th edition, McGraw Hill International Edition.
2. Software Engineering- Sommerville, 7th edition, Pearson Education.
3. The unified modelling language user guide, Grady Booch, James Rumbaugh, Ivar Jacobson, Pearson Education.

REFERENCE BOOKS:

1. Software Engineering, an Engineering approach- James F. Peters, WitoldPedrycz, John Wiley.
2. Software Engineering principles and practice- Waman S Jawadekar, The McGraw-Hill Companies.
3. Fundamentals of object-oriented design using UML Meilerpage-Jones: Pearson Education. 4. Fundamentals of Software Engineering-Rajib Mall, PHI.

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.

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 diagrams into 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 Non-volatile 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,DatabaseManagementSystemTataMcGrawHill 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 programmingtechniques.

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.
   ii) Write a. java program to implement the BFS algorithm for a graph.
4. Write a java program to implement backtracking algorithm for the N-queensproblem.
5. Write a java program to implement the backtracking algorithm for the sum of subsetsproblem.
6. Write a java program to implement the backtracking algorithm for the HamiltonianCircuits problem.
7. Write a java program to implement greedy algorithm for job sequencing withdeadlines.
8. Write a java program to implement Dijkstra's algorithm for the Single source shortestpath problem.
9. Write a java program that implements Prim's algorithm to generate minimum costspanning tree.
10. Write a java program that implements Kruskal's algorithm to generate minimum costspanning tree
11. Write a java program to implement Floyd's algorithm for the all pairs shortest pathproblem.
12. Write a java program to implement Dynamic Programming algorithm for the 0/1Knapsack problem.
13. 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 anemployee_id=5
i. Create a view which lists out the emp_name, department, basic, deductions,netsalary
j. Create a view which lists the emp_name and hisnetsalary

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 detail 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 whichliststhe 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.

ARTIFICIAL INTELLIGENCE LABORATORY

Sl. No.	Course Code	Course	L	T	P	Credits
9	25AM409PC	Artificial Intelligence Laboratory	0	0	2	1

Course Objectives:
Become familiar with basic principles of AI toward problem solving, knowledge representation, and learning.

Course Outcomes:
Apply basic principles of AI in solutions that require problem solving, knowledge representation, and learning.

LIST OF EXPERIMENTS
Write a Program to Implement the following using Python.
1. Breadth First Search
2. Depth First Search
3. Tic-Tac-Toe game
4. 8-Puzzle problem
5. Water-Jug problem
6. Travelling Salesman Problem
7. Tower of Hanoi
8. Monkey Banana Problem
9. Alpha-Beta Pruning
10. 8-Queens Problem

TEXT BOOK:
1. Artificial Intelligence a Modern Approach, Third Edition, Stuart Russell and Peter Norvig, Pearson Education.

REFERENCE BOOKS:
1. Artificial Intelligence, 3rd Edn, E. Rich and K. Knight (TMH)
2. Artificial Intelligence, 3rd Edn., Patrick Henny Winston, Pearson Education.
3. Artificial Intelligence, ShivaniGoel, Pearson Education.

SOFTWARE ENGINEERING LABORATORY

Sl. No.	Course Code	Course	L	T	P	Credits
10	25AM410PC	Software Engineering Laboratory	0	0	2	1

Course Objectives:

- To have hands-on experience in developing a software project by using various software engineering principles and methods in each of the phases of software development.

Course Outcomes:

- Ability to translate end-user requirements into system and software requirements
- Ability to generate a high-level design of the system from the software requirements
- Will have experience and/or awareness of testing problems and will be able to develop a simple testing report

List of Experiments

Do the following seven exercises for any two projects given in the list of sample projects or any other Projects:

1. Development of problem statements.
2. Preparation of Software Requirement Specification Document, Design Documents and Testing Phase related documents.
3. Preparation of Software Configuration Management and Risk Management related documents.
4. Study and usage of any Design phase CASE tool
5. Performing the Design by using any Design phase CASE tools.
6. Develop test cases for unit testing and integration testing
7. Develop test cases for various white box and black box testing techniques.

Sample Projects:

1. Passport automation System
2. Book Bank
3. Online Exam Registration
4. Stock Maintenance System
5. Online course reservation system
6. E-ticketing
7. Software Personnel Management System
8. Credit Card Processing
9. E-book management System.
10. Recruitment system

TEXT BOOKS:

1. Software Engineering, A practitioner's Approach- Roger S. Pressman, 6th edition, McGraw Hill International Edition.
2. Software Engineering- Sommerville, 7th edition, Pearson Education.
3. The unified modeling language user guide Grady Booch, James Rambaugh, Ivar Jacobson, Pearson Education.

REFERENCE BOOKS:

1. Software Engineering, an Engineering approach- James F. Peters, WitoldPedrycz, John Wiley.
2. Software Engineering principles and practice- Waman S Jawadekar, The McGraw-Hill

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.
- Analyse 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.