

**DEBRA THANA SAHID KSHUDIRAM
SMRITI MAHAVIDYALAYA**
Gangaram Chak, Chakshyampur, Debra, West Bengal



PROPOSED SYLLABUS (DRAFT) OF

**MAJOR COURSE UNDER CCFUP, 2023
FOR SEMESTER-V & VI**

**FOR COMPUTER SCIENCE(MAJOR) PROGRAMMES
*(w.e.f. Academic Year 2025-2026)***

Based on

**Curriculum & Credit Framework for Undergraduate
Programmes (CCFUP), 2023 & NEP, 2020**

Level	YR.	SEM	Course Type	Course Code	Course Title	Credit	L-T-P	Marks					
								CA	ESE	TOTAL			
B.Sc. (Hons.)	3rd	V	SEMESTER-V										
			Major-8	UG/V/COMP/4/MJ-8T	Design and Analysis of Algorithms; T: Theory	4	3	3-0-1	15	60	75		
				UG/V/COMP/4/MJ-8P	Design and Analysis of Algorithms; P: Practical		1						
			Major-9	UG/V/COMP/4/MJ-9T	Database Management Systems; T: Theory	4	3	3-0-1	15	60	75		
				UG/V/COMP/4/MJ-9P	Database Management Systems; P: Practical		1						
			Major-10	UG/V/COMP/4/MJ-10T	Discrete Structure; T: Theory	4		3-1-0	15	60	75		
			Major -Elective 1	UG/V/COMP/4/MJ/DSE-1T	Programming in Java / Kotlin Programming / MATLAB Programming; T	4	3	3-1-0/	15	60	75		
				UG/V/COMP/4/MJ/DSE-1P	Programming in Java / Kotlin Programming / MATLAB Programming; P		1	3-0-1					
			Minor- 5 (Disc.-I)	UG/V/COMP/4/MI-5T	Operating System; T: Theory (To be taken from other Discipline)	4	3	3-0-1	15	60	75		
				UG/V/COMP/4/MI-5P	Operating System; P: Practical		1						
			Semester-V Total						20				375
			VI	SEMESTER- VI									
				Major-11	UG/VI/COMP/4/MJ-11T	Automata Theory; T: Theory	4		3-0-1	15	60	75	
				Major-12	UG/VI/COMP/4/MJ-12T	Artificial Intelligence; T: Theory	4	3	3-0-1	15	60	75	
		UG/VI/COMP/4/MJ-12P			Artificial Intelligence; P: Practical		1						
		Major-13		UG/VI/COMP/4/MJ-13T	Probability and Statistics; T: Theory	4	3	3-0-1	15	60	75		
				UG/VI/COMP/4/MJ-13P	Probability and Statistics; P: Practical		1						
		Major -Elective 2		UG/VI/COMP/4/MJ/DSE-2T	Advance JAVA / Internet of thing (IoT) / Cloud Computing; T: Theory	4	3	3-1-0 /	15	60	75		
				UG/VI/COMP/4/MJ/DSE-2P	Advance JAVA / Internet of thing (IoT) / Cloud Computing; P: Practical		1	3-0-1					
		Minor- 6 (Disc.-II)		UG/VI/COMP/4/MI-6T	Database Management Systems; T: Theory (To be taken from other Discipline)	4	3	3-0-1	15	60	75		
				UG/VI/COMP/4/MI-6P	Database Management Systems; P: Practical		1						
		Semester-VI Total						20				375	
		Eligible to be awarded Bachelor of Science in Computer Science on Exit						126	Marks (Year: I+II+III)		2325		

MJ = Major, MI = Minor Course, DSE = Discipline Specific Elective Course, CA= Continuous Assessment, ESE= End Semester Examination, T = Theory, P= Practical, L-T-P = Lecture-Tutorial-Practical

Semester V

MAJOR

UG/V/COMP/4/MJ-8: Design and Analysis of Algorithms

Credit 04

Objective of the course:

- Equip students with the skills to create efficient algorithms and analyze their performance.
- Cover fundamental algorithmic techniques such as divide and conquer, dynamic programming, and greedy algorithms.
- Teach students to analyze time and space complexity of algorithms using Big O notation.
- Emphasize problem-solving strategies for designing optimal solutions to computational problems.
- Study data structures and their impact on algorithm efficiency.
- Provide practical experience through programming assignments and projects.
- Explore advanced topics like NP-completeness and approximation algorithms for addressing complex problems.
- Prepare students to evaluate and develop efficient algorithms, laying the foundation for careers in software development, data analysis, and research in computer science.

MJ-8(T): Design and Analysis of Algorithms (T)

Credits 03

Course Contents:

Module I : Basics of an Algorithm

15 Hrs.

Definition, characteristics and steps in designing of algorithms, Asymptotic notations (O , Ω , Θ etc.), Recursion (Hanoi problem, Tail Recursion.) and Recurrence relation, Substitution Method, Iteration Method, Master Method. Concept of efficiency of analysis of an algorithm (Linear Search, Sorting, Bubble sort, Insertion Sort, Selection sort.) Comparative efficiencies of algorithms: Linear, Quadratic, Polynomial and Exponential.

Module II: Divide and Conquer Approach

08 Hrs.

General Issues in Divide and Conquer, Binary Search, Merge Sort, Quick Sort, Heap Sort.

Module III: Greedy Technique

08 Hrs.

Elements of Greedy strategy, Activity Selection Problem, Knapsack Problem, Coin changing Problem, single-source shortest path (Dijkstra's algorithm), Job sequencing with deadlines.

Module IV: Dynamic Programming.

08 Hrs.

Basic method, use, Examples: matrix-chain multiplication, All pair shortest paths, single source shortest path(Bellman Ford),Traveling Salesman problem, Longest common sequence.

Module V: Graph Algorithm

05 Hrs.

Representation of Graphs, Adjacency Matrix, Adjacency List, Depth First Search and Examples, Breadth First Search and Examples, Spanning Tree(Prims, Kruskal's).

Module VI : String Processing

03 Hrs.

String matching, KMP algorithm, Rabin-Karp algorithm.

Module VII : Limitations of Algorithmic Power

03 Hrs.

Overview of non-deterministic algorithms, P, NP, NP-Complete and NP-hard problems

Reference books:

1. T.H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein Introduction to Algorithms, PHI, 3rd Edition 2009
2. Sarabasse& A.V. Gelder Computer Algorithm – Introduction to Design and Analysis, Publisher – Pearson 3rd Edition 1999

MJ8P: Design and Analysis of Algorithms Practical

Credits 01

1. Implement Bubble Sort, Selection Sort, Insertion Sort, Merge Sort, Quick Sort.(The program should report the number of comparisons)
2. Implement Breadth-First Search in a graph
3. Implement Depth-First Search in a graph
4. Write a program to determine the minimum spanning tree of a graph
5. Write a program to find minimum cost spanning tree using Kruskal's Algorithm.
6. Write a program to perform Single source shortest path problem for a given graph.
7. Write a program to find solution for job sequencing with deadlines problem.
8. Write a program for all pairs shortest path problem.
9. Write a Program to perform Binary Search for a given set of integer values recursively and non-recursively.
10. Write a program to find solution for knapsack problem using greedy method.

UG/V/COMP/4/MJ-9: Database Management System

Credit 04

Objective of the Course:

- Provide a comprehensive understanding of database design, implementation, and management.
- Cover fundamental concepts of database architecture, data models, and relational database management systems (RDBMS).
- Teach students to design and normalize databases to ensure data integrity and eliminate redundancy.
- Emphasize the use of SQL for querying and manipulating databases, including creating, updating, and deleting records.
- Explore advanced topics such as transactions, concurrency control, and database security for reliable data handling.
- Offer hands-on experience through projects involving the design and implementation of database systems.
- Equip students with skills to design, manage, and optimize databases.
- Prepare students for careers in database administration, data analysis, and software development.

MJ-9(T): Database Management System(T)

Credit 03

Course Contents:

Module- I: Introduction

04 Hrs.

Concept & Overview of DBMS, Data Models, Database Languages, Database Administrator, Database Users, Data Abstraction, Three Schema architecture of DBMS.

Module- II: Entity Relationship (ER) Model

04 Hrs.

Entity Set, Simple and composite Attribute, Single valued and multivalued attribute, Relationship sets, Mapping cardinality, keys, , Entity Relationship Diagram : Need for E-R Model, Various steps of database design, Mapping Constraints, E-R diagram, Subclass, Generalization, Specialization, Aggregation, Strong Entity-Weak Entity, Tabular representation of Strong entity set, tabular representation of weak entity set, ER diagram.

Module- III: Relational Algebra

09 Hrs.

Select operation, Project Operation, Set operations (union, intersection, difference), Join operations, Division operation, outer join and outer union, Examples queries in Relational Algebra.

Module- IV: SQL

06 Hrs.

Concept of DDL, DML. Basic Structure Relational databases and tables, Set operations, Aggregate Functions, Null Values, Domain Constraints, Referential Integrity Constraints, assertions, views, Nested Subqueries.

Module- V: Relational Model and Relational Database Design

09 Hrs.

Concept of Relational Model, Design Issues, Keys, Closure set, Functional Dependency, Different anomalies in designing a Database., Normalization using functional dependencies, Decomposition, 3NF, Boyce-Codd Normal Form.

Module- VI: Transaction Management

09 Hrs.

ACID properties, Transaction definition, properties, transaction state diagram, commit and rollback, Serializability (Conflict and View), Concurrency control, lock based protocols, Two phase locking, Timestamp ordering protocol, Recovery management, Deadlock handling and prevention.

Module- VII: File Structure and Indexing

09 Hrs.

Operations on files, File of Unordered and ordered records, overview of File organizations, indexing structures for files (Primary index, secondary index, clustering index), Multilevel indexing using B and B + trees.

Reference books:

1. R. Elmasri, S.B. Navathe, Fundamentals of Database Systems 6th Edition, Pearson Education, 2010.
2. R. Ramakrishanan, J. Gehrke, Database Management Systems 3rd Edition, McGraw-Hill, 2002.
3. A. Silberschatz, H.F. Korth, S. Sudarshan, DatabaseSystem Concepts 6th Edition, McGraw Hill, 2010.
4. R. Elmasri, S.B. Navathe Database Systems Models, Languages, Design and application Programming, 6th Edition, Pearson Education, 2013.

MJ-9P: Database Management Systems (Practical)

Credit 01

1. Create the following tables, enter at least 5 records in each table and answer the queries given below.

Item (Item_Code, Item_Name, Price)

Purchase (Item_Code, Quantity, Purchase_date)

Select appropriate primary keys and foreign keys. Select appropriate data types for all the fields.

- a) Create the tables with the above attributes and enter 5-7 tuples into each table.

- b) Display the List of all items with their price which have minimum 10 quantities.
- c) Display the List of items which are not purchased by anyone.
- d) Display all the item with their Purchase date.
- e) Display all items according to their price.

2. Create the following tables with the fields given below:

Teacher (T_ID, Department, Year of Exp, Name)

Subject (Sub_Paper_ID, T_ID, Title_of Paper, Programme, Semester)

Select appropriate primary keys and foreign keys. Select appropriate data types for all the fields.

- a) Create the tables with the above attributes and enter 5-7 tuples into each table.
- b) Display Name and Year-Of-Exp of all the teachers of "Computer Science" department.
- c) List the Subject Paper which are handled by T_ID = 101.
- d) List the name of Programme and Semester of paper titled "C Programming".
- e) List the name of teacher who are allotted for 4th semester.

3. Create the following tables with the fields given below:

Product (Product_ID, Supplier_ID, Proudct_Name, Category, Price, Quantity)

Supplier (Supplier_ID, S_Name, Area, City)

Select appropriate primary keys and foreign keys. Select appropriate data types for all the fields.

- a) Create the tables with the above attributes and enter 5-7 tuples into each table.
- b) List price and product name of all the products whose Quantity is more than 100.
- c) Display product name and product ID of the products which are supplied by Supplier_ID = 10111.
- d) Display the number of products which price is more than Rs. 1,500 and Quantity is more than 5.
- e) Display name and area of all the suppliers of city "Kolkata".

4. Create the following tables with the fields given below:

Doctor (Did, DName, Dept_name, Salary, Joining_date)

Patient (Pid, Did, Pname, P_address, Phone_no, Admitted_dept)

Select appropriate primary keys and foreign keys. Select appropriate data types for all the fields.

- a) Create the tables with the above attributes and enter 5-7 tuples into each table.
- b) Find the list of patients admitted in "Orthopedic" department.
- c) Display the name, Department and Salary of doctor who get maximum salary.
- d) Display the total number of patients admitted under "Dr. M Chatterjee".
- e) List the number of patients in each Department.

5. Create the following tables with the fields given below:

Student (Student ID, Name, Programme, Teacher ID)

Teacher (Teacher ID, Department, Name, City, Specialization)

Select appropriate primary keys and foreign keys. Select appropriate data types for all the fields.

- a) Create the tables with the above attributes and enter 5-7 tuples into each table.
- b) Display total number of students registered in "MCA" programme.
- c) Display name of all the students who are associated with TeacherID = 123.
- d) Display name, specialization and city of all the teachers who are associated with "Compute Science" department.
- e) Display unique Department from the Teacher Table.

6. Create the following tables with the fields given below:

Department (Dept_no, D_name, loc)

Employee (Empno, E_name, job, mgr, hiredate, sal, comm, Dept_no)

Select appropriate primary keys and foreign keys. Select appropriate data types for all the fields.

- Create the tables with the above attributes and enter 5-7 tuples into each table.
- Update the employee salary by 15%, whose experience is greater than 10 years.
- Display the manager who is having maximum number of employees working under him.
- Display Name of all the employees where the third letter of their name is 'A'.
- Display Unique Listing of all Jobs that are in Department no 30.

7. Create the following tables with the fields given below:

Customer (Cust ID, email, Name, Phone, Referrer ID)

Bicycle (Bicycle ID, Date Purchased, Color, Cust ID, Model No)

Bicycle Model (Model No, Manufacturer, Style) **Service** (Start Date, Bicycle ID, End Date)

Select appropriate primary keys and foreign keys and select appropriate data types for all the fields.

- Identify primary and foreign keys. Create the tables and insert at least 5 records in each table.
- List all the customers who have the bicycles manufactured by manufacturer Honda.
- List the bicycles purchased by the customers who have been referred by customer C11.
- List the manufacturer of red colored bicycles.
- List the models of the bicycles given for service.

8. Create the following tables, enter at least 5 records in each table and answer the queries given below.

Employee (Person_Name, Street, City)

Works (Person_Name, Company_Name, Salary)

Company (Company_Name, City)

Manages (Person_Name, Manager_Name)

- Identify primary and foreign keys.
- Alter table employee, add a column —email of type varchar (20).
- Find the name of all managers who work for both Samba Bank and NCB Bank.
- Find the names, street address and cities of residence and salary of all employees who work for Samba Bank and earn more than \$10,000.
- Find the names of all employees who live in the same city as the company for which they work.
- Find the highest salary, lowest salary and average salary paid by each company.
- Find the sum of salary and number of employees in each company.
- Find the name of the company that pays highest salary.

9. Create the following tables, enter at least 5 records in each table and answer the queries given below.

SUPPLIERS (SNo, Sname, Status, SCity)

PARTS (PNo, Pname, Colour, Weight, City)

PROJECT (JNo, Jname, Jcity)

SHIPMENT (Sno, Pno, Jno, Qunatity)

- Identify primary and foreign keys.
- Get supplier numbers for suppliers in Paris with status>20.
- Get suppliers details for suppliers who supply part P2. Display the supplier list in increasing order of supplier numbers.

- d) Get suppliers names for suppliers who do not supply part P2.
- e) For each shipment get full shipment details, including total shipment weights.
- f) Get all the shipments where the quantity is in the range 300 to 750 inclusive.
- g) Get part nos. for parts that either weigh more than 16 pounds or are supplied by suppliers S2, or both.
- h) Get the names of cities that store more than five red parts.
- i) Get full details of parts supplied by a supplier in London.
- j) Get part numbers for part supplied by a supplier in London to a project in London.
- k) Get the total number of projects supplied by a supplier (say, S1).
- l) Get the total quantity of a part (say, P1) supplied by a supplier (say, S1).

10. Create a database having three tables to store the details of students of Computer Department in your college.

Personal information about Student (College roll number, Name of student, Date of birth, Address, Marks (rounded off to whole number) in percentage at 10 + 2, Phone number)

Paper Details (Paper code, Name of the Paper)

Student's Academic and Attendance details (College roll number, Paper code, Attendance, Marks in home examination).

- a) Identify primary and foreign keys. Create the tables and insert at least 5 records in each table.
- b) Design a query that will return the records (from the second table) along with the name of student from the first table, related to students who have more than 75% attendance and
- c) more than 60% marks in paper 2.
- d) List all students who live in —Delhil and have marks greater than 60 in paper 1.
- e) Find the total attendance and total marks obtained by each student.
- f) List the name of student who has got the highest marks in paper 2.

UG/V/COMP/4/MJ-10: Discrete Structure

Credit 04

Objectives of the Course

- Cover fundamental topics such as logic, set theory, combinatorics, graph theory, and discrete probability.
- Teach students to apply logical reasoning and proof techniques, including induction, to solve complex problems.
- Emphasize the use of mathematical structures to model and analyze real-world situations.
- Develop skills in mathematical reasoning and problem-solving through practical exercises.
- Equip students with analytical tools for advanced studies in computer science, engineering, and mathematics.
- Prepare students for roles in research and development.

Course Content:

Module – I

10 Hrs.

Mathematical Logic: Propositional Logic, Applications of Propositional Logic, Propositional Equivalences; Predicates and Quantifiers, Nested Quantifiers, Rules of Inference; Introduction to Proofs, Proof Methods and Strategy.

Module – II**12 Hrs.**

Basic Structures: Set Theory: Sets, Combination of Sets and Set Cardinality. Functions: Injections, Surjections, Bijections, Inverse and Composition of functions. Relations: Binary relations, properties and equivalence relations. Partial Orderings: Partially Ordered Sets, Hasse diagrams, and Lattices. Chains and antichains.

Module – III**08 Hrs.**

Permutations, Combinations: Rules of sum and product; Permutations; Combinations; Generation of permutations and combinations; Combination with repetitions. Binomial theorem; Multinomial theorem. Derangements. Principle of Inclusion and Exclusion. Pigeon-hole principle.

Module – IV**06 Hrs.**

Generating functions: Definition, operations, applications to counting. Recurrence relations: Linear recurrence relations with constant coefficients; Homogeneous solutions; Particular solutions; Total Solutions; Solution by the method of generating functions.

Module – V**16 Hrs.**

Graphs Theory: Basic Terminology and Representation; Subgraphs; Graph Isomorphism; Bipartite Graph; Multigraphs and weighted graphs; Paths and circuits; Shortest paths in weighted graphs; Eulerian paths and circuits; Hamiltonian paths and circuits; The traveling salesman problem; Factors of a graph; Planar graphs. Graph colouring.

Module – VI**08 Hrs.**

Trees and Cut-sets: Trees; Rooted trees; Path lengths in rooted trees; Prefix codes; Binary search trees; Spanning trees and cut-sets; Minimum spanning trees.

Reference books:

1. Discrete Mathematics and Its Applications by Kenneth Rosen (TMH)
2. Discrete and Combinatorial Mathematics by Ralph P. Grimaldi (Pearson)
3. Discrete Mathematical Structures with Applications to Computer Science by J.P. Tremblay and R. Manohar (TMH)
4. Graph Theory with Applications To Engineering And Computer Science by Narsingh Deo (PHI)
5. Introduction to Graph Theory by Douglas B. West (Pearson)

Major Elective (DSE-1)**DSE-1: Programming in Java****Credit 04****Objective of the course:**

- Understand Java environment, JVM architecture, features, and differences from C++, and develop basic programs using variables, data types, operators, control structures, and methods.
- Apply concepts of arrays, strings, and input/output streams to develop programs for data handling, manipulation, and file operations.
- Demonstrate object-oriented programming concepts by designing and implementing classes, constructors, methods, and memory management using garbage collection.
- Implement advanced OOP features such as inheritance, interfaces, packages, polymorphism, and use

wrapper classes, enumerations, and metadata effectively.

- Develop robust and efficient Java applications using exception handling, multithreading.

Course Content

Module-1: Introduction to Java

10 Hrs.

Java Environment: Features, Editions, Architecture, JVM, JDK, JRE, Understanding the semantic and syntax differences between C++ and Java, Compiling and Executing a Java Program, Command Line Arguments, Constants, Variables, Data Types: Constants, Variables, Data Types, Declaration of Variables, Giving values to Variables, Symbolic Constants, Type casting. Operators & Expressions: Arithmetic operators, Relational operators, Logical operators, Assignment operators, Increment & Decrement operators, conditional operators, Bitwise Operators, Arithmetic Expressions, Evaluation of Expressions, Type Conversions in Expressions, Operator Precedence & Associativity. Decision Making, Branching, Looping: Decision Making Constructs (conditional statements and loops) and Nesting, Java Methods (Defining Scope, Passing and Returning Arguments, Built-in Java Class Methods).

Module-2: Object-Oriented Programming Overview

12 Hrs.

Object-Oriented Programming concepts: Principles and features of Object-Oriented Programming, Class, Objects, Encapsulation, Polymorphism, Inheritance, Abstraction. Class and objects: Define and Using Class and objects, Controlling Access to Class Members (Variables & Methods), Application of two classes. Methods: Method declaration, Overloading, Use of Constructors, Final keyword, static keyword, this keyword, Abstract keyword and Garbage Collection.

Module-3: Inheritance, Interfaces, Packages, Wrapper Class

12 Hrs.

Inheritance: Inheritance (Single Level, Multilevel and Hierarchical), Method Overriding, Dynamic Method Dispatch, Abstract Classes. Interfaces: Defining Interface, Extending and Implementing Interface, Multiple Inheritance, Abstraction. Packages: Basics of packages, System packages (util, lang, io, net), Creating and accessing packages, creating user defined packages, Adding class to a package. Wrapper Classes: Wrapper Classes, Autoboxing/Unboxing, Enumerations and Metadata.

Module-4: Arrays, Array List and Strings

08 Hrs.

Arrays: Creating & Using Arrays (One Dimension and Multi-dimensional), Referencing Arrays Dynamically. Array List: Creating and Accessing List, Array List Class, List Interface. String: The Java String class, Creating & Using String Objects, Manipulating Strings, String Immutability & Equality, Passing Strings To & From Methods, String Buffer Classes, String Builder Classes.

Module-5: Exception Handling, Multithreading

08 Hrs.

Exception Handling: Using the main keywords of exception handling: try, catch, throw, throws and finally; Nested try, Multiple catch statements, Creating user defined exceptions. Multithreading: The Thread class and Runnable interface, creating single and multiple threads, Thread prioritization, synchronization and communication, suspending/resuming threads.

Reference books:

1. Ken Arnold, James Gosling, David Homes, "The Java Programming Language", 4th Edition, 2005.
2. James Gosling, Bill Joy, Guy L Steele Jr, Gilad Bracha, Alex Buckley "The Java Language Specification, Java SE 8 Edition (Java Series)", Published by Addison Wesley, 2014.
3. Joshua Bloch, "Effective Java", 2nd Edition, Publisher: Addison-Wesley, 2008.
4. Cay S. Horstmann, Gary Cornell, "Core Java 2 Volume 1 ,9th Edition, Printice Hall.2012
5. Cay S. Horstmann, Gary Cornell, "Core Java 2 Volume 2 - Advanced Features)", 9th Edition, Printice Hall.2013
6. Bruce Eckel, "Thinking in Java", 3rd Edition, PHI, 2002.
7. E. Balaguruswamy, "Programming with Java", 4th Edition, McGraw Hill.2009.
8. Paul Deitel, Harvey Deitel, "Java: How to Program", 10th Edition, Prentice Hall, 2011 "Head First Java", Orielly Media Inc. 2nd Edition, 2005.

10. David J. Eck, "Introduction to Programming Using Java", Published by CreateSpace Independent Publishing Platform, 2009.
11. John R. Hubbard, "Programming with JAVA", Schaum's Series, 2nd Edition, 2004.

DSE-1: Programming in Java Lab (Practical)

Credits 01

1. Write a program to find the sum of two integers using command line arguments.
2. Write a program to swap two numbers as input from the keyboard.

3. Write a program to find the factorial of a given number.
4. Write a program to print first n Fibonacci numbers.
5. Write a program to check a number is prime or not.
6. Write a program to find sum of prime factors of given number
7. Write a program to find the reverse of a given number and check it is palindrome number or not.
8. Write a program to print Armstrong number between a given range entered from keyboard.
9. Program to demonstrate method overloading by creating multiple add() methods with different parameter types and arguments to perform addition and show compile-time polymorphism.
10. Program to demonstrate constructor overloading by defining multiple constructors in a class to initialize objects with different values.
11. Program to demonstrate static method by creating a utility method (like finding square or cube) that can be called without creating an object.
12. Program to demonstrate single inheritance by creating a base class (e.g., Animal) and a derived class (Dog) to reuse properties and methods.
13. Program to demonstrate multilevel inheritance using three classes (Grandparent, Parent, Child) to show inheritance across multiple levels.
14. Program to demonstrate hierarchical inheritance where multiple child classes inherit from a single parent class and use its properties.
15. Program to demonstrate method overriding by redefining a method in the child class to provide its specific implementation (runtime polymorphism).
16. Program to demonstrate use of super keyword to access parent class constructor and methods from the child class.
17. Program to demonstrate both method overloading and method overriding together to show compile-time and runtime polymorphism in a single application.
18. Program to demonstrate use of package by creating a user-defined package, defining a class inside it, and accessing that class from another program using import statement to show modular programming and code reusability.
19. Program to find the sum and average of elements in an array by taking input values and performing arithmetic operations.
20. Program to check whether a given string is a palindrome by comparing it with its reverse.
21. Program to demonstrate basic string functions like length (), toUpperCase(), toLowerCase(), and trim(), append(), insert(), concat() and equals() on a given string.
22. Program to compare two strings using equals() and compareTo() methods to check equality and lexical order.
23. Program to search a character or substring in a string using indexOf() and lastIndexOf() methods.
24. Program to demonstrate exception handling using try-catch-finally block by handling arithmetic exception (like division by zero) and ensuring program execution continues normally.
25. Program to demonstrate user-defined exception by creating a custom exception class and throwing it using throw and handling it using catch.
26. Program to demonstrate multithreading by creating threads using Thread class and Runnable interface to execute multiple tasks simultaneously.

Objective of the course:

- This course aims to provide students with a comprehensive understanding of Kotlin programming language and its application in modern software development.
- It introduces fundamental programming concepts along with Kotlin-specific features such as null safety, functional programming, object-oriented design, and Android development basics.
- Students will learn how to write clean, concise, and efficient code using Kotlin, and develop applications ranging from console-based programs to introductory mobile applications.
- By the end of the course, students will be able to design, develop, and test Kotlin-based applications and understand current trends in cross-platform and mobile development.

Course content**Module-I: Introduction to Kotlin****10 Hrs.**

Kotlin and its features, Setting up Kotlin environment, Kotlin vs Java, Basic syntaxes and structures, Variables and data types, Input and output operations.

Module-II: Control Structures and Functions**08 Hrs.**

Branching and Conditional statements (if, when), Looping constructs (for, while, dowhile), Functions and recursion, Default and named arguments, Lambda expressions, Higher-order functions.

Module-III: Object Oriented Programming in Kotlin**08 Hrs.**

Classes and objects, Constructors (primary and secondary), Properties and methods, Inheritance and interfaces, Abstract classes, Data classes, Sealed classes, Visibility modifiers, Other important OOP concepts.

Module-IV: Advanced Features and their Usage**12 Hrs.**

Null safety and safe calls, Collections framework (List, Set, Map), Extensibility and Extension functions, Exception handling with types of exceptions, Coroutines and asynchronous programming concepts, File handling basics and operations.

Module-V: Android Application Development using Kotlin**12 Hrs.**

Introduction to Android development with Kotlin, Activity lifecycle, Basic and Advanced UI components, Event handling and organizing events, Introduction to Jetpack libraries, Overview of Kotlin Multiplatform, Threading and Multiprogramming.

Reference books:

1. Dawn Griffiths & David Griffiths, Head First Kotlin, O'Reilly Publications
2. Bruce Eckel & Svetlana Isakova, Atomic Kotlin – MindView LLC
3. Hardik Trivedi, Android Application Development with Kotlin, BPB Publications
4. Andrew Bailey, David Greenhalgh & Josh Skeen, Kotlin Programming, AddisonWesley Professional.

DSE-1: Kotlin Programming (Practical)**Credit 1**

1. Write a Kotlin program to print "Hello, World!"
2. Write a Kotlin program to perform menu driven arithmetic operations.
3. Write a Kotlin program to demonstrate type conversion.
4. Write a Kotlin program using conditional statements to determine a number is positive even or negative-odd or neither.

5. Write a Kotlin program to implement loops to generate Pascal's triangle of nth level.
6. Write a Kotlin program with default and named arguments to implement generic function.
7. Write a Kotlin program using lambda expressions to show inline features.
8. Create and use classes and objects using Kotlin program.
9. Demonstrate inheritance and method overriding using Kotlin program.
10. Write a Kotlin program using data classes and sealed classes.
11. Write a Kotlin program to implement collections for a group or set of data items/records.
12. Write a Kotlin program demonstrating null safety and prioritization.
13. Write a Kotlin program using exception handling and customize it.
14. Demonstrate a File handling program in Kotlin to copy and paste the textual contexts.
15. Write a Kotlin program to implement higher-order functions and extension functions.
16. Write a Kotlin program to demonstrate simple coroutine example.
17. Develop a basic Android UI screen with username and password using Kotlin.
18. Make a Handle button click event for Color palette in Android using Kotlin for both foreground and background colors to opt for interface.
19. Develop a message passing chatbox using Kotlin for two or more users. 20. Create a simple Kotlin-based mobile application for Block game.

DSE-1: MATLAB Programming (T)

Credit 03

Objectives of the Course

- To introduce the basic concepts of programming and the fundamental components of a computer system.
- To familiarize students with the MATLAB programming environment, including variables, expressions, arrays, and basic program execution.
- To develop the ability to visualize data using MATLAB plotting tools and built-in functions.
- To enable students to write modular programs using functions, scripts, and formatted input/output operations.
- To provide understanding of control structures such as conditional statements and looping constructs for problem solving.
- To develop skills in handling text data, including reading from and writing to text files, and performing operations like sorting and searching.
- To introduce the basics of graphical user interface (GUI) development for creating simple interactive MATLAB applications.

Course content

Module-I: Introduction to Programming:	06 Hrs.
Components of a computer, working with numbers, Machine code, Software hierarchy.	
Module-II: Programming Environment	06 Hrs.
MATLAB Windows, A First Program, Expressions, Constants, Variables and assignment statement, Arrays.	
Module-III: Graph Plots	06 Hrs.
Basic plotting, built in functions, generating waveforms, Sound replay, load and save.	
Module-IV: Procedures and Functions	10 Hrs.
Arguments and return values, M-files, Formatted console input-output, String handling.	
Module-V: Control Statements	06 Hrs.
Conditional statements: If, Else, Else-if, Repetition statements: While, for loop.	

Module-VI: Manipulating Text**06 Hrs.**

Writing to a text file, reading from a text file, Randomizing and sorting a list, searching a list.

Module-VI: GUI Interface**10 Hrs.**

Attaching buttons to actions, Getting Input, Setting Output.

Reference books:

1. MATLAB: An Introduction with Applications, by Amos Gilat, 2nd edition, Wiley, 2004,
2. C.B. Moler, Numerical Computing with MATLAB, SIAM, 2004.

MJ DSE-1P: MATLAB Programming (Practical)**Credit 01**

1. Write a program to assign the following expressions to a variable A and then to print out the value of A.
 - a) $(3+4)/(5+6)$
 - b) $2 \pi 2$
 - c) $\sqrt{2}$
 - d) $(0.0000123 + 5.67 \times 10^{-3}) \times 0.4567 \times 10^{-4}$
2. Celsius temperatures can be converted to Fahrenheit by multiplying by 9, dividing by 5, and adding 32. Assign a variable called C the value 37, and implement this formula to assign a variable F the Fahrenheit equivalent of 37 Celsius.
3. Set up a vector called N with five elements having the values: 1, 2, 3, 4, 5. Using N, create assignment statements for a vector X which will result in X having these values:
 - a) 2, 4, 6, 8, 10
 - b) $1/2, 1, 3/2, 2, 5/2$
 - c) $1, 1/2, 1/3, 1/4, 1/5$
 - d) $1, 1/4, 1/9, 1/16, 1/25$
4. A supermarket conveyor belt holds an array of groceries. The price of each product (in pounds) is [0.6, 1.2, 0.5, 1.3]; while the numbers of each product are [3, 2, 1, 5]. Use MATLAB to calculate the total bill.
5. The `sortrows(x)` function will sort a vector or matrix X into increasing row order. Use this function to sort a list of names into alphabetical order.
6. The identity matrix is a square matrix that has ones on the diagonal and zeros elsewhere. You can generate one with the `eye()` function in MATLAB. Use MATLAB to find a matrix B, such that when multiplied by matrix $A = \begin{bmatrix} 1 & 2 \\ -1 & 0 \end{bmatrix}$ the identity matrix $I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ is generated. That is $A * B = I$.
7. Create an array of N numbers. Now find a single MATLAB statement that picks out from that array the 1, 4, 9, 16, ..., \sqrt{N} th entries, i.e. those numbers which have indices that are square numbers.
8. Draw a graph that joins the points (0,1), (4,3), (2,0) and (5,-2).
9. Calculate 10 approximate points from the function $y=2x$ by using the formulae:

$$x_n = n$$

$$y_n = 2n + \text{rand} - 0.5$$
10. Fit a line of best fit to these points using the function `polyfit()` with `degree=1`, and generate co-

ordinates from the line of best fit using polyval(). Use the on-line help to find out how to use these functions. Plot the raw data and the line of best fit.

11. Calculate and replay 1 second of a sinewave at 500Hz with a sampling rate of 11025Hz. Save the sound to a file called "ex35.wav". Plot the first 100 samples.

12. Calculate and replay a 2 second chirp. That is, a sinusoid that steadily increases in frequency with time, from say 250Hz at the start to 1000Hz at the end.

13. Build a square wave by adding together 10 odd harmonics: 1f, 3f, 5f, etc. The amplitude of the nth harmonic should be 1/n. Display a graph of one cycle of the result superimposed on the individual harmonics.

14. Write a function called FtoC (ftoc.m) to convert Fahrenheit temperatures into Celsius. Make sure the program has a title comment and a help page.

15. Write a program to input 2 strings from the user and to print out (i) the concatenation of the two strings with a space between them, (ii) a line of asterisks the same length as the concatenated strings, and (iii) the reversed concatenation.

For example:

Enter string 1: Mark

Enter string 2: Huckvale

Mark Huckvale

MINOR

UG/V/COMP/4/MI-5: Operating System

Credit 04

Objective of the course

- Provide a comprehensive understanding of fundamental concepts and functions of modern operating systems.
- Cover the architecture and components of operating systems, including process management, memory management, file systems, and input/output systems.
- Teach about concurrency, process synchronization, and inter-process communication for efficient task management.
- Emphasize the role of operating systems in resource allocation and system security.
- Offer hands-on experience in implementing and configuring operating system features through labs and projects.
- Explore various operating systems like Windows, Linux, and macOS to understand their differences and commonalities.

MI-5T: Operating System (T)

Credit 03

Course Content

Module I: Introduction

10 Hrs.

Basic OS functions, resource abstraction, types of operating systems—multiprogramming systems, batch systems, time sharing systems; operating systems for personal computers & workstations, process control & real time systems.

Module II: Operating System Organization

06 Hrs.

Processor and user modes, kernels, system calls and system programs.

Module III: Process Management

16 Hrs.

System view of the process and resources, process abstraction, process hierarchy, threads, threading issues, thread libraries; Process Scheduling, non-pre-emptive and pre-emptive scheduling algorithms; concurrent processes, critical section, semaphores, methods for inter- process communication, Deadlocks: Detection, prevention, avoidance (Banker's Algorithm).

Module IV: Memory Management

10 Hrs.

Memory hierarchy and memory allocation strategies, Paging and Segmentation, Virtual Memory, Page replacement algorithm (FIFO, LRU, Optimal), Thrashing and working set model.

Module V: File Systems and

04 Hrs.

File concepts, access methods, and directory structures, File allocation methods: Contiguous, Linked, and Indexed.

Module VI: Storage Management

04 Hrs.

Disk scheduling algorithms: FCFS, SSTF, SCAN, C-SCAN, RAID levels and storage hierarchy.

Reference books:

1. A Silberschatz, P.B. Galvin, G. Gagne, Operating Systems Concepts, 8th Edition, John Wiley Publications 2008.
2. A.S. Tanenbaum, Modern Operating Systems, 3rd Edition, Pearson Education 2007.
3. G. Nutt, Operating Systems: A Modern Perspective, 2nd Edition Pearson Education 1997.
4. W. Stallings, Operating Systems, Internals & Design Principles, 5th Edition, Prentice Hall of India. 2008.
5. M. Milenkovic, Operating Systems- Concepts and design, Tata McGraw Hill 1992.

UG/IV/COMP/4/MI-5P:

Credit 01

1. Implement a simple fork() program.
2. Implement number of times hello is printed.
3. Implement a program where parent and child execute.
4. Implement a FCFS program where process arrival times are same.
5. Implement a FCFS program where process arrival times are different.
6. Write program to implement Round Robin scheduling algorithm.
7. Write program to implement SJF scheduling algorithm.

Semester VI

MAJOR

UG/VI/COMP/4/MJ-11: Theory of Computation:

Credit 04

Objective of the course:

On completing of the paper learners are expected to be able to-

- Analyze and construct Finite State Machines, regular grammar.
- Define and working with regular expressions.
- Designing context free grammars and pushdown automata.
- Know the capabilities and limitations of these models.
- Understand language hierarchy and analyze their properties.
- Designing Turing machines and apply Church-Turing thesis.
- Distinguish between decidable and undecidable languages.
- Identify and analyze complexity classes.

Course Contents:

Module I: Basic terminology

02 Hrs.

Concepts of Automata Theory: Alphabets, Strings, Languages and Problems.

Module II: Chomsky Classification of Languages

02 Hrs.

Formal languages: Introduction, type-0, type-1, type-2 and type-3 languages.

Module III: Finite Automata

10 Hrs.

Finite Automata: Deterministic finite automata; Nondeterministic finite automata; Equivalence of Deterministic finite automata and nondeterministic finite automata; Finite automata with epsilon-transitions; Elimination of epsilon-transitions. Mealy and Moore machine.

Module IV: Regular Grammar and Regular Expression

10 Hrs.

Regular sets and regular grammars: Regular expressions; Operators of regular expression; Finite automata and regular expressions – Converting deterministic finite automata to regular expression, Converting regular expression to finite automata, Algebraic laws for regular expressions; Pumping lemma for regular languages: Statement, proof and applications. Closure properties of regular languages.

Module V: Pushdown automata

10 Hrs.

Pushdown automata: Definition; Deterministic pushdown automata. Equivalence of pushdown automata and context free grammars; Non deterministic pushdown automata. Equivalence of deterministic and nondeterministic pushdown automata; CNF and GNF.

Module VI: Context free language, context free grammar and ambiguity in CFG

08 Hrs.

Context free languages: Context free grammars–Definition; Derivations using a grammar; Parse trees; Ambiguity in Grammars and Languages; Pumping lemma for context free languages: Statement and applications, no proof. Closure properties of Context free languages.

Module VII: Turing Machines ad Linear Bounded Automata

10 Hrs.

Turing machines and linear bounded automata: Definition; Context sensitive language and Context

sensitive grammar; Church – Turing thesis. Universal Turing Machine; Halting problem, PCP problem. Recursive and recursively enumerable language; Non-deterministic Turing machines.

Module VIII: Undecidability and Intractability

08 Hrs.

Computability: Undecidability and Intractability; The Classes P and NP; NP- Complete problems; NP-Completeness of SAT problem; NP-Completeness of 3SAT problem by polynomial-time reduction to SAT.

References:

1. Theory of Computer Science (Automata, Languages and Computation) by Mishra and Chandrasekharan (PHI) (required)
2. Introduction to Automata Theory, Languages and Computation by Hopcroft, Motwani and Ullman. (Pearson)(recommended)
3. Introduction to the Theory of Computation by Michael Sipser (CENGAGE)(optional)

UG/VI/COMP/4/MJ-12: Artificial Intelligence

Credit 04

Course Objective

- This course aims to provide students with a comprehensive understanding on Artificial Intelligence and its fundamental concepts, techniques, and applications of AI.
- In solving complex problems along with key topics such as machine learning, search algorithms, natural language processing, knowledge representation, and robotics.
- Students will learn how to design and implement intelligent systems that can perform tasks typically requiring human intelligence, such as reasoning, decision-making, and pattern recognition.
- By the end of the course, students will be able to apply AI algorithms to real-world problems, evaluate their performance, and explore emerging trends in AI technologies, preparing them for advanced studies and careers in the field.

UG/V/COMP/4/MJ-12T: Artificial Intelligence

Module I: Introduction to AI

06 Hrs.

Introduction to Artificial Intelligence, Background and Applications, Turing Test and Rational Agent approaches to AI, Introduction to Intelligent Agents, their structure, behavior and environment.

Module II: Problem Solving and Searching Techniques

12 Hrs.

Problem Characteristics, Production Systems, Control Strategies, Breadth First Search, Depth First Search, Hill climbing and its Variations, Heuristics Search Techniques: Best First Search, A* algorithm, Constraint Satisfaction Problem, Means-End Analysis,

Introduction to Game Playing, Concepts of Game Theory with AI, Min-Max and Alpha-Beta pruning algorithms.

Module III: Knowledge Representation

10 Hrs.

Introduction to First Order Predicate Logic, Resolution Principle, Unification, Semantic Nets, Conceptual Dependencies, Frames, and Scripts, Production Rules, Conceptual Graphs, Programming in Logic (PROLOG).

Module IV: Dealing with Uncertainties and Inconsistencies

04 Hrs.

Truth Maintenance System, Default Reasoning, Probabilistic Reasoning, Bayesian Probabilistic Inference, Possible World Representations.

Module V: Understanding Natural Languages

08 Hrs.

Introduction to Grammars and Languages, Chomsky Classifications, Parsing Techniques, Processing of Natural Languages, Context-Sensitive, Context-Free and Transformational Grammars, Recursive and Augmented Transitions.

Reference Books:

1. Russell & Norvig, Artificial Intelligence - A Modern Approach, Pearson Prentice Hall
2. Rich & Knight, Artificial Intelligence – Tata McGraw Hill
3. John Mason, Using Artificial Intelligence – ACS
4. Hitesh Motwani, Generative AI 3600 – A practical approach, Zebra Learn

UG/V/COMP/4/MJ-12P: Artificial Intelligence Practical

1. Write a Prolog program to calculate the sum of two numbers.
2. Write a Prolog program to find the maximum of two numbers.
3. Write a Prolog program to calculate the factorial of a given number.
4. Write a Prolog program to calculate the nth Fibonacci number.
5. Write a Prolog program, insert `_nth(item, n, into_list, result)` that asserts that result is the list into `_list` with item inserted as the nth element into every list at all levels.
6. Write a Prolog program to remove the `_nth` item from a list.
7. Write a Prolog program, remove `_nth(Before, After)` that asserts the After list is the Before list with the removal of every `_nth` item from every list at all levels.
8. Write a Prolog program to implement append for two lists.
9. Write a Prolog program to implement palindrome(List).
10. Write a Prolog program to implement `max(X,Y,Max)` so that Max is the greater of two numbers X and Y.
11. Write a Prolog program to implement `maxlist(List,Max)` so that Max is the greatest number in the list of numbers List.
12. Write a Prolog program to implement `sumlist(List,Sum)` so that Sum is the sum of a given list of numbers List.
13. Write a Prolog program to implement two predicates `evenlength(List)` and `oddlength(List)` so that they are true if their argument is a list of even or odd length respectively.
14. Write a Prolog program to implement `reverse(List,ReversedList)` that reverses lists.
15. Write a Prolog program to implement `maxlist(List,Max)` so that Max is the greatest number in the list of numbers list using cut predicate.
16. Write a Prolog program to implement GCD of two numbers.
17. Write a Prolog program to implement A* algorithm.
18. Write a Prolog program to make a Tic-Tac-Toe game.
19. Write a Prolog program to demonstrate subject and predicate of a given whole context and generate answer as per the given subject or predicate.
20. Write a prolog program to implement Semantic Network/Frame Structures.

UG/VI/COMP/4/MJ-10: Computational Statistics

Credit 04

Course Objective

On completing the paper learners are expected to be able to:

- Understand and apply basic probability concepts
- Define and work with random variables
- Recognize and use standard probability distributions
- Analyze functions of random variables

- Apply statistical estimation techniques
- Interpret sampling distributions and tests of hypotheses
- Reinforce advanced probability foundations
- Characterize and analyze stochastic processes
- Explain the Markov property and Markov chains

Module I: Fundamentals of Probability:

06 Hrs

Basic Concepts: Random experiments, sample space, events, axiomatic probability; Conditional Probability and Independence: Multiplication rule, law of total probability, Bayes' theorem: interpretation and application

Module II: Random Variables and Distributions

10 Hrs

Discrete Random Variables: PMF, CDF, properties; Continuous Random Variables: PDF, CDF, properties; Important Distributions: Discrete: Bernoulli, Binomial, Poisson, Geometric; Continuous: Uniform, Exponential, Normal; Functions of Random Variables: Transformation techniques, Joint Distribution.

Module III: Expectation, Variance, and Moments

10 Hrs

Expectation: Definition, linearity, properties; Variance and Standard Deviation: Properties and interpretation; Covariance and Correlation: Definition, properties, interpretation; Moment Generating Functions (MGF); Inequalities: Markov's inequality, Chebyshev's inequality.

Module IV: Descriptive Statistics

06 Hrs

Central Limit Theorem, Measures of Central Tendency: Mean, median, mode, geometric mean, harmonic mean; Measures of Dispersion: Range, variance, standard deviation, coefficient of variation; Shape of Distributions: Skewness and kurtosis; Moments of a Distribution: First (mean), second (variance), third (skewness), fourth (kurtosis); Data visualization concepts: Histograms, box plots, bar charts.

Module V: Statistical Inference

10 Hrs

Population vs Sample; Point Estimation: Methods of moments, maximum likelihood estimation, unbiasedness, efficiency; Confidence Intervals: For mean, variance, proportion (normal, t, chi-square); Hypothesis Testing: Null & alternative hypothesis; Type I & II errors; Tests: z-test, t-test, chi-square test, F-test; p-value concept; Interpretation pitfalls.

Module VI: Regression and Correlation

08 Hrs

Correlation Analysis: Pearson and Spearman; Simple Linear Regression: Least squares, interpretation; Multiple Linear Regression: Assumptions, interpretation, limitations.

Module VI: Random Process and Markov Chains

10 Hrs

Stochastic Processes: Definitions, classification; Poisson Process: Definition, properties, applications; Markov Chains (Discrete Time): Transition matrix; Chapman-Kolmogorov equations; Classification of states: recurrent, transient, absorbing; Steady-state probabilities; Applications

UG/VI/COMP/4/MJ-10: Computational Statistics Lab. (Practical)

1. Write a program in python to simulate rolling a die and calculate probability distribution.
2. Write a program in python to generate random numbers and verify uniform distribution.
3. Write a program in python to generate and plot Binomial distribution.
4. Write a program in python to simulate Poisson distribution.
5. Write a program in python to plot normal distribution and compute the statistics.
6. Write a program in python to compute PDF and CDF of normal distribution.

7. Write a program in python to calculate variance and standard deviation.
8. Write a program in python to create histograms and frequency tables.
9. Write a program in python to perform linear regression and plot regression line.
10. Write a program in python for prediction using regression model.
11. Write a program in python to estimate population mean.
12. Write a program in python to simulate Central Limit Theorem.
13. Write a program in python to perform various tests like Z-test, T-test, ANOVA.

Reference:

1. A First Course in Probability by Sheldon M Ross.
2. Introduction to Probability and Statistics for Engineers and the Scientist by Sheldon M Ross.
3. Introduction to Probability Models by Sheldon M Ross.
4. Outline of Statistics – Goon , Gupt, Dasgupta
5. An Introduction To Statistical Learning with Applications in Python by Gareth James.
6. Think Stats: Exploratory Data Analysis, Third Edition (Grayscale Indian Edition)- by Allen B. Downey.

Major Elective (DSE-2)

UG/VI/COMP/4/MJ/DSE-2: Advance JAVA

Credit 04

Objectives of the course:

- Understand and apply client-side scripting using JavaScript to create dynamic and interactive web pages.
- Develop database-driven applications using JDBC by establishing connections, executing queries, and handling result sets.
- Design and implement server-side applications using Servlets to handle client requests and generate dynamic responses.
- Develop web applications using JSP by embedding Java code in HTML and using implicit objects.
- Apply JavaBeans concepts to create reusable software components and integrate them into JSP based applications.
- Integrate Java technologies (JDBC, JSP, Servlets, JavaBeans) to build complete dynamic web applications.

Course content

Unit-1: JAVA Basics

06 Hrs.

Use of Class, Objects, Constructors, Interface, Array, ArrayList, Strings

Unit-2: JavaScript

10 Hrs.

Introduction to JavaScript, variables and data types, operators, control statements, looping statements, functions, arrays, strings, objects, Document Object Model (DOM), event handling, form validation, error handling, JSON and AJAX, and timing functions such as setTimeout and setInterval.

Unit-3: JDBC

10 Hrs.

Introduction to JDBC, need for JDBC, architecture of JDBC, types of JDBC drivers, establishing database connection, loading drivers, connection interface, statement interface (Statement, PreparedStatement, CallableStatement), executing queries, handling result sets, transaction management, exception handling in JDBC, batch processing, metadata (DatabaseMetaData, ResultSetMetaData), and introduction to

connection pooling.

Unit-4: JSP

08 Hrs.

Introduction to JSP, advantages of JSP over Servlets, JSP architecture and life cycle, JSP syntax and structure, JSP directives (page, include, taglib), JSP scripting elements (declaration, scriptlet, expression), implicit objects, JSP standard actions (useBean, setProperty, getProperty, include, forward), handling forms in JSP, session tracking, error handling in JSP, and introduction to Expression Language (EL) and JSTL (JavaServer Pages Standard Tag Library), JSP Application Design with MVC.

Unit-5: Servlets

10 Hrs.

Introduction to Servlets, advantages of Servlets over CGI, Servlet architecture and life cycle, setting up servlet environment, handling client requests and responses, HttpServlet class, handling GET and POST methods, servlet configuration and context, session management (cookies, URL rewriting, HttpSession), servlet chaining, request dispatching (forward, include), and exception handling in servlets.

Unit-6: Java Beans

08 Hrs.

Introduction to JavaBeans, features and advantages of JavaBeans, JavaBean conventions (private properties, public getter and setter methods), creating and using JavaBeans, properties and methods of a bean, types of properties (simple, indexed, bound, constrained), bean life cycle, introspection, persistence, and using JavaBeans with JSP (useBean, setProperty, getProperty).

Reference books:

1. Santosh Kumar K., JDBC, Servlets and JSP Black Book, Dreamtech Press, Latest Revised Edition.
2. Cay S. Horstmann and Gary Cornell, Core Java, Volume II: Advanced Features, Pearson Education, Latest Edition.
3. M.T. Savaliya, Advanced Java, Dreamtech Press, Revised Edition, 2016.
4. Marty Hall and Larry Brown, Core Servlets and JavaServer Pages, Pearson Education, 2 nd Edition.
5. Sarika Agarwal and Vivek Gupta, Java for Web Development, Wiley India, Latest Edition.
6. Bruce Eckel, Thinking in Java, Pearson Education, 4 th Edition.

Advanced Java Lab (Practical)

Credits 01

1. Program to perform operations on an array such as finding sum, average, maximum and minimum elements using looping constructs.
2. Program to demonstrate interface by defining an interface with abstract methods and implementing it in a class to perform specific operations (e.g., area calculation).
3. Program to implement interface with multiple classes to show how different classes provide their own implementation of the same method (runtime polymorphism concept).
4. Program to validate a form using JavaScript by checking input fields like name, email, and password before submission.
5. Program to perform basic arithmetic operations using JavaScript and display the result dynamically on a web page.
6. Program to change the content and style of HTML elements using DOM manipulation methods.
7. Program to handle button click event using addEventListener and display a message or perform an action.
8. Program to create a simple image slider or slideshow using JavaScript and timing functions like setInterval.
9. Program to establish a database connection using JDBC by loading the driver, creating a connection, and displaying a success message.
10. Program to insert records into a database table using PreparedStatement to demonstrate parameterized queries.
11. Program to retrieve and display records from a database table using Statement and ResultSet.

12. Program to update records in a database table using JDBC and display the number of rows affected.
13. Program to delete records from a database table using JDBC based on a specific condition.
14. Program to demonstrate transaction management in JDBC using commit and rollback operations.
15. Program to retrieve database metadata and result set metadata using DatabaseMetaData and ResultSetMetaData interfaces to display database and table information.
16. Program to create a simple Servlet that handles client request and displays a message using doGet() method.
17. Program to demonstrate handling of form data in a Servlet by accepting user input and displaying the result dynamically.
18. Program to create a JSP page that displays dynamic content using JSP scripting elements and implicit objects.
19. Program to demonstrate session management using Servlets and JSP by tracking user login information using HttpSession.
20. Program to demonstrate request forwarding and redirection using Servlet (RequestDispatcher) and JSP pages.

OR

DSE2: Internet of thing (IoT)

Credits 04

Course contents:

Module – I: Fundamentals of IoT:

Introduction, Definitions & Characteristics of IoT, IoT Architectures, Physical & Logical Design of IoT, Enabling Technologies in IoT, History of IoT, About Things in IoT, The Identifiers in IoT, About the Internet in IoT, IoT frameworks, IoT and M2M.

Module – II: Sensors Networks:

Definition, Types of Sensors, Types of Actuators, Examples and Working, IoT Development Boards: Arduino IDE and Board Types, RaspberriPi Development Kit, RFID Principles and components, Wireless Sensor Networks: History and Context, The node, Connecting nodes, Networking Nodes, WSN and IoT.

Module – III: Data Handling & Analytics:

Introduction, Bigdata, Types of data, Characteristics of Bigdata, Data handling Technologies, Flow of data, Data acquisition, Data Storage, Introduction to Hadoop. Introduction to data Analytics, Types of Data analytics, Local Analytics, Cloud analytics and applications.

Module – IV: Applications of IoT:

Home Automation, Smart Cities, Energy, Retail Management, Logistics, Agriculture, Health and Lifestyle, Industrial IoT, Legal challenges, IoT design Ethics, IoT in Environmental Protection.

Reference books:

1. Hakima Chaouchi, — “The Internet of Things Connecting Objects to the Web” ISBN : 978-1- 84821-140-7, Wiley Publications
2. Olivier Hersent, David Boswarthick, and Omar Elloumi, — “The Internet of Things: Key Applications and Protocols”, WileyPublications
3. Vijay Madiseti and ArshdeepBahga, — “Internet of Things (A Hands-on-Approach)”, 1st Edition, VPT, 2014.
4. J. Biron and J. Follett, "Foundational Elements of an IoT Solution", O'Reilly Media, 2016.
5. Keysight Technologies, “The Internet of Things: Enabling Technologies and Solutions for Design and Test”, Application Note, 2016.

6. Daniel Minoli, — “Building the Internet of Things with IPv6 and MIPv6: The Evolving World of M2M Communications”, ISBN: 978-1-118-47347-4, Willy Publications
7. Pethuru Raj and Anupama C. Raman, "The Internet of Things: Enabling Technologies, Platforms, and Use Cases", CRC Press
8. https://onlinecourses.nptel.ac.in/noc17_cs22/course
9. http://www.cse.wustl.edu/~jain/cse570-15/ftp/iot_prot/index.html

UG/VI/COMP/4/MJ/DSE2: Practical of IoT (Arduino Lab)

Credits 01

1. Understanding Arduino UNO Board and Components
2. Installing and work with Arduino IDE
3. Blinking LED sketch with Arduino
4. Simulation of 4-Way Traffic Light with Arduino
5. Ultrasonic sensor handing with Arduino
6. Servo motor, stepper handing with Arduino
7. Different sensor handing related to smart home using Arduino.
8. Spinning a DC Motor and Motor Speed Control Sketch
9. Working with Shields
10. Interfacing Arduino with Cloud.

OR

DSE2: Cloud Computing

Credits 04

Course contents:

Module – I: Overview of Computing Paradigm:

8Hrs.

Recent trends in Computing: Grid Computing, Cluster Computing, Distributed Computing, Utility Computing, Cloud Computing.

Module – II: Introduction to Cloud Computing

8Hrs

Introduction to Cloud Computing, History of Cloud Computing, Cloud service providers, Benefits and limitations of Cloud Computing.

Module – III: Cloud Computing Architecture

12 Hrs

Comparison with traditional computing architecture (client/server), Services provided at various levels, Service Models- Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS), How Cloud Computing Works, Deployment Models- Public cloud, Private cloud, Hybrid cloud, Community cloud, Case study of NIST architecture.

Module – IV: Case Studies

8Hrs

Case study of Service model using Google App Engine, Microsoft Azure, Amazon EC2, Eucalyptus.

Module – V: Service Management in Cloud computing

12 Hrs

Service Level Agreements (SLAs), Billing & Accounting, Comparing Scaling Hardware: Traditional vs. Cloud, Economics of scaling.

Module – VI: Cloud Security

12 Hrs

Infrastructure Security- Network level security, Host level security, Application-level security, Data security and Storage Data privacy and security Issues, Jurisdictional issues raised by Data location, Authentication in cloud computing.

Reference books:

1. Cloud Computing: Principles and Paradigms, Editors: Rajkumar Buyya, James Broberg, Andrzej M. Goscinski, Wiley, 2011
2. Cloud Computing: Principles, Systems and Applications, Editors: Nikos Antonopoulos, Lee Gillam, Springer, 2012
3. Cloud Security: A Comprehensive Guide to Secure Cloud Computing, Ronald L. Krutz, Russell Dean Vines, Wiley-India, 2010.

DSE-2: Cloud Computing Lab (Practical)

Credit 01

1. Create virtual machines that access different programs on same platform.
2. Create virtual machines that access different programs on different platforms.
3. Working on tools used in cloud computing online -
 - a) Storage
 - b) Sharing of data
 - c) Manage your calendar, to-do lists,
 - d) A document editing tool
4. Exploring Google cloud.
5. Exploring Microsoft cloud.

MINOR

UG/VI/COMP/4/MI-C6: Database Management System

Credit 04

MI-C6T: Database Management Systems Theory

Credits 03

Module- I:Introduction

6 Hrs.

Concept & Overview of DBMS, Data Models, Database Languages, Database Administrator, Database Users, Data Abstraction, Three Schema architecture of DBMS.

Module- II: Entity Relationship (ER) Model

6 Hrs.

Entity Set, Simple and composite Attribute, Single valued and multivalued attribute, Relationship sets, Mapping cardinality, keys, , Entity Relationship Diagram : Need for E-R Model, Various steps of database design, Mapping Constraints, E-R diagram, Subclass, Generalization, Specialization, Aggregation, Strong Entity-Weak Entity.

Module- III: Relational Algebra

10 Hrs.

Select operation, Project Operation, Set operations (union, intersection, difference), Join operations, Division operation, outer join and outer union, Examples queries in Relational Algebra.

Module- IV: SQL

8 Hrs.

Concept of DDL, DML. Basic Structure Relational databases and tables, Set operations, Aggregate Functions, Null Values, Domain Constraints, Referential Integrity Constraints, assertions, views, Nested Sub queries.

Module- V: Relational Model and Relational Database Design

10 Hrs.

Concept of Relational Model, Design Issues, Keys, Closure set, Functional Dependency, Different anomalies in designing a Database., Normalization using functional dependencies, Decomposition, 3NF, Boyce-Codd Normal Form.

References:

1. R. Elmasri, S.B. Navathe, Fundamentals of Database Systems 6th Edition, Pearson Education, 2010.
2. R. Ramakrishanan, J. Gehrke, Database Management Systems 3rd Edition, McGraw-Hill, 2002.
3. A. Silberschatz, H.F. Korth, S. Sudarshan, DatabaseSystem Concepts 6th Edition, McGraw Hill, 2010.
4. R. Elmasri, S.B. Navathe Database Systems Models, Languages, Design and application Programming, 6th Edition, Pearson Education, 2013.

MI-C6P: Database Management Systems (Practical)

Credits 01

1. An inventory database has the following tables:

ITEM (Item_Code, Item_Name, Price)

PURCHASE (Item_Code, Quantity, Purchase_date)

Select appropriate primary keys and foreign keys. Select appropriate data types for all the fields.

- a. Create the tables with the above attributes and enter 5-7 tuples into each table.
- b. Display the List of all items with their price which have minimum 10 quantity.
- c. Display the List of items which are not purchased by anyone.
- d. Display all the item with their Purchase date.
- e. Display all items according to their price.

2. Create the following tables with the fields given below:

TEACHER (T_ID, Department, Year of Exp, Name)

SUBJECT (Sub_Paper_ID, T_ID, Title_of Paper, Programme, Semester)

Select appropriate primary keys and foreign keys. Select appropriate data types for all the fields.

- a. Create the tables with the above attributes and enter 5-7 tuples into each table.
- b. Display Name and Year-Of-Exp of all the teachers of "Computer Science" department.
- c. List the Subject Paper which are handled by T_ID = 101.
- d. List the name of Programme and Semester of paper titled "C Programming".
- e. List the name of teacher who are allotted for 4+ semester.

3. Create the following tables with the fields given below:

PRODUCT (Product_ID, Supplier_ID, Proudct_Name, Category, Price, Quantity)

SUPPLIER (Supplier_ID, S_Name, Area, City)

Select appropriate primary keys and foreign keys. Select appropriate data types for all the fields.

- a. Create the tables with the above attributes and enter 5-7 tuples into each table.
- b. List price and product name of all the products whose Quantity is more than 100.
- c. Display product name and product ID of the products which are supplied by Supplier_iD = 10111.
- d. Display the number of products which price is more than Rs. 1,500 and Quantity is more than 5.
- e. Display name and area of all the suppliers of city "Kolkata".

4. Create the following tables with the fields given below:

DOCTOR(Did, DName, Dept_name, Salary, Joining_date)

PATIENT(Pid, Did, Pname, P_address, Phone_no, Admitted_dept)

Select appropriate primary keys and foreign keys. Select appropriate data types for all the fields.

- a. Create the tables with the above attributes and enter 5-7 tuples into each table.
- b. Find the list of patients admitted in "Orthopedic" department.
- c. Display the name, Department and Salary of doctor who get maximum salary.
- d. Display the total number of patients admitted under "Dr. M Chatterjee".
- e. List the number of patients in each Department.

5. Create the following tables with the fields given below:

STUDENT (Student ID, Name, Programme, Teacher ID)

TEACHER (Teacher ID, Department, Name, City, Specialization)

Select appropriate primary keys and foreign keys. Select appropriate data types for all the fields.

- a. Create the tables with the above attributes and enter 5-7 tuples into each table.
- b. Display total number of students registered in "MCA" programme.
- c. Display name of all the students who are associated with TeacherID = 123.
- d. Display name, specialization and city of all the teachers who are associated with "Compute Science" department.
- e. Display unique Department from the Teacher Table.

6. Create the following tables with the fields given below:

Department (Dept_no, D_name, loc)

Employee (Empno, E_name, job, mgr, hiredate, sal, comm, Dept_no)

Select appropriate primary keys and foreign keys. Select appropriate data types for all the fields.

- a. Create the tables with the above attributes and enter 5-7 tuples into each table.
- b. Update the employee salary by 15%, whose experience is greater than 10 years.
- c. Display the manager who is having maximum number of employees working under him.
- d. Display Name of all the employees where the third letter of their name is 'A'.
- e. Display Unique Listing of all Jobs that are in Department no 30.