1. Single Inheritance

- Write a Java program where a class Person has attributes name and age. A derived class Student inherits Person and has an additional attribute studentID. Create an object of Student and display its details.

2. Multilevel Inheritance

- Write a Java program where a class Grandfather has an attribute familyName. A subclass Father inherits Grandfather and has an additional attribute profession. Another subclass Son inherits Father and has an additional attribute hobby. Create an object of Son and display all details.

3. Hierarchical Inheritance

- Write a Java program where a base class Vehicle has an attribute brand. Two subclasses, Car and Bike, inherit from Vehicle. The Car class has an attribute seat, and the Bike class has an attribute type. Create objects of both subclasses and display their details.

4. Hybrid Inheritance (Using a Common Base Class)

- Write a Java program where a class Employee has attributes name and salary. Two classes, Developer and Manager, inherit from Employee. Another class TeamLead inherits from both Developer and Manager using a **common base class** approach. Create an object of TeamLead and display all details.

5. Combination of Single and Multilevel Inheritance

- Write a Java program where a base class Animal has an attribute species. A class Mammal inherits Animal and has an attribute warmBlooded. Another class Dog inherits Mammal and has an attribute breed. Create an object of Dog and display all attributes.

Basic Class and Object

- Write a Java program to create a class named Car with attributes brand, model, and year. Create an object of this class, assign values, and display them.

Constructor Usage

- Write a Java program to create a class Student with instance variables name and age. Use a constructor to initialize the values and a method to display them.

Encapsulation

- Write a Java program to create a class BankAccount with private attributes accountNumber and balance. Provide getter and setter methods to access and update these attributes.

Constructor & Encapsulation

Car Details with Secure Data Handling

- Write a Java program to create a class Car with private attributes: brand, model, and price.
- Use a **constructor** to initialize these values.
- Provide getter methods to access brand and model, but allow modification of price using a setter method.
- Create an object of Car and update the price after creation.

➢ **Create** a class **Square** that has an attribute length. One default constructor that will set the length 0. One parameterized constructor that will receive an argument and set the length. The setLength and getLength method must there. Add another method getArea that will calculate and return area of the Square.

Problem 1: Employee Management System

- Create a class called Employee with **three attributes**: employeeID, name, and department.
- Implement a **parameterized constructor** to initialize these attributes.
- Provide **set and get methods** for all attributes.
- Create an **array of Employee class** with a size of 4.
- Use a loop that iterates 4 times to input employeeID, name, and department.
- Use setter methods within the loop to set the values for each Employee object.

Problem 2: Book Collection System

- Create a class called Book with **three attributes**: bookID, title, and author.
- Implement a **parameterized constructor** to initialize these attributes.
- Provide **set and get methods** for all attributes.
- Create an **array of Book class** with a size of 4.
- Use a loop that iterates 4 times to input bookID, title, and author.
- Use setter methods within the loop to set the values for each Book object.

Problem Statement:

Write a Java program with a class Numeric that provides utility methods for checking numbers. Implement:

1. isPositive(int n): Returns true if the number is positive or zero, otherwise false.
2. isEven(int n): Returns true if the number is even, otherwise false.

In the Main class, test these methods by calling them with different numbers and printing the results.

- ➢ Write a java program to calculate the factorial of a n number.

Problem: Usages of String functions below…

i) charAt(); ii) concat(); iii) contains(); iv) equals(); v) indexOf(); vi)isEmpty(); vii) length(); viii) replace(); ix) splits(); x) trim(); xi)toUpperCase(); xii) toLowerCase();