**Lab Exercise 5**

**Inheritance and Abstract Classes**

**1. Create a base class Employee with fields for name and salary. Derive a Manager class**

**that adds a department field. Write a program to display details of a manager.**

**2. Create a superclass Vehicle with methods for start and stop. Extend it into classes Car**

**and Bike with their own methods. Demonstrate method overriding.**

**3. Create a base class Shape with methods area() and perimeter(). Derive Circle and**

**Rectangle classes that override area() and perimeter() to calculate respective areas and perimeters.**

**4. Create a class BankAccount with fields account number and balance. Derive classes**

**SavingsAccount and CurrentAccount with additional functionalities like interest calculation a overdraft limit.**

**5. Create a Book class with title and author. Derive a Magazine class that adds issue number. Display details using inheritance.**

**6. Create an abstract class Animal with an abstract method makeSound(). Implement it in Dog and Cat classes.**

**7.Create an abstract class Shape with abstract methods calculateArea() and calculatePerimeter(). Implement in Triangle and Square classes.**

1. **Employee and Manager (Inheritance)**

class Employee {

String name;

double salary;

Employee(String name, double salary) {

this.name = name;

this.salary = salary;

}

}

class Manager extends Employee {

String department;

Manager(String name, double salary, String department) {

super(name, salary);

this.department = department;

}

void display() {

System.out.println("Name: " + name);

System.out.println("Salary: " + salary);

System.out.println("Department: " + department);

}

}

public class Main1 {

public static void main(String[] args) {

Manager m = new Manager("Alice", 50000, "IT");

m.display();

}

}

1. **Vehicle, Car, and Bike (Method Overriding)**

class Vehicle {

void start() {

System.out.println("Vehicle started");

}

void stop() {

System.out.println("Vehicle stopped");

}

}

class Car extends Vehicle {

void start() {

System.out.println("Car started");

}

void drive() {

System.out.println("Car is driving");

}

}

class Bike extends Vehicle {

void start() {

System.out.println("Bike started");

}

void ride() {

System.out.println("Bike is riding");

}

}

public class Main2 {

public static void main(String[] args) {

Car c = new Car();

c.start();

c.drive();

c.stop();

Bike b = new Bike();

b.start();

b.ride();

b.stop();

}

}

**3. Shape, Circle, Rectangle (Override area and perimeter)**

class Shape {

double area() { return 0; }

double perimeter() { return 0; }

}

class Circle extends Shape {

double radius;

Circle(double r) {

radius = r;

}

double area() {

return Math.PI \* radius \* radius;

}

double perimeter() {

return 2 \* Math.PI \* radius;

}

}

class Rectangle extends Shape {

double length, width;

Rectangle(double l, double w) {

length = l;

width = w;

}

double area() {

return length \* width;

}

double perimeter() {

return 2 \* (length + width);

}

}

public class Main3 {

public static void main(String[] args) {

Circle c = new Circle(5);

System.out.println("Circle Area: " + c.area());

System.out.println("Circle Perimeter: " + c.perimeter());

Rectangle r = new Rectangle(4, 6);

System.out.println("Rectangle Area: " + r.area());

System.out.println("Rectangle Perimeter: " + r.perimeter());

}

}

**4. BankAccount, SavingsAccount, CurrentAccount**

class BankAccount {

int accNo;

double balance;

BankAccount(int accNo, double balance) {

this.accNo = accNo;

this.balance = balance;

}

}

class SavingsAccount extends BankAccount {

double interestRate = 0.05;

SavingsAccount(int accNo, double balance) {

super(accNo, balance);

}

void calculateInterest() {

double interest = balance \* interestRate;

System.out.println("Interest: " + interest);

}

}

class CurrentAccount extends BankAccount {

double overdraftLimit = 1000;

CurrentAccount(int accNo, double balance) {

super(accNo, balance);

}

void checkOverdraft() {

System.out.println("Overdraft limit: " + overdraftLimit);

}

}

public class Main4 {

public static void main(String[] args) {

SavingsAccount sa = new SavingsAccount(101, 5000);

sa.calculateInterest();

CurrentAccount ca = new CurrentAccount(102, 3000);

ca.checkOverdraft();

}

}

**5. Book and Magazine**

class Book {

String title, author;

Book(String title, String author) {

this.title = title;

this.author = author;

}

}

class Magazine extends Book {

int issueNo;

Magazine(String title, String author, int issueNo) {

super(title, author);

this.issueNo = issueNo;

}

void show() {

System.out.println("Title: " + title);

System.out.println("Author: " + author);

System.out.println("Issue No: " + issueNo);

}

}

public class Main5 {

public static void main(String[] args) {

Magazine m = new Magazine("Tech Today", "John", 12);

m.show();

}

**6. Animal (Abstract Class)**

abstract class Animal {

abstract void makeSound();

}

class Dog extends Animal {

void makeSound() {

System.out.println("Dog barks");

}

}

class Cat extends Animal {

void makeSound() {

System.out.println("Cat meows");

}

}

public class Main6 {

public static void main(String[] args) {

Animal a1 = new Dog();

Animal a2 = new Cat();

a1.makeSound();

a2.makeSound();

}

}

**7. Shape (Abstract Class)**

abstract class Shape {

abstract double calculateArea();

abstract double calculatePerimeter();

}

class Triangle extends Shape {

double a, b, c;

Triangle(double a, double b, double c) {

this.a = a; this.b = b; this.c = c;

}

double calculateArea() {

double s = (a + b + c) / 2;

return Math.sqrt(s \* (s-a) \* (s-b) \* (s-c));

}

double calculatePerimeter() {

return a + b + c;

}

}

class Square extends Shape {

double side;

Square(double side) {

this.side = side;

}

double calculateArea() {

return side \* side;

}

double calculatePerimeter() {

return 4 \* side;

}

}

public class Main7 {

public static void main(String[] args) {

Triangle t = new Triangle(3, 4, 5);

System.out.println("Triangle Area: " + t.calculateArea());

System.out.println("Triangle Perimeter: " + t.calculatePerimeter());

Square s = new Square(4);

System.out.println("Square Area: " + s.calculateArea());

System.out.println("Square Perimeter: " + s.calculatePerimeter());

}

}