**Experiment – 4**

Q1) A set of 5 words (strings) will be taken as command line arguments. Write a program to reverse each word and check whether it is palindrome or not using method.

Ans:

**Program:**

public class First {

    public static void checkPalindrome(String s) {

        String reversed = "";

        int n = s.length();

        for (int i = n - 1; i >= 0; i--) {

            reversed += s.charAt(i);

        }

        if (s.equals(reversed)) {

            System.out.println(s + " is a palindrome");

        } else {

            System.out.println(s + " is not a palindrome");

        }

    }

    public static void main(String[] args) {

        if (args.length < 5) {

            System.out.println("Please provide at least 5 arguments.");

            return;

        }

        for (int i = 0; i < 5; i++) {

            checkPalindrome(args[i]);

        }

    }

}

**Output:**

A computer screen with white text

AI-generated content may be incorrect.

Q2) Define the class BankAccount to represent an account we open with bank. Define the subclasses SavingAccount and FixedDepositAccount. Implement the operations like openAccount(), deposit(), checkBalance(), withdraw() and calInterest() for these classes.

Ans:

**Program:**

class BankAccount{

    long AccNo;

    String AccHolderName;

    String DateOfOpening;

    double interestRate;

    double balance;

    BankAccount(){

        this.AccNo = 0;

        this.AccHolderName = null;

        this.DateOfOpening = null;

        this.interestRate = 0.0;

        this.balance = 0.0;

    }

    BankAccount(long AccNo, String AccHolderName, String DateOfOpening, double interestRate, double balance){

        this.AccNo = AccNo;

        this.AccHolderName = AccHolderName;

        this.DateOfOpening = DateOfOpening;

        this.interestRate = interestRate;

        this.balance = balance;

    }

    void displayDetails(){

        System.out.println("Account number: "+AccNo);

        System.out.println("Account holder name: "+AccHolderName);

        System.out.println("Date of account opening: "+DateOfOpening);

        System.out.println("Interest Rate: "+interestRate);

        System.out.println("Balance amount: "+balance);

    }

    void openAccount(long AccNo, String AccHolderName, String DateOfOpening, double interestRate, double balance){

        this.AccNo = AccNo;

        this.AccHolderName = AccHolderName;

        this.DateOfOpening = DateOfOpening;

        this.interestRate = interestRate;

        this.balance = balance;

    }

    double checkBalance(){

        return balance;

    }

}

class SavingsAccount extends BankAccount{

    double minBalance;

    double withdrawlLimit;

    SavingsAccount(long AccNo, String AccHolderName, String DateOfOpening, double interestRate, double balance, double minBalance, double withdrawlLimit ){

        super(AccNo, AccHolderName, DateOfOpening, interestRate,balance);

        this.minBalance = minBalance;

        this.withdrawlLimit = withdrawlLimit;

    }

    void displayDetails(){

        super.displayDetails();

        System.out.println("Minimum Balance: "+minBalance);

        System.out.println("Withdrawal limit: "+withdrawlLimit);

    }

    void openAccount(long AccNo, String AccHolderName, String DateOfOpening, double interestRate, double balance, double minBalance, double withdrawlLimit){

        super.openAccount(AccNo, AccHolderName, DateOfOpening, interestRate, balance);

        this.minBalance = minBalance;

        this.withdrawlLimit = withdrawlLimit;

        System.out.println("Succesfully opened new account");

    }

    double deposit(double amountToDeposit){

        double currentBalance = super.balance;

        currentBalance += amountToDeposit;

        return currentBalance;

    }

    double withdraw(double amountToWithdraw){

        if(amountToWithdraw > withdrawlLimit){

            System.out.println("Withdrawl amount greater than limit");

            return -1;

        }

        double currentBalance = super.balance;

        if(amountToWithdraw > currentBalance){

            System.out.println("Withdrawl amount greater than bank balance");

            return -1;

        }

        currentBalance -= amountToWithdraw;

        return currentBalance;

    }

}

class FixedDepositAccount extends BankAccount{

    double tenure;

    String maturityDate;

    double penalty;

    FixedDepositAccount(long AccNo, String AccHolderName, String DateOfOpening, double interestRate, double balance, double tenure,String maturityDate, double penalty){

        super(AccNo, AccHolderName, DateOfOpening, interestRate,balance);

        this.tenure = tenure;

        this.maturityDate = maturityDate;

        this.penalty = penalty;

    }

    double calInterest(){

        double currentBalance = super.balance;

        double interestRate = super.interestRate;

        return currentBalance \* (interestRate / 100);

    }

}

class Bank{

    public static void main(String[] args) {

        SavingsAccount sa = new SavingsAccount(100, "Aryan Shah", "10-12-2023", 5, 100000, 1000, 10000);

        sa.displayDetails();

        System.out.println("After deposit, current balance: "+sa.deposit(50000));

        System.out.println("After withdrawl, current balance: "+sa.withdraw(9000));

        sa.checkBalance();

        FixedDepositAccount fd = new FixedDepositAccount(101, "Jay Patel", "1-1-2014", 10, 100000, 3, "1-1-2017", 5000);

        fd.displayDetails();

        fd.calInterest();

    }

}

**Output:**

A computer screen shot of a number

AI-generated content may be incorrect.

Q3) Write a program that finds area of any shape by overloading area () method for Square, Rectangle, Triangle and Square

Ans:

**Program:**

import java.util.Scanner;

import java.lang.Math;

class Shape{

    double l;

    double b;

    double a;

    double c;

    Shape(){}

    Shape(double l){

        this.l = l;

    }

    Shape(double l,double b){

        this.l = l;

        this.b = b;

    }

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

        this.a = a;

        this.b = b;

        this.c = c;

    }

}

class Square extends Shape{

    Square(){}

    Square(double l){

        super(l);

    }

    double area(){

        return l\*l;

    }

}

class Rectangle extends Shape{

    Rectangle(){}

    Rectangle(double l,double b){

        super(l,b);

    }

    double area(){

        return l\*b;

    }

}

class Triangle extends Shape{

    Triangle(){}

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

        super(a,b,c);

    }

    double area(){

        double s = (a+b+c)/2;

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

    }

}

class Main{

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

System.out.print("Enter side of square: ");

        Square sq = new Square(sc.nextDouble());

System.out.print("Enter sides of rectangle: ");

        Rectangle rec = new Rectangle(sc.nextDouble(),sc.nextDouble());

System.out.print("Enter sides of triangle: ");

        Triangle tr = new Triangle(sc.nextDouble(),sc.nextDouble(),sc.nextDouble());

        System.out.println("Area of square: "+sq.area());

        System.out.println("Area of rectangle: "+rec.area());

        System.out.println("Area of triangle: "+tr.area());

        sc.close();

    }

}

**Output:**

A computer screen shot of a program

AI-generated content may be incorrect.

Q4) Write a program that finds Volume of any shape by overloading volume () method for Cube, Rectangular Cube and Sphere.

Ans:

**Program:**

import java.util.Scanner;

class Volume {

    double a,l,b,h,r;

    Volume(){}

    Volume(double a){

        this.a = a;

    }

    Volume(double l,double b, double h){

        this.l = l;

        this.b = b;

        this.h = h;

    }

    Volume(double r, boolean isSphere){

        if (isSphere) {

            this.r = r;

        }

    }

}

class Cube extends Volume{

    Cube(double a){

        super(a);

    }

    double volume(){

        return a\*a\*a;

    }

}

class Cuboid extends Volume{

    Cuboid(double l,double b,double h){

        super(l,b,h);

    }

    double volume(){

        return l\*b\*h;

    }

}

class Sphere extends Volume{

    Sphere(double r, boolean isSphere){

        super(r,isSphere);

    }

    double volume(){

        return (4/3)\*3.14\*r\*r\*r;

    }

}

class Shapes{

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter dimensions of cube: ");

        Cube cube = new Cube(sc.nextDouble());

        System.out.println("Enter dimensions of cuboid: ");

        Cuboid cuboid = new Cuboid(sc.nextDouble(),sc.nextDouble(),sc.nextDouble());

        System.out.println("Enter radius of sphere: ");

        Sphere sphere = new Sphere(sc.nextDouble(),true);

        System.out.println("Volume of cube: "+cube.volume());

        System.out.println("Volume of cube: "+cuboid.volume());

        System.out.println("Volume of cube: "+sphere.volume());

        sc.close();

    }

}

**Output:**

A computer screen shot of a black screen

AI-generated content may be incorrect.

Q5) Write a Program to maintain employee’s information. Program should illustrate Inheritance concept. (Use your imagination to create class or subclass used for employee).

Ans:

**Program:**

class Employee{

    double eid;

    double salary;

    String fname;

    String lname;

    Employee(){

        this.eid = 0;

        this.salary = -1;

        this.fname = null;

        this.lname = null;

    }

    Employee(double eid, double salary, String fname, String lname){

        this.eid = eid;

        this.salary = salary;

        this.fname = fname;

        this.lname = lname;

    }

    void showEmployeeDetails(){

        System.out.println("Employee id: "+eid);

        System.out.println("Employee name: "+fname+" "+lname);

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

    }

    double incrementSalary(double incrementRate){

        salary = salary + (salary \* (incrementRate/100));

        return salary;

    }

}

class SoftwareEngineer extends Employee{

    String projectName;

    String position;

    double teamId;

    String teamLeadName;

    SoftwareEngineer(){

        super();

        this.projectName = null;

        this.position = null;

        this.teamId = 0;

        this.teamLeadName = null;

    }

    SoftwareEngineer(double eid, double salary, String fname, String lname, String projectName, String position,

                      double teamId, String teamLeadName){

        super(eid,salary,fname,lname);

        this.projectName = projectName;

        this.position = position;

        this.teamId = teamId;

        this.teamLeadName = teamLeadName;

    }

    void showTeamDetails(){

        super.showEmployeeDetails();

        System.out.println("Software engineer position: "+position);

        System.out.println("Team id: "+teamId);

        System.out.println("Team project: "+projectName);

        System.out.println("Team leader name: "+teamLeadName);

    }

    void changeTeam(String newProjectName, double newTeamId, String newTeamLeadName){

        this.projectName = newProjectName;

        this.teamId = newTeamId;

        this.teamLeadName = newTeamLeadName;

        System.out.println("Successfully changed team of employee "+fname+" "+lname);

    }

}

class Manager extends Employee{

    long noOfEmployeesManaging;

    Manager(){

        super();

        this.noOfEmployeesManaging = -1;

    }

    Manager(double eid, double salary, String fname, String lname, long noOfEmployeesManaging){

        super(eid,salary,fname,lname);

        this.noOfEmployeesManaging = noOfEmployeesManaging;

    }

    void showManagerDetails(){

        super.showEmployeeDetails();

        System.out.println("Number of employees managed by "+fname+" "+lname+" are: "+noOfEmployeesManaging);

    }

}

public class EmployeeTest {

    public static void main(String[] args) {

        SoftwareEngineer se1 = new SoftwareEngineer();

        SoftwareEngineer se2 = new SoftwareEngineer(1,100,"Aryan","Shah","AI Chatbot","Senior SE",394,"Jay Singhania");

        Manager mg1 = new Manager();

        Manager mg2 = new Manager(150,10000,"Krish","Gupta",500);

        se1.showTeamDetails();

        se2.showTeamDetails();

        System.out.println("New salary: "+se2.incrementSalary(5));

        se2.changeTeam("AI interface", 500, "Rahil Shah");

        se2.showTeamDetails();

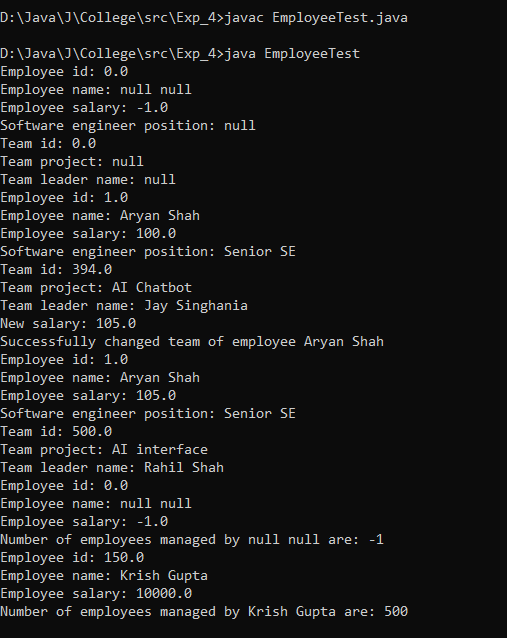
        mg1.showManagerDetails();

        mg2.showManagerDetails();

    }

}

**Output:**



Q6) Create a base class Shape. Use this class to store two double type values that could be used to compute area of any shape. Derive two specific classes called Triangle and Rectangle from the base shape. Add to the base a member function getdata() to initialize base class data member and another member function display\_area() to compute and display the area of figures. (Use Method Overriding).

Ans:

**Program:**

import java.lang.Math;

class Shape{

    double a;

    double b;

    Shape(){

        this.a = 0;

        this.b = 0;

    }

    Shape(double a,double b){

        this.a = a;

        this.b = b;

    }

    void getData(double a,double b){

        this.a = a;

        this.b = b;

    }

    double display\_area(){return 0.0;}

}

class RectangleClass extends Shape{

    RectangleClass(){

        super();

    }

    RectangleClass(double a,double b){

        super(a,b);

    }

    double display\_area(){

        return a\*b;

    }

}

class TriangleClass extends Shape{

    double c;

    TriangleClass(){

        super();

    }

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

        super(a,b);

        this.c = c;

    }

    double display\_area(){

        double s = (a+b+c)/2;

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

    }

}

public class ShapeTesting{

    public static void main(String[] args) {

        Shape sh;

        sh = new RectangleClass(1,2);

        System.out.println(sh.display\_area());

        sh = new TriangleClass(3,4,5);

        System.out.println(sh.display\_area());

    }

}

**Output:**

A screen shot of a computer code

AI-generated content may be incorrect.