

Prg:-01

```
import java.util.Scanner;
public class Program1 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter the order of the matrices (N x N): ");
        int N = scanner.nextInt();
        int matrix1 [][] = new int[N][N];
        int[][] matrix2 = new int[N][N];
        int[][] sumMatrix = new int[N][N];
        System.out.println("Enter elements of the first matrix:");
        for (int i = 0; i < N; i++) {
            for (int j = 0; j < N; j++) {
                matrix1[i][j] = scanner.nextInt();
            }
        }
        System.out.println("Enter elements of the second matrix:");
        for (int i = 0; i < N; i++) {
            for (int j = 0; j < N; j++) {
                matrix2[i][j] = scanner.nextInt();
            }
        }
        for (int i = 0; i < N; i++) {
            for (int j = 0; j < N; j++) {
                sumMatrix[i][j] = matrix1[i][j] + matrix2[i][j];
            }
        }
        System.out.println("Sum of the two matrices:");
        for (int i = 0; i < N; i++) {
            for (int j = 0; j < N; j++) {
                System.out.print(sumMatrix[i][j] + " ");
            }
        }
        System.out.println();
    }
    scanner.close();
} }
```

Prg:-02

```
import java.util.Scanner;
class Stack {
    private int[] stack = new int[5];
    private int top = -1;
    void push(int value) {
        if (top < 4) {
            stack[++top] = value;
            System.out.println(value + " pushed.");
        } else {
            System.out.println("Stack Overflow!");
        }
    }
    void pop() {
        if (top >= 0) {
            System.out.println(stack[top--] + " popped.");
        } else {
            System.out.println("Stack Underflow!");
        }
    }
}
```

```

void display() {
if (top == -1) {
    System.out.println("Stack is empty.");
} else {
    System.out.println("Stack elements:");
    for (int i = 0; i <= top; i++) {
        System.out.print(stack[i] + " ");
    }
    System.out.println();
} } }
public class Program2 {
public static void main(String[] args) {
    Stack obj = new Stack();
    Scanner sc = new Scanner(System.in);
    int choice, value;
    do {
        System.out.println("\nChoose an operation:");
        System.out.println("1. Push");
        System.out.println("2. Pop");
        System.out.println("3. Display");
        System.out.println("4. Exit");
        System.out.print("Enter your choice: ");
        choice = sc.nextInt();
        switch (choice) {
        case 1:
            System.out.print("Enter value to push: ");
            value = sc.nextInt();
            obj.push(value);
            break;
        case 2:
            obj.pop();
            break;
        case 3:
            obj.display();
            break;
        case 4:
            System.out.println("Exiting...");
            break;
        default:
            System.out.println("Invalid choice! Please choose again.");
        } } while (choice != 4);
    } }

```

Prg:-03

```
class Employee {
    private int id;
    private String name;
    private double salary;
    public Employee(int id, String name, double salary) {
        this.id = id;
        this.name = name;
        this.salary = salary;
    }
    public void raiseSalary(double percent) {
        if (percent > 0) {
            salary += salary * (percent / 100);
        } else {
            System.out.println("Invalid raise percentage.");
        } }
    public int getId() {
        return id;
    }
    public String getName() {
        return name;
    }
    public double getSalary() {
        return salary;
    }
    public void displayEmployeeDetails() {
        System.out.println("Employee ID: " + id);
        System.out.println("Employee Name: " + name);
        System.out.println("Employee Salary: $" + salary);
    }
}

public class Program3 {
    public static void main(String[] args) {
        Employee emp = new Employee(101, "XYZ", 50000);
        System.out.println("Initial Employee Details:");
        emp.displayEmployeeDetails();
        emp.raiseSalary(10);
        System.out.println("\nEmployee Details after 10% raise:");
        emp.displayEmployeeDetails();
    } }
}
```

Prg:-04

```
class MyPoint
{
    private int x;
    private int y;
    public MyPoint()
    {
        this.x = 0;
        this.y = 0;
    }
    public MyPoint(int x, int y)
    {
        this.x = x;
```

```

    this.y = y;
}
public void setXY(int x, int y) {
    this.x = x;
    this.y = y;
}
public int[] getXY()
{
    return new int[] { x, y };
}
@Override
public String toString()
{
    return "(" + x + ", " + y + ")";
}
public double distance(int x, int y)
{
    int dx = this.x - x;
    int dy = this.y - y;
    return Math.sqrt(dx * dx + dy * dy);
}
public double distance(MyPoint another)
{
    return distance(another.x, another.y);
}
public double distance()
{
    return distance(0, 0);
}
public int getX()
{
    return x;
}
public int getY()
{
    return y;
}}
public class Program4 {
    public static void main(String[] args) {
        MyPoint p1 = new MyPoint();
        System.out.println("Point p1: " + p1);
        MyPoint p2 = new MyPoint(3, 4);
        System.out.println("Point p2: " + p2); // Should print (3, 4)
        p1.setXY(5, 6);
        System.out.println("After setting p1 to (5, 6): " + p1);
        int[] coordinates = p2.getXY();
        System.out.println("Coordinates of p2: (" + coordinates[0] + ", " + coordinates[1] + ")"); // Should
        print (3, 4)
        System.out.println("Distance from p1 to (1, 1): " + p1.distance(1, 1));
        System.out.println("Distance from p1 to p2: " + p1.distance(p2));
        System.out.println("Distance from p1 to origin: " + p1.distance()); }

```

Prg:-05

```
class Shape {
    void draw() {
        System.out.println("Drawing a shape");
    }
    void erase() {
        System.out.println("Erasing a shape");
    }
}
class Circle extends Shape {
    @Override
    void draw() {
        System.out.println("Drawing a circle");
    }
    @Override
    void erase() {
        System.out.println("Erasing a circle");
    }
}
class Triangle extends Shape {
    @Override
    void draw() {
        System.out.println("Drawing a triangle");
    }
    @Override
    void erase() {
        System.out.println("Erasing a triangle");
    }
}
class Square extends Shape {
    @Override
    void draw() {
        System.out.println("Drawing a square");
    }
    @Override
    void erase() {
        System.out.println("Erasing a square");
    }
}
public class P5 {
    public static void main(String[] args) {
        Shape shape;
        shape = new Circle();
        shape.draw();
        shape.erase();
        shape = new Triangle();
        shape.draw();
        shape.erase();
        shape = new Square();
        shape.draw();
        shape.erase();
    } }
```

Prg:-06

```
abstract class Shape {
    abstract double calculateArea();
    abstract double calculatePerimeter();
}
class Circle extends Shape {
    private double radius;
    public Circle(double radius) {
        this.radius = radius;
    }
    @Override
    double calculateArea() {
        return Math.PI * radius * radius;
    }
    @Override
    double calculatePerimeter() {
        return 2 * Math.PI * radius;
    }
}
class Triangle extends Shape {
    private double base;
    private double height;
    private double sideA, sideB, sideC;
    public Triangle(double base, double height, double sideA, double sideB, double sideC)
    {
        this.base = base;
        this.height = height;
        this.sideA = sideA;
        this.sideB = sideB;
        this.sideC = sideC;
    }
    @Override
    double calculateArea() {
        return 0.5 * base * height;
    }
    @Override
    double calculatePerimeter() {
        return sideA + sideB + sideC;
    }
}
public class P6 {
    public static void main(String[] args) {
        Shape circle = new Circle(5.0); // Radius = 5
        System.out.println("Circle Area: " + circle.calculateArea());
        System.out.println("Circle Perimeter: " + circle.calculatePerimeter());
        Shape triangle = new Triangle(5.0, 4.0, 3.0, 4.0, 5.0);
        System.out.println("Triangle Area: " + triangle.calculateArea());
        System.out.println("Triangle Perimeter: " + triangle.calculatePerimeter());
    } }
```