**Assignment 3**

Name:- Bibek Chand Sah

Roll No. :- 22054029

Branch :- CSE

Q.1) Write a program in Java to find the largest among 3 user entered nos. through command line.

import java.util.Scanner;

public class large {

    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);

        System.out.print("Enter first num: ");

        double num1 = scan.nextDouble();

        System.out.print("Enter second num: ");

        double num2 = scan.nextDouble();

        System.out.print("Enter third num: ");

        double num3 = scan.nextDouble();

        System.out.print("\n");

        if (num1 > num2 && num1 > num3) {

            System.out.println(num1 + " is greater among three");

        } else if (num2 > num1 && num2 > num3) {

            System.out.println(num2 + " is greater among three");

        } else if (num3 > num1 && num3 > num2) {

            System.out.println(num3 + " is greater among three");

        } else {

            System.out.println("Please enter a valid number sequence:)");

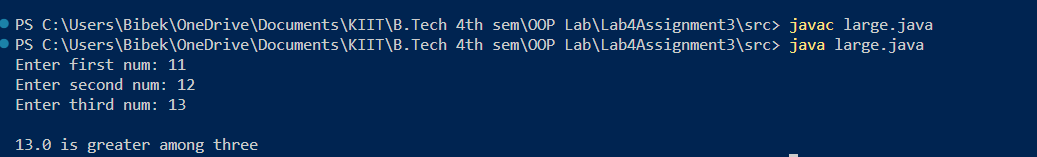
        }

        scan.close();

    }

}

**Output**



Q.2) Write a program in Java to accept 10 numbers from command line and check how many of them are even and how many odd.

import java.util.\*;

public class even {

    public static void main(String args[]) {

        int even\_count = 0;

        int odd\_count = 0;

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter 10 number:");

        int num[] = new int[10];

        for (int i = 0; i <= 9; i++) {

            num[i] = sc.nextInt();

        }

        for (int i = 0; i <= 9; i++) {

            if (num[i] % 2 == 0) {

                even\_count++;

            } else {

                odd\_count++;

            }

        }

        sc.close();

        System.out.println("even:" + even\_count);

        System.out.println("Odd:" + odd\_count);

    }

}

Output

A blue screen with white text

Description automatically generated

Q.3)  
Write a program in Java to create a class – Box with three data members (length, width, height) and a method volume (). Also implement the application class Demo where an object of the box class is created withuser entered dimensions and volume is printed.

import java.util.\*;

class Box {

    int length;

    int width;

    int height;

    void volume() {

        System.out.println(length \* width \* height);

    }

}

public class BoxDimension {

    public static void main(String args[]) {

        Scanner sc = new Scanner(System.in);

        Box b1 = new Box();

        System.out.print("Enter length :- ");

        b1.length = sc.nextInt();

        System.out.print("Enter width :- ");

        b1.width = sc.nextInt();

        System.out.print("Enter height :- ");

        b1.height = sc.nextInt();

        b1.volume();

        sc.close();

    }

}

Output

Enter length :- 10

Enter width :- 11

Enter height :- 12

1320

Q.4) Write a program in Java to overload subtract method with various parameters in a class.

class subtractOverload {

    public int subtract(int a, int b) {

        return a - b;

    }

    public int subtract(int a, int b, int c) {

        return a - b - c;

    }

}

public class Subtract {

    public static void main(String[] args) {

        subtractOverload s1 = new subtractOverload();

        System.out.println("Subtract of 2 integers:" + s1.subtract(3, 6));

        System.out.println("Subtract of 3 integers:" + s1.subtract(4, 6, 7));

    }

}

output

Subtract of 2 integers:-3

Subtract of 3 integers:-9

Q.5)Write a program in Java which will overload the area () method and display the area of a circle, triangle and square as per user choice and user entered dimensions.

import java.util.Scanner;

public class ShapeAreaOverloading {

    public double area(double radius) {

        return Math.PI \* radius \* radius;

    }

    public double area(double base, double height) {

        return 0.5 \* base \* height;

    }

    public double areaa(double sideLength) {

        return sideLength \* sideLength;

    }

    public static void main(String[] args) {

        ShapeAreaOverloading shapeAreaCalculator = new ShapeAreaOverloading();

        Scanner scanner = new Scanner(System.in);

        System.out.println("Choose a shape to calculate area:");

        System.out.println("1. Circle");

        System.out.println("2. Triangle");

        System.out.println("3. Square");

        System.out.print("Enter your choice (1, 2, or 3): ");

        int choice = scanner.nextInt();

        double result = 0;

        switch (choice) {

            case 1:

                System.out.print("Enter the radius of the circle: ");

                double radius = scanner.nextDouble();

                result = shapeAreaCalculator.area(radius);

                break;

            case 2:

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

                double base = scanner.nextDouble();

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

                double height = scanner.nextDouble();

                result = shapeAreaCalculator.area(base, height);

                break;

            case 3:

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

                double sideLength = scanner.nextDouble();

                result = shapeAreaCalculator.areaa(sideLength);

                break;

            default:

                System.out.println("Invalid choice!");

                break;

        }

        if (choice >= 1 && choice <= 3) {

            System.out.println("The area is: " + result);

        }

        scanner.close();

    }

}

Output

Choose a shape to calculate area:

1. Circle

2. Triangle

3. Square

Enter your choice (1, 2, or 3): 3

Enter the side length of the square: 5

The area is: 25.0

Q.6) Write a program in Java to define a class Rectangle having data member: length and breadth; to calculate the area and perimeter of the rectangle. Use constructor to read, and member functions to calculate and display.

import java.util.Scanner;

class Rectangle {

    private double length;

    private double breadth;

    public Rectangle(double length, double breadth) {

        this.length = length;

        this.breadth = breadth;

    }

    public double calculateArea() {

        return length \* breadth;

    }

    public double calculatePerimeter() {

        return 2 \* (length + breadth);

    }

    public void display() {

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

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

    }

}

public class RectangleDimension {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

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

        double length = scanner.nextDouble();

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

        double breadth = scanner.nextDouble();

        Rectangle rectangle = new Rectangle(length, breadth);

        rectangle.display();

        scanner.close();

    }

}

Output

Enter the length of the rectangle: 5

Enter the breadth of the rectangle: 7

Area: 35.0

Perimeter: 24.0

Q.7) Write a program in java to input the details of a student having roll, name, full\_mark and  
secured\_mark as data members using constructor, then calculate the CGPA and display the details of student with CGPA.

import java.util.Scanner;

public class StudentCGPA {

    private int roll;

    private String name;

    private int fullMarks;

    private int securedMarks;

    private double cgpa;

    public StudentCGPA(int roll, String name, int fullMarks, int securedMarks) {

        this.roll = roll;

        this.name = name;

        this.fullMarks = fullMarks;

        this.securedMarks = securedMarks;

    }

    public void calculateCGPA() {

        cgpa = (securedMarks \* 100.0) / fullMarks;

    }

    public void displayDetails() {

        System.out.println("Roll: " + roll);

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

        System.out.println("Full Marks: " + fullMarks);

        System.out.println("Secured Marks: " + securedMarks);

        System.out.println("CGPA: " + cgpa);

    }

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        System.out.print("Enter student roll: ");

        int roll = input.nextInt();

        System.out.print("Enter student name: ");

        String name = input.next();

        System.out.print("Enter full marks: ");

        int fullMarks = input.nextInt();

        System.out.print("Enter secured marks: ");

        int securedMarks = input.nextInt();

        StudentCGPA student = new StudentCGPA(roll, name, fullMarks, securedMarks);

        student.calculateCGPA();

        student.displayDetails();

        input.close();

    }

}

Output

Enter student roll: 4029

Enter student name: Bibek

Enter full marks: 92

Enter secured marks: 89

Roll: 4029

Name: Bibek

Full Marks: 92

Secured Marks: 89

CGPA: 96.73913043478261

Q.8) Write a program in Java to create a class MyVolume with required data members and find the volume of cube, cuboid and sphere using constructor overloading.

import java.util.Scanner;

class MyVolume {

    private double volume;

    public MyVolume(double side) {

        volume = Math.pow(side, 3);

    }

    public MyVolume(double length, double width, double height) {

        volume = length \* width \* height;

    }

    public MyVolume(double radius, String shape) {

        if (shape.equalsIgnoreCase("sphere")) {

            volume = (4.0 / 3.0) \* Math.PI \* Math.pow(radius, 3);

        } else {

            System.out.println("Invalid shape provided for volume calculation.");

        }

    }

    public void displayVolume() {

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

    }

}

public class MyVolumes {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter side length of the cube: ");

        double cubeSide = scanner.nextDouble();

        MyVolume cubeVolume = new MyVolume(cubeSide);

        System.out.print("Volume of Cube - ");

        cubeVolume.displayVolume();

        System.out.print("\nEnter length of the cuboid: ");

        double cuboidLength = scanner.nextDouble();

        System.out.print("Enter width of the cuboid: ");

        double cuboidWidth = scanner.nextDouble();

        System.out.print("Enter height of the cuboid: ");

        double cuboidHeight = scanner.nextDouble();

        MyVolume cuboidVolume = new MyVolume(cuboidLength, cuboidWidth, cuboidHeight);

        System.out.print("Volume of Cuboid - ");

        cuboidVolume.displayVolume();

        System.out.print("\nEnter radius of the sphere: ");

        double sphereRadius = scanner.nextDouble();

        MyVolume sphereVolume = new MyVolume(sphereRadius, "sphere");

        System.out.print("Volume of Sphere - ");

        sphereVolume.displayVolume();

        scanner.close();

    }

}

Output

Enter side length of the cube: 3

Volume of Cube - Volume: 27.0

Enter length of the cuboid: 3

Enter width of the cuboid: 5

Enter height of the cuboid: 6

Volume of Cuboid - Volume: 90.0

Enter radius of the sphere: 5

Volume of Sphere - Volume: 523.5987755982989

Q.9) A plastic manufacturer sells plastic in different shapes like 2D sheet and 3D box. The cost of sheet is Rs 40/ per square ft. and the cost of box is Rs 60/ per cubic ft. Implement it in Java to calculate the cost of plastic as per the dimensions given by the user where 3D inherits from 2D.

abstract class Plastic {

    double costPerUnitArea;

    Plastic(double costPerUnitArea) {

        this.costPerUnitArea = costPerUnitArea;

    }

    abstract double calculateCost(double[] dimensions);

    double calculateCost(double dimension) {

        return costPerUnitArea \* dimension;

    }

}

class PlasticSheet extends Plastic {

    PlasticSheet() {

        super(40);

    }

    @Override

    double calculateCost(double[] dimensions) {

        double length = dimensions[0];

        double width = dimensions[1];

        return costPerUnitArea \* length \* width;

    }

}

class PlasticBox extends Plastic {

    PlasticBox() {

        super(60);

    }

    @Override

    double calculateCost(double[] dimensions) {

        double length = dimensions[0];

        double width = dimensions[1];

        double height = dimensions[2];

        return costPerUnitArea \* length \* width \* height;

    }

}

public class PlasticCostCalculator {

    public static void main(String[] args) {

        PlasticSheet sheet = new PlasticSheet();

        double[] sheetDimensions = {10.0, 15.0};

        double sheetCost = sheet.calculateCost(sheetDimensions);

        System.out.println("Cost of plastic sheet: " + sheetCost);

        PlasticBox box = new PlasticBox();

        double[] boxDimensions = {12.0, 8.0, 6.0};

        double boxCost = box.calculateCost(boxDimensions);

        System.out.println("Cost of plastic box: " + boxCost);

    }

}

Output

Cost of plastic sheet: 6000.0

Cost of plastic box: 34560.0

Q. 10) SOLVE THE PROBLEM AVAILABLE IN THE ATTACHMENT.

a) Implement a class Triangle having data members a,b,c as it's sides. Include the following

methods:

1. set Dim(): is used to initialize the data members a,b and c.
2. boolean is\_Triangle(): is used to check whether the sides are permissible values to form triangle or not.
3. float\_find\_area(): this method is only invoked if is\_Triangle() is true otherwise appropriate message is displayed.
4. float\_find\_perimeter(): this method is only invoked if is\_Triangle() is true otherwise appropriate message is displayed.
5. void show(): is used to display the sides, area and perimeter of the triangle.

Write down the application class Triangle\_Demo where the functionality of the Triangle class is tested.

import java.util.Scanner;

class Triangle {

    private float a, b, c;

    public void setDim(float sideA, float sideB, float sideC) {

        a = sideA;

        b = sideB;

        c = sideC;

    }

    public boolean is\_Triangle() {

        return (a + b > c) && (b + c > a) && (c + a > b);

    }

    public float find\_area() {

        if (is\_Triangle()) {

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

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

        } else {

            System.out.println("Invalid sides for a triangle. Cannot calculate area.");

            return -1;

        }

    }

    public float find\_perimeter() {

        if (is\_Triangle()) {

            return a + b + c;

        } else {

            System.out.println("Invalid sides for a triangle. Cannot calculate perimeter.");

            return -1;

        }

    }

    public void show() {

        System.out.println("Sides of the triangle: a=" + a + ", b=" + b + ", c=" + c);

        System.out.println("Area of the triangle: " + find\_area());

        System.out.println("Perimeter of the triangle: " + find\_perimeter());

    }

}

public class TriangleDimension {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        Triangle triangle = new Triangle();

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

        System.out.print("Enter side a: ");

        float sideA = scanner.nextFloat();

        System.out.print("Enter side b: ");

        float sideB = scanner.nextFloat();

        System.out.print("Enter side c: ");

        float sideC = scanner.nextFloat();

        triangle.setDim(sideA, sideB, sideC);

        if (triangle.is\_Triangle()) {

            triangle.show();

        } else {

            System.out.println("Invalid sides for a triangle.");

        }

        scanner.close();

    }

}

**Output**

Enter the sides of the triangle:

Enter side a: 5

Enter side b: 3

Enter side c: 6

Sides of the triangle: a=5.0, b=3.0, c=6.0

Area of the triangle: 7.483315

Perimeter of the triangle: 14.0

b) Implement a class Number having data members value. Include the following predicate methods:

1. boolean is\_Even():(): is used to check whether the value is even or not.
2. boolean is\_Prime():(): is used to check whether the value is prime or not.
3. boolean is\_Perfect():(): is used to check whether the value is Perfect or not.
4. int fin\_Factorial(): recursive method to find the factorial.

Write down the application class Number\_Demo where the functionality of the Number class is tested.

import java.util.Scanner;

class Number {

    private int value;

    public void setValue(int value) {

        this.value = value;

    }

    public boolean isEven() {

        return value % 2 == 0;

    }

    public boolean isPrime() {

        if (value <= 1) {

            return false;

        }

        for (int i = 2; i <= Math.sqrt(value); i++) {

            if (value % i == 0) {

                return false;

            }

        }

        return true;

    }

    public boolean isPerfect() {

        int sum = 1;

        for (int i = 2; i <= value / 2; i++) {

            if (value % i == 0) {

                sum += i;

            }

        }

        return sum == value;

    }

    public int findFactorial() {

        if (value < 0) {

            System.out.println("Factorial is not defined for negative numbers.");

            return -1;

        }

        return recursiveFactorial(value);

    }

    private int recursiveFactorial(int n) {

        if (n == 0 || n == 1) {

            return 1;

        } else {

            return n \* recursiveFactorial(n - 1);

        }

    }

}

public class Numbers {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        Number number = new Number();

        System.out.print("Enter a number: ");

        int inputValue = scanner.nextInt();

        number.setValue(inputValue);

        System.out.println("Number is even: " + number.isEven());

        System.out.println("Number is prime: " + number.isPrime());

        System.out.println("Number is perfect: " + number.isPerfect());

        System.out.println("Factorial of the number: " + number.findFactorial());

        scanner.close();

    }

}

**Output**

Enter a number: 3

Number is even: false

Number is prime: true

Number is perfect: false

Factorial of the number: 6

c) Implement a Point class for three diemensional point(x,y,z). Include appropriate constructors :

Point(), Point(int,int,int) and Point(Point). Include the following methods

1. float find\_distance(): distance from the origin (0,0,0).
2. float find\_distance(int x1, int y2, int z2): distance between (x,y,z) and (x1,y1,z1).
3. float find\_distance(Point P1): distance between two points.
4. boolean is\_Equal(Point P1): equality between two points.
5. void show(): shows the description about the point object.

Write down the application class Point Demo where the functionality of the Point class is tested.

import java.util.Scanner;

class Point {

    private int x, y, z;

    public Point() {

        x = 0;

        y = 0;

        z = 0;

    }

    public Point(int x, int y, int z) {

        this.x = x;

        this.y = y;

        this.z = z;

    }

    public Point(Point point) {

        this.x = point.x;

        this.y = point.y;

        this.z = point.z;

    }

    public float find\_distance() {

        return (float) Math.sqrt(x \* x + y \* y + z \* z);

    }

    public float find\_distance(int x1, int y1, int z1) {

        return (float) Math.sqrt(Math.pow(x1 - x, 2) + Math.pow(y1 - y, 2) + Math.pow(z1 - z, 2));

    }

    public float find\_distance(Point P1) {

        return (float) Math.sqrt(Math.pow(P1.x - x, 2) + Math.pow(P1.y - y, 2) + Math.pow(P1.z - z, 2));

    }

    public boolean is\_Equal(Point P1) {

        return x == P1.x && y == P1.y && z == P1.z;

    }

    public void show() {

        System.out.println("Point coordinates: (" + x + ", " + y + ", " + z + ")");

    }

}

public class Points {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        Point originPoint = new Point();

        Point userPoint = new Point(3, 4, 5);

        Point copiedPoint = new Point(userPoint);

        System.out.println("Distance from origin: " + originPoint.find\_distance());

        System.out.println("Distance from user-defined point: " + userPoint.find\_distance());

        System.out.println("Distance between user-defined point and (0,0,0): " + userPoint.find\_distance(0, 0, 0));

        System.out.println(

                "Distance between user-defined point and copied point: " + userPoint.find\_distance(copiedPoint));

        System.out.println("Equality between user-defined point and copied point: " + userPoint.is\_Equal(copiedPoint));

        originPoint.show();

        userPoint.show();

        copiedPoint.show();

        scanner.close();

    }

}

Output

Distance from origin: 0.0

Distance from user-defined point: 7.071068

Distance between user-defined point and (0,0,0): 7.071068

Distance between user-defined point and copied point: 0.0

Equality between user-defined point and copied point: true

Point coordinates: (0, 0, 0)

Point coordinates: (3, 4, 5)

Point coordinates: (3, 4, 5)