Class & Objects

Q1. Room Volume Calculation

Design a class named Room with three data members: height, width, and breadth. Include a method volume() to compute and return the volume of the room. Create a separate class RoomDemo that creates instances of the Room class and displays the volume for each instance.

```
import java.util.Scanner; // Import Scanner

public class Room {
   double height;
   double width;
   double breadth;

   public Room(double height, double width, double breadth) {
      this.height = height;
      this.width = width;
      this.breadth = breadth;
   }

   public double volume() {
      return height * width * breadth;
   }
}

class VolumeDemo {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in); // Create Scanner object

      // Taking user input
      System.out.print("Enter height: ");
      double height = scanner.nextDouble();
```

```
    System.out.print("Enter width: ");
    double width = scanner.nextDouble();

    System.out.print("Enter breadth: ");
    double breadth = scanner.nextDouble();

    // Close scanner (optional, but recommended)
    scanner.close();

    // Create Room object with user inputs
    Room c = new Room(height, width, breadth);

    // Display volume
    System.out.println("Volume of the room = " + c.volume());
  }
}
```

Q2. Student Marks and Average

Create a class Student with the following members:

- Name of the student
- Marks in three subjects
- A method to assign initial values
- A method to compute the total and average marks
- A method to display the student’s name and total marks

```
import java.util.Scanner;
class Student{
        String Name;
        double ScienceMarks;
        double MathMarks;
```

```
	double HistoryMarks;

	public Student(String Name,double ScienceMarks,double MathMarks,double HistoryMarks){
		this.Name=Name;
		this.ScienceMarks=ScienceMarks;
		this.MathMarks=MathMarks;
		this.HistoryMarks=HistoryMarks;
	}
	public double Total(){
		return ScienceMarks + MathMarks + HistoryMarks ;
	}

	public double Avg(){

		return (ScienceMarks + MathMarks + HistoryMarks)/3;
	}

	public void Display(){
		System.out.println("Name of Student= " +Name+ "  Average Marks= " +Avg());
	}
}
class AvgMarks{
	public static void main(String[] args){
		Scanner sc = new Scanner(System.in);
		System.out.println("Enter name of Student= ");
		String Name = sc.next();
		System.out.println("Enter name Marks of Science= ");
		double ScienceMarks = sc.nextDouble();
		System.out.println("Enter name Marks of Math= ");
		double MathMarks = sc.nextDouble();
		System.out.println("Enter name of History= ");
```

```
        double HistoryMarks = sc.nextDouble();

        Student m = new Student(Name,ScienceMarks,MathMarks,HistoryMarks);
        m.Display();
    }
}
```

Q3. Box Area and Volume

Write a class Box with three member variables: height, width, and breadth. Include appropriate constructors to initialize these variables. Also, implement two methods:

- getVolume() to return the volume of the box
- getArea() to return the surface area of the box

Create two instances of the Box class and display their volumes and surface areas.

```
import java.util.Scanner;

class Box{
    double height;
    double width;
    double breadth;

    public Box(double height,double width,double breadth){
        this.height = height;
        this.width = width;
        this.breadth =breadth;
    }

    public double getVolume(){
        return height*width*breadth;
    }
```

```
        public double getArea(){
                return height*width;
        }
}
class BoxAreaVolume{
        public static void main(String[] args){
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter height of box= ");
                double height = sc.nextDouble();
                System.out.println("Enter width of box= ");
                double width = sc.nextDouble();
                System.out.println("Enter breadth of box= ");
                double breadth = sc.nextDouble();

                Box b = new Box(height,width,breadth);
                System.out.println("Enter volume of box= " + b.getVolume());
                System.out.println("Enter area of box= " + b.getArea());

        }
}
```

Q4. Complex Number Operations

Create a class to represent complex numbers. Include the following constructors:

1. A default constructor that sets both real and imaginary parts to 0
2. A constructor that initializes the real part only
3. A constructor that initializes both real and imaginary parts

Also, write member functions to:

- Add two complex numbers
- Multiply two complex numbers

In the main() method:

- Create two complex numbers: 3 + 2i and 4 - 2i
- Display their sum and product

```
class Complex {
    private double real;
    private double imaginary;

    // Default constructor (sets real and imaginary to 0)
    public Complex() {
        this.real = 0;
        this.imaginary = 0;
    }

    // Constructor with only real part
    public Complex(double real) {
        this.real = real;
        this.imaginary = 0;
    }

    // Constructor with both real and imaginary parts
    public Complex(double real, double imaginary) {
        this.real = real;
        this.imaginary = imaginary;
    }

    // Method to add two complex numbers
    public Complex add(Complex c) {
        return new Complex(this.real + c.real, this.imaginary + c.imaginary);
    }

    // Method to multiply two complex numbers
    public Complex multiply(Complex c) {
        double realPart = (this.real * c.real) - (this.imaginary * c.imaginary);
        double imaginaryPart = (this.real * c.imaginary) + (this.imaginary * c.real);
        return new Complex(realPart, imaginaryPart);
    }

    // Method to display a complex number
    public void display() {
        System.out.println(this.real + " + " + this.imaginary + "i");
    }
}

public class ComplexNumberOperations {
    public static void main(String[] args) {
        // Creating complex numbers: 3 + 2i and 4 - 2i
        Complex c1 = new Complex(3, 2);
        Complex c2 = new Complex(4, -2);

        // Adding the two complex numbers
        Complex sum = c1.add(c2);

        // Multiplying the two complex numbers
        Complex product = c1.multiply(c2);

        // Displaying results
        System.out.print("Sum: ");
        sum.display();
        System.out.print("Product: ");
        product.display();
    }
}
```

Q5. BMI Calculator

Design a Java program to implement a BMI (Body Mass Index) calculator. The program should consist of a class named BMICalculator with the following specifications:

CDAC Mumbai

Class: BMICalculator

Fields

- height (double): To store the height of the person in meters.
- weight (double): To store the weight of the person in kilograms.

Constructors

- A parameterized constructor to initialize the height and weight fields.

Methods

- Getter and Setter methods for both height and weight.
- double calculateBMI(): This method calculates and returns the BMI using the formula:

BMI=weight(height×height)\text{BMI} = \frac{\text{weight}}{(\text{height} \times \text{height})}BMI=(height×height)weight

Main Program : Write a separate class containing the main() method to

1. Create an object of the BMICalculator class.
2. Prompt the user to enter their height and weight.
3. Use setter methods to assign these values to the object.
4. Call the calculateBMI() method to compute the BMI.
5. Print the calculated BMI to the console.

```
import java.util.Scanner;


class BMICalculator {
    private double height; // Height in meters
    private double weight; // Weight in kilograms


    // Parameterized Constructor
    public BMICalculator(double height, double weight) {
        this.height = height;
        this.weight = weight;
    }

    // Getter for height
    public double getHeight() {
        return height;
    }

    // Setter for height
    public void setHeight(double height) {
        this.height = height;
    }

    // Getter for weight
    public double getWeight() {
        return weight;
    }

    // Setter for weight
    public void setWeight(double weight) {
        this.weight = weight;
    }

    // Method to calculate BMI
    public double calculateBMI() {
        return weight / (height * height); // BMI formula
    }
}

public class BMIProgram {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        // Asking user for height and weight
        System.out.print("Enter your height in meters: ");
        double height = sc.nextDouble();

        System.out.print("Enter your weight in kilograms: ");
        double weight = sc.nextDouble();

        // Creating an object of BMICalculator
        BMICalculator bmiCalc = new BMICalculator(height, weight);

        // Printing the calculated BMI
        System.out.println("Your BMI is: " + bmiCalc.calculateBMI());

        sc.close(); // Closing scanner
    }
}
```

Q6. Electricity Bill Calculation – Java Program

Design a Java program to calculate the electricity bill for a customer based on the number of units consumed. Implement a class named ElectricityBill with the following specifications:

Class: ElectricityBill

Instance Variables

- customerName (String): Name of the customer
- unitsConsumed (double): Number of electricity units consumed
- billAmount (double): The calculated bill amount

Constructor

- A parameterized constructor to initialize the customerName and unitsConsumed fields.

Method

- void calculateBillAmount(): This method calculates the electricity bill amount based on the following tariff rules:

  - First 100 units: Rs. 5 per unit
  - Next 200 units (i.e., 101 to 300): Rs. 7 per unit
  - Remaining units (above 300): Rs. 10 per unit

Main Program

In the main() method:

CDAC Mumbai

1. Create an object of the ElectricityBill class.
2. Set the customerName and unitsConsumed values (can be taken from user input or hardcoded).
3. Call the calculateBillAmount() method to compute the bill.
4. Display the customer's name, units consumed, and final bill amount.

```
import java.util.Scanner;
class ElectricityBill{
        private String Name;
        private double unitsConsumed;
        private double billAmount;

        public ElectricityBill(String Name,double unitsConsumed){
                this.Name= Name;
                this.unitsConsumed = unitsConsumed;
        }
        public double calculateBillAmount(){
                double Amount = 0;
                if(unitsConsumed >= 100){
                         Amount = 5* unitsConsumed;
                        return Amount;
                }
                if(unitsConsumed > 100 && unitsConsumed <= 300){
                         Amount = 7 * unitsConsumed;
                        return Amount;
                }
                if(unitsConsumed > 300){
                         Amount = 10 * unitsConsumed;
                        return Amount;
                }
                return Amount;
        }
}

class ElectricityBillCalculatio{
        public static void main(String[] args){
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter name of customer= ");
                String Name = sc.next();
                System.out.println("Enter units consumed= ");
                double unitsConsumed = sc.nextDouble();

                ElectricityBill e = new ElectricityBill(Name, unitsConsumed);
                System.out.println("Bill Amount of = " +Name +" is "+e.calculateBillAmount());


        }

}
```