

## Assignment 4

### *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.

```
class Room {  
  
    private double height, width, 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;  
    }  
}  
  
public class RoomDemo {  
    public static void main(String[] args) {  
        Room room1 = new Room(10, 12, 15);  
        Room room2 = new Room(8, 10, 12);  
  
        System.out.println("Volume of Room 1: " + room1.volume());  
        System.out.println("Volume of Room 2: " + room2.volume());  
    }  
}
```

```
}
```

```
C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignmen  
t 4>javac RoomDemo.java
```

```
C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignmen  
t 4>java RoomDemo  
Volume of Room 1: 1800.0  
Volume of Room 2: 960.0
```

## 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

```
class Student {  
  
    String name;  
  
    int marks1, marks2, marks3;  
  
    // Method to assign initial values  
    void setDetails(String studentName, int m1, int m2, int m3) {  
        name = studentName;  
        marks1 = m1;  
        marks2 = m2;  
        marks3 = m3;  
    }  
  
    // Method to compute total and average marks  
    int getTotalMarks() {  
        return marks1 + marks2 + marks3;  
    }  
  
    double getAverageMarks() {  
        return getTotalMarks() / 3.0;  
    }  
}
```

```
// Method to display student details

void displayDetails() {

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

    System.out.println("Total Marks: " + getTotalMarks());

    System.out.println("Average Marks: " + getAverageMarks());

}

}
```

```
// Main class to demonstrate the Student class
```

```
public class StudentDemo {

    public static void main(String[] args) {

        Student student1 = new Student();

        student1.setDetails("Alice", 85, 90, 80);

        student1.displayDetails();

        Student student2 = new Student();

        student2.setDetails("Bob", 75, 88, 92);

        student2.displayDetails();

    }

}
```

```
C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignmen
t 4>javac StudentDemo.java

C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignmen
t 4>java StudentDemo
Student Name: Alice
Total Marks: 255
Average Marks: 85.0
Student Name: Bob
Total Marks: 255
Average Marks: 85.0
```

### 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.

```
class Box {  
    double height, width, breadth;  
  
    // Constructor to initialize the box dimensions  
    Box(double h, double w, double b) {  
        height = h;  
        width = w;  
        breadth = b;  
    }  
  
    // Method to calculate volume  
    double getVolume() {  
        return height * width * breadth;  
    }  
  
    // Method to calculate surface area  
    double getArea() {  
        return 2 * (height * width + width * breadth + height * breadth);  
    }  
}  
  
// Main class to test the Box class  
public class BoxDemo {  
    public static void main(String[] args) {
```

```

// Creating two box objects with different dimensions

Box box1 = new Box(5, 4, 3);

Box box2 = new Box(7, 6, 2);


// Displaying volume and surface area for each box

System.out.println("Box 1:");

System.out.println("Volume: " + box1.getVolume());

System.out.println("Surface Area: " + box1.getArea());


System.out.println("\nBox 2:");

System.out.println("Volume: " + box2.getVolume());

System.out.println("Surface Area: " + box2.getArea());

}

}

```

```

C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignmen
t 4>javac BoxDemo.java

C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignmen
t 4>java BoxDemo
Box 1:
Volume: 60.0
Surface Area: 94.0

Box 2:
Volume: 84.0
Surface Area: 136.0

```

#### 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)  
    Complex() {  
        this.real = 0;  
        this.imaginary = 0;  
    }  
  
    // Constructor to initialize real part only (imaginary = 0)  
    Complex(double real) {  
        this.real = real;  
        this.imaginary = 0;  
    }  
  
    // Constructor to initialize both real and imaginary parts  
    Complex(double real, double imaginary) {  
        this.real = real;  
        this.imaginary = imaginary;  
    }  
  
    // Method to add two complex numbers  
    Complex add(Complex other) {  
        return new Complex(this.real + other.real, this.imaginary + other.imaginary);  
    }  
  
    // Method to multiply two complex numbers  
    Complex multiply(Complex other) {  
        double newReal = (this.real * other.real) - (this.imaginary * other.imaginary);  
        double newImaginary = (this.real * other.imaginary) + (this.imaginary * other.real);  
    }  
}
```

```

        return new Complex(newReal, newImaginary);
    }

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

// Main class to demonstrate complex number operations
public class ComplexDemo {
    public static void main(String[] args) {
        // Creating two 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);
        System.out.print("Sum: ");
        sum.display();

        // Multiplying the two complex numbers
        Complex product = c1.multiply(c2);
        System.out.print("Product: ");
        product.display();
    }
}

```

```

C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignmen
t 4>javac ComplexDemo.java

C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignmen
t 4>java ComplexDemo
Sum: 7.0 + 0.0i
Product: 16.0 + 2.0i

```

### 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:

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:

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

$\text{BMI} = (\text{height} \times \text{height}) \text{ weight}$

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  
  
    // Constructor to initialize height and weight  
    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);
    }
}

// Main class to demonstrate BMI calculation
```

```
public class BMIDemo {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
  
        // Taking input from the user  
        System.out.print("Enter your height in meters: ");  
        double height = scanner.nextDouble();  
        System.out.print("Enter your weight in kilograms: ");  
        double weight = scanner.nextDouble();  
  
        // Creating an object of BMICalculator  
        BMICalculator bmiCalculator = new BMICalculator(height, weight);  
  
        // Calculating and displaying BMI  
        double bmi = bmiCalculator.calculateBMI();  
        System.out.println("Your BMI is: " + bmi);  
  
        // Checking BMI category  
        if (bmi < 18.5) {  
            System.out.println("Category: Underweight");  
        } else if (bmi >= 18.5 && bmi < 24.9) {  
            System.out.println("Category: Normal weight");  
        } else if (bmi >= 25 && bmi < 29.9) {  
            System.out.println("Category: Overweight");  
        } else {  
            System.out.println("Category: Obese");  
        }  
  
        scanner.close();  
    }  
}
```

```

C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignmen
t 4>javac BMIDemo.java

C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignmen
t 4>java BMIDemo
Enter your height in meters: 163
Enter your weight in kilograms: 50
Your BMI is: 0.0018818924310286425
Category: Underweight

C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignmen

```

## 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:

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 customerName;  
    private double unitsConsumed;  
    private double billAmount;  
  
    // Constructor to initialize customer name and units consumed  
    ElectricityBill(String customerName, double unitsConsumed) {  
        this.customerName = customerName;  
        this.unitsConsumed = unitsConsumed;  
        this.billAmount = 0; // Initialize bill amount to 0  
    }  
  
    // Method to calculate the electricity bill  
    public void calculateBillAmount() {  
        if (unitsConsumed <= 100) {  
            billAmount = unitsConsumed * 5;  
        } else if (unitsConsumed <= 300) {  
            billAmount = (100 * 5) + ((unitsConsumed - 100) * 7);  
        } else {  
            billAmount = (100 * 5) + (200 * 7) + ((unitsConsumed - 300) * 10);  
        }  
    }  
  
    // Method to display the bill details  
    public void displayBill() {  
        System.out.println("\nElectricity Bill Details:");  
        System.out.println("Customer Name: " + customerName);  
        System.out.println("Units Consumed: " + unitsConsumed);  
        System.out.println("Total Bill Amount: Rs. " + billAmount);  
    }  
}
```

```
}
```

```
// Main class to demonstrate electricity bill calculation
```

```
public class ElectricityBillDemo {
```

```
    public static void main(String[] args) {
```

```
        Scanner scanner = new Scanner(System.in);
```

```
        // Taking user input
```

```
        System.out.print("Enter Customer Name: ");
```

```
        String name = scanner.nextLine();
```

```
        System.out.print("Enter Units Consumed: ");
```

```
        double units = scanner.nextDouble();
```

```
        // Creating object of ElectricityBill class
```

```
        ElectricityBill bill = new ElectricityBill(name, units);
```

```
        // Calculating bill amount
```

```
        bill.calculateBillAmount();
```

```
        // Displaying bill details
```

```
        bill.displayBill();
```

```
    }
```

```
}
```

```
C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignmen
t 4>java ElectricityBillDemo
Enter Customer Name: aman
Enter Units Consumed: 26

Electricity Bill Details:
Customer Name: aman
Units Consumed: 26.0
Total Bill Amount: Rs. 130.0

C:\Users\prajy\OneDrive\Desktop\feb 25\java\java codes\Assignments\Assignmen
```