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.

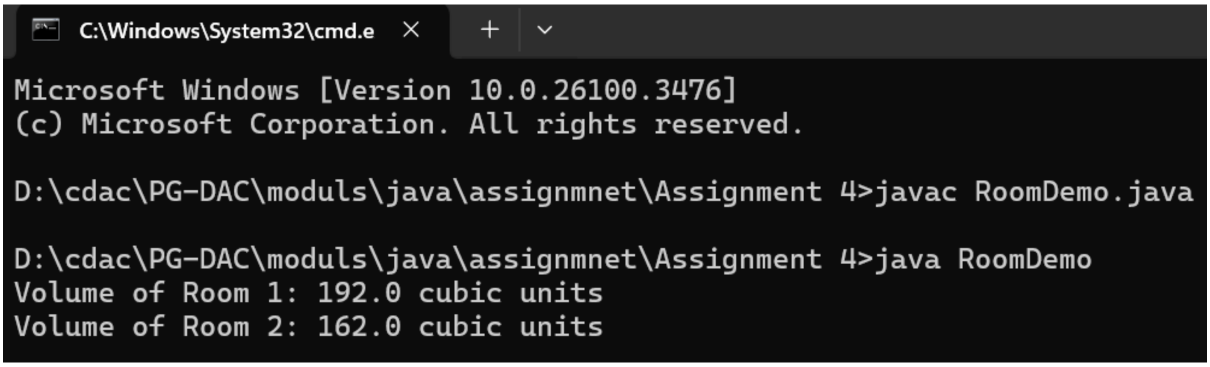
Ans:

Input:

```
class Room{
	int height;
	int width;
	int breadth;
	Room(int height,int width,int breadth){
		this.height=height;
		this.width=width;
		this.breadth=breadth;
	}
	double volume(){
		return height*width*breadth;
	}
}
class RoomDemo{
	public static void main(String[] args){
		Room r= new Room(8,4,6);
		Room r1= new Room(9,3,6);
		System.out.println("Volume of Room 1: " + r.volume() + " cubic units");
		System.out.println("Volume of Room 2: " + r1.volume() + " cubic units");
	}
```

}

Output:



```
Microsoft Windows [Version 10.0.26100.3476]
(c) Microsoft Corporation. All rights reserved.

D:\cdac\PG-DAC\moduls\java\assignmnet\Assignment 4>javac RoomDemo.java

D:\cdac\PG-DAC\moduls\java\assignmnet\Assignment 4>java RoomDemo
Volume of Room 1: 192.0 cubic units
Volume of Room 2: 162.0 cubic units
```

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**

Write a main() method to demonstrate the functionality of the class.

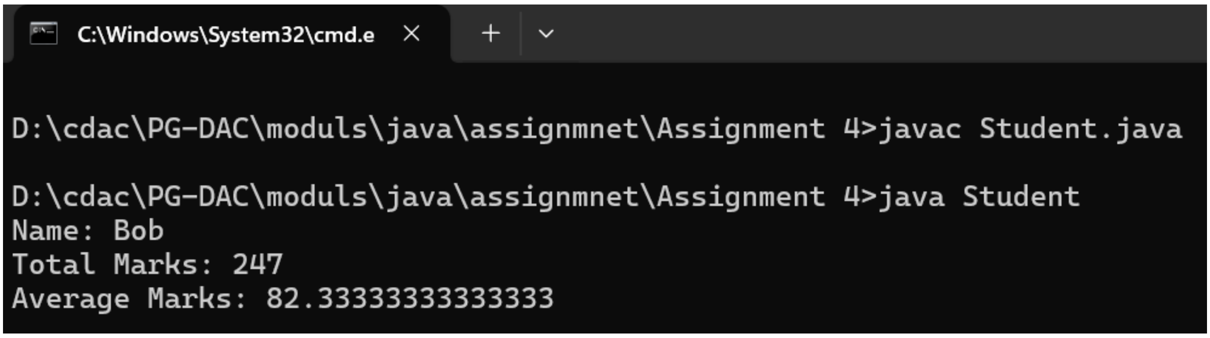
Ans:

Input:

```
class Student{
        String Name;
        int M1;
        int M2;
        int M3;
        void assign(String Name,int M1,int M2,int M3){
                this.Name=Name;
                this.M1=M1;
                this.M2=M2;
                this.M3=M3;
```

```
    }
    int computeTotal(){
        return M1+M2+M3;
    }
    double computeAvg(){
        return computeTotal()/3.0;
    }
    void display(){
        System.out.println("Name: "+Name);
        System.out.println("Total Marks: "+computeTotal());
        System.out.println("Average Marks: "+computeAvg());
    }
    public static void main(String[] args){
        Student s=new Student();
        s.assign("Bob",99,98,50);
        s.display();


    }
}
```

Output:



```
D:\cdac\PG-DAC\moduls\java\assignmnet\Assignment 4>javac Student.java

D:\cdac\PG-DAC\moduls\java\assignmnet\Assignment 4>java Student
Name: Bob
Total Marks: 247
Average Marks: 82.33333333333333
```

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.

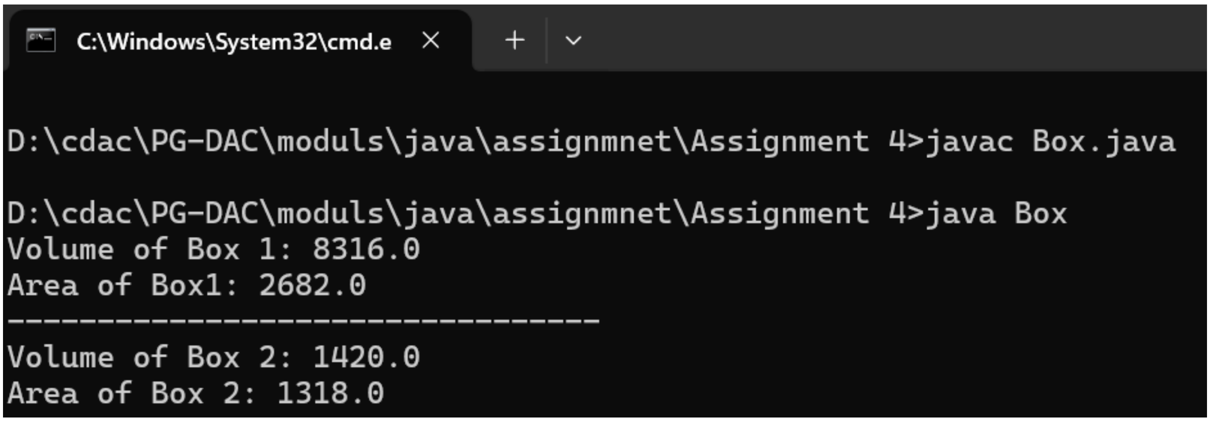
Ans:

Input:

```
class Box{
        int height;
        int width;
        int breadth;

        Box(int height,int width,int breadth){
                this.height = height;
                this.width = width;
                this.breadth = breadth;
        }
        double getVolume(){
                return height*width*breadth;
        }
        double getArea(){
                double Area= 2*((height * width) + (height * breadth) + (width * breadth));
                return Area;
        }
        public static void main(String[] args){
                Box B1= new Box(12,21,33);
                Box B2= new Box(71,5,4);
                System.out.println("Volume of Box 1: "+B1.getVolume());
                System.out.println("Area of Box1: "+B1.getArea());
                System.out.println("--------------------------------");
```

```
                    System.out.println("Volume of Box 2: "+B2.getVolume());

                    System.out.println("Area of Box 2: "+B2.getArea());

          }

}
```

Output:



```
D:\cdac\PG-DAC\moduls\java\assignmnet\Assignment 4>javac Box.java

D:\cdac\PG-DAC\moduls\java\assignmnet\Assignment 4>java Box
Volume of Box 1: 8316.0
Area of Box1: 2682.0
---------------------------------
Volume of Box 2: 1420.0
Area of Box 2: 1318.0
```

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**

Ans:

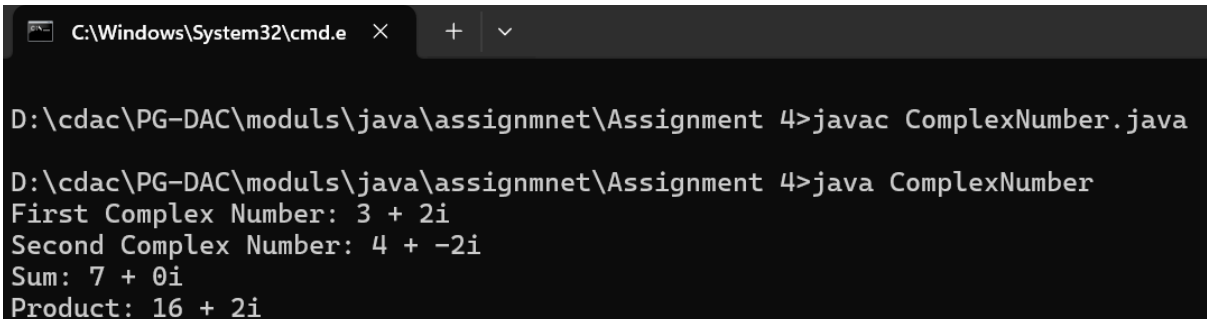
Input:

```
class ComplexNumber{

          int real;
```

```
}
       int img;
       ComplexNumber(){
                 real=0;
                 img=0;
       }
       ComplexNumber(int real){
                 this.real=real;
                 img=0;
       }
       ComplexNumber(int real,int img){
                 this.real=real;
                 this.img=img;
       }
       ComplexNumber add(ComplexNumber other){
                 return new ComplexNumber(this.real + other.real, this.img + other.img);
       }
       public ComplexNumber multiply(ComplexNumber other) {
   int realPart = this.real * other.real - this.img * other.img;
   int imgPart = this.real * other.img + this.img * other.real;
   return new ComplexNumber(realPart, imgPart);
}
       @Override
public String toString() {
   return String.format("%d + %di", real, img);
}
       public static void main(String[] args){
                 ComplexNumber num1 = new ComplexNumber(3, 2);
   ComplexNumber num2 = new ComplexNumber(4, -2);
```

```
        ComplexNumber sum = num1.add(num2);

        ComplexNumber product = num1.multiply(num2);

        System.out.println("First Complex Number: " + num1);

        System.out.println("Second Complex Number: " + num2);

        System.out.println("Sum: " + sum);

        System.out.println("Product: " + product);

            }

}
```

Output:



```
D:\cdac\PG-DAC\moduls\java\assignmnet\Assignment 4>javac ComplexNumber.java

D:\cdac\PG-DAC\moduls\java\assignmnet\Assignment 4>java ComplexNumber
First Complex Number: 3 + 2i
Second Complex Number: 4 + -2i
Sum: 7 + 0i
Product: 16 + 2i
```

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:**

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.**

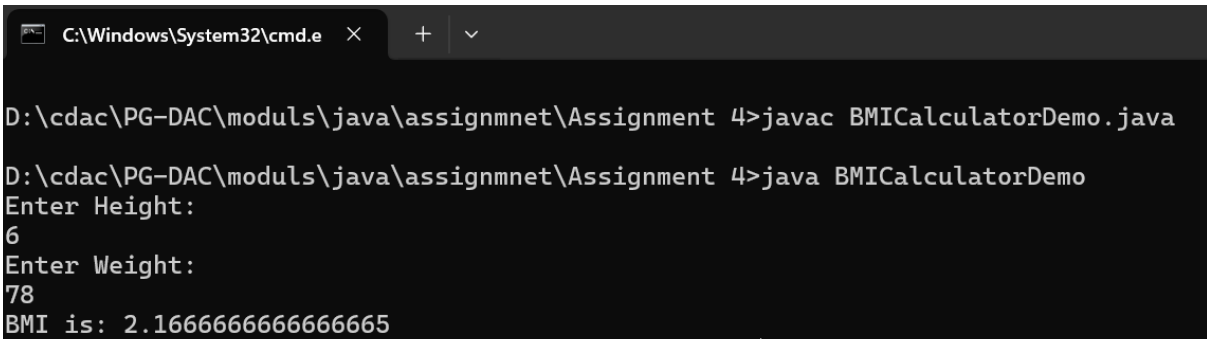
Ans:

Input:

```
import java.util.*;
class BMICalculator{
        double height;
        double weight;
        BMICalculator(double height,double weight){
                this.height=height;
                this.weight=weight;
        }
        double getHeight(){
                return height;
        }
        void setHeight(double height){
                this.height=height;
        }
        double getWeight(){
                return weight;
        }
        void setWeight(double weight){
                this.weight=weight;
```

```
    }
    double calculateBMI(){
        return weight / (height * height);
    }
}
class BMICalculatorDemo {
    public static void main(String[] args){
        Scanner input= new Scanner(System.in);
        System.out.println("Enter Height: ");
        double height=input.nextDouble();
        System.out.println("Enter Weight: ");
        double weight=input.nextDouble();
        BMICalculator e=new BMICalculator(height,weight);
        double bmiResult= e.calculateBMI();
        System.out.print("BMI is: "+bmiResult);
    }
}
```

Output:



```
D:\cdac\PG-DAC\moduls\java\assignmnet\Assignment 4>javac BMICalculatorDemo.java

D:\cdac\PG-DAC\moduls\java\assignmnet\Assignment 4>java BMICalculatorDemo
Enter Height:
6
Enter Weight:
78
BMI is: 2.1666666666666665
```

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.**

Ans:

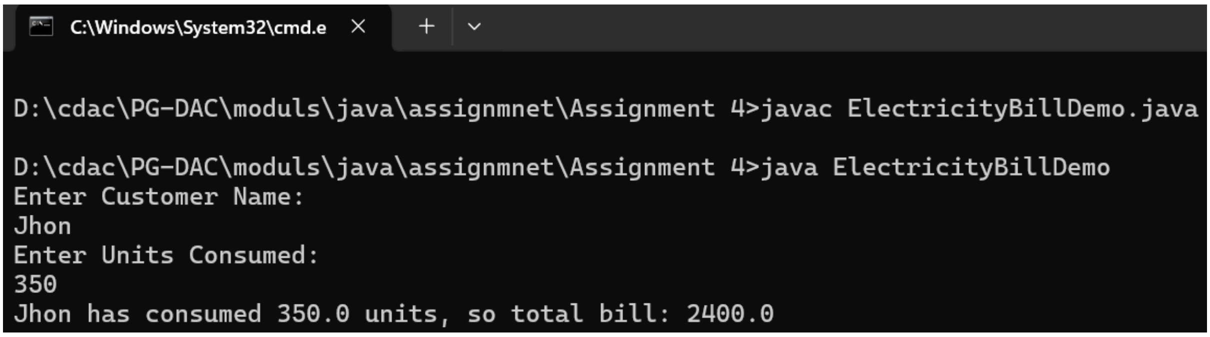
Input:

```
import java.util.*;
class ElectricityBill{
        String customerName;
        double unitsConsumed;
```

```
    double billAmount;
    ElectricityBill(String customerName,double unitsConsumed){
        this.customerName=customerName;
        this.unitsConsumed=unitsConsumed;
    }
    double 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);
  }
        return billAmount;
    }
}
class ElectricityBillDemo{
    public static void main(String[] args){
        Scanner input=new Scanner(System.in);
        System.out.println("Enter Customer Name: ");
        String customerName= input.nextLine();
        System.out.println("Enter Units Consumed: ");
        double unitsConsumed= input.nextDouble();
```

```
                ElectricityBill e = new ElectricityBill(customerName,unitsConsumed);

                double BIll=e.calculateBillAmount();

                System.out.println(customerName+" has consumed "+unitsConsumed+" units, so total bill: "+BIll);

        }

}
```

Output:



```
D:\cdac\PG-DAC\moduls\java\assignmnet\Assignment 4>javac ElectricityBillDemo.java

D:\cdac\PG-DAC\moduls\java\assignmnet\Assignment 4>java ElectricityBillDemo
Enter Customer Name:
Jhon
Enter Units Consumed:
350
Jhon has consumed 350.0 units, so total bill: 2400.0
```