Assignment 4 Class & Objects

Name Akshay Janrao

PRN 250240520006

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 Room Demo that creates instances of the Room class and displays the volume for each instance.

Answer

```
class Room{
    static int height;
    static int width;
    static int length;

    static int Volume(){

    return height*width*length;
    }
}

class RoomDemo extends Room{

 public static void main(String args[]){

 Room r1 = new Room();
 Room r2 = new Room();
 Room r3 = new Room();

  r1.height = 4;
  r2.width = 4;
 r3.length = 5;

 System.out.println("the volume of the Room is "+Volume()+"m^3");

 }
 }
```

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.

Answer

```java
/*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.
*/
import java.util.*;
   class StudentAvg{

    private String name;
      private double p,c,m,b;

      public StudentAvg(String N1,double a,double q,double w,double e){

       this.name = N1;
         this.p = a;
        this.c = q;
       this.m = w;
        this.b = e;

      }

      public double Total(){
         return p + c + b + m ;

         }

       public double Avg(){

       return (double)(p + c + b + m)/4;
       }

       void display(){
           System.out.println("Name  "+name);
           System.out.println("Total marks "+Total());
           System.out.println("Average marks "+Avg());
       }


       public static void main(String args[]){

           StudentAvg s1 = new StudentAvg("Bhau ", 65.7,45.55,87.2,78.2);

           s1.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

```java
/*  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{

    private int height;
    private int length;
    private int width;

    public  Box(int h,int l,int w){

 this.height = h;
this.length = l;
this.width = w;

}

    public int getVolume(){

return height*length*width;
}
    public int getArea(){

return 2*(length*width + length*height +width*height);
}
    void display(){
    System.out.println("The volume of box is "+getVolume());
    System.out.println("The Area of box is "+getArea());
    }
public static void main(String args[]){

    Box b1 = new Box(65,75,88);
    b1.display();
    }
}
```

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

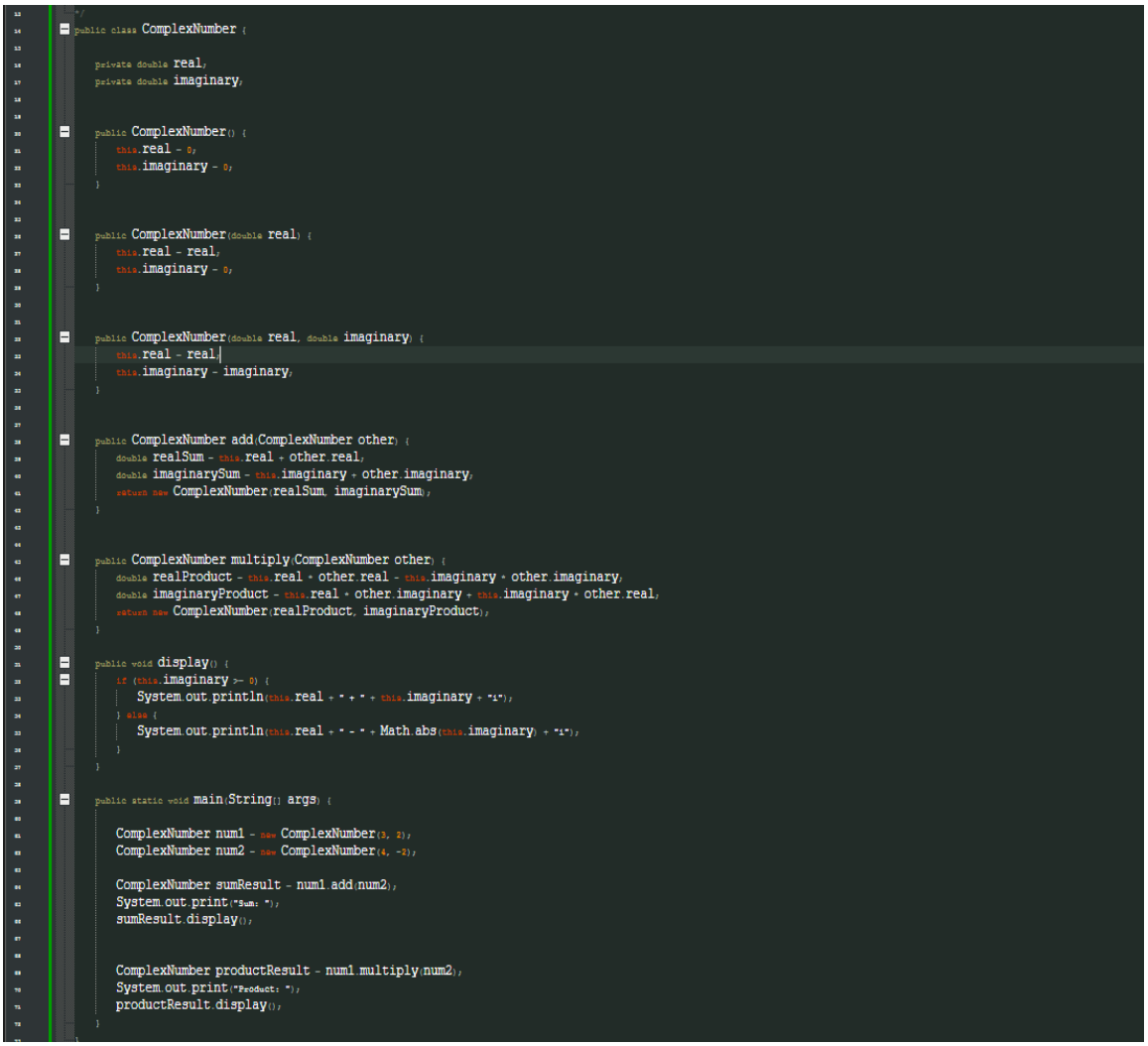
Also, write member functions to:

● 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



```
*/
public class ComplexNumber {

    private double real;
    private double imaginary;


    public ComplexNumber() {
        this.real = 0;
        this.imaginary = 0;
    }


    public ComplexNumber(double real) {
        this.real = real;
        this.imaginary = 0;
    }


    public ComplexNumber(double real, double imaginary) {
        this.real = real;
        this.imaginary = imaginary;
    }


    public ComplexNumber add(ComplexNumber other) {
        double realSum = this.real + other.real;
        double imaginarySum = this.imaginary + other.imaginary;
        return new ComplexNumber(realSum, imaginarySum);
    }


    public ComplexNumber multiply(ComplexNumber other) {
        double realProduct = this.real * other.real - this.imaginary * other.imaginary;
        double imaginaryProduct = this.real * other.imaginary + this.imaginary * other.real;
        return new ComplexNumber(realProduct, imaginaryProduct);
    }

    public void display() {
        if (this.imaginary >= 0) {
            System.out.println(this.real + " + " + this.imaginary + "i");
        } else {
            System.out.println(this.real + " - " + Math.abs(this.imaginary) + "i");
        }
    }

    public static void main(String[] args) {

        ComplexNumber num1 = new ComplexNumber(3, 2);
        ComplexNumber num2 = new ComplexNumber(4, -2);

        ComplexNumber sumResult = num1.add(num2);
        System.out.print("Sum: ");
        sumResult.display();


        ComplexNumber productResult = num1.multiply(num2);
        System.out.print("Product: ");
        productResult.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: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.

```java
import java.util.*;
class BMICalculator {

    private double height;
    private double weight;


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

    public double getHeight() {
        return height;
    }

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


    public double getWeight() {
        return weight;
    }

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

    public double calculateBMI() {
        return weight / (height * height);
    }

public class BMICalculatorM{
    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);


        System.out.print("Enter your height in meters: ");
        double height = scanner.nextDouble();

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


        BMICalculator bmiCalculator = new BMICalculator(height, weight);


        double bmi = bmiCalculator.calculateBMI();
        System.out.printf("Your BMI is: %.2f\n", bmi);


    }
```

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.

```java
import java.util.Scanner;

class ElectricityBill {

    String customerName;
    double unitsConsumed;
    double billAmount;

    public ElectricityBill(String customerName, double unitsConsumed) {
        this.customerName = customerName;
        this.unitsConsumed = unitsConsumed;
        this.billAmount = 0;
    }

    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);
        }
    }

    public void displayBill() {
        System.out.println("Customer Name: " + customerName);
        System.out.println("Units Consumed: " + unitsConsumed);
        System.out.println("Total Bill Amount: Rs. " + billAmount);
    }

    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);


        System.out.print("Enter customer name: ");
        String customerName = scan.nextLine();

        System.out.print("Enter units consumed: ");
        double unitsConsumed = scan.nextDouble();

        ElectricityBill bill = new ElectricityBill(customerName, unitsConsumed);

        bill.calculateBillAmount();

        bill.displayBill();


    }
}
```