

Name: Aarya Sanjay Dange

Assignment 04 (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:

```
import java.util.Scanner;
class Room{
    int height;
    int width;
    int breadth;
    Room(int height,int width,int breadth){
        this.height=height;
        this.width=width;
        this.breadth=breadth;
    }
    void volume(){
        System.out.println("Volume of a room:
"+(height*width*breadth));
    }
}
class RoomDemo{
    public static void main(String[] args) {
```

```
Room r1=new Room(10,20,30);
    Room r2=new Room(5,6,7);
    r1.volume();
    r2.volume();
}
```

```
D:\CDAC\OOP Java>javac RoomDemo.java
```

```
D:\CDAC\OOP Java>java RoomDemo
```

```
Volume of a room: 6000
```

```
Volume of a room: 210
```

```
D:\CDAC\OOP 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.

Ans:

```
import java.util.Scanner;
class Student{
```

```

String name;
int mark1,mark2,mark3;
int breadth;
Student(String name,int mark1,int mark2,int mark3){
    this.name=name;
    this.mark1=mark1;
    this.mark2=mark2;
    this.mark3=mark3;
}
String calculation(){
    return " Total marks: "+(mark1+mark2+mark3)+" and
Average: "+(mark1+mark2+mark3)/2;
}
void display(){
    System.out.println("Student name:
"+name+calculation());
}
}
class StudentDemo{
    public static void main(String[] args) {
        Student s1=new Student("arya",10,20,30);
        Student s2=new Student("shweta",50,60,70);
        s1.display();
        s2.display();
    }
}

```

```
D:\CDAC\OOP Java>javac StudentDemo.java
```

```
D:\CDAC\OOP Java>java StudentDemo
```

```
Student name: arya Total marks: 60 and Average: 30
```

```
Student name: shweta Total marks: 180 and Average: 90
```

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:

```
class Box{
    int height;
    int width;
    int breadth;
    Box(int height,int width,int breadth){
        this.height=height;
        this.width=width;
        this.breadth=breadth;
    }
    void getVolume(){
        System.out.println("Volume of Box:
"+(height*width*breadth));
    }
    void getArea(){
        System.out.println("Area of Box:
"+(2*((height*width)+(width*breadth)+(height*breadth))));
    }
}
```

```
public static void main(String[] args) {  
    Box r1=new Box(10,20,30);  
        Box r2=new Box(5,6,7);  
        r1.getVolume();  
        r1.getArea();  
        r2.getVolume();  
        r2.getArea();  
    }  
}
```

```
D:\CDAC\OOP Java>javac Box.java
```

```
D:\CDAC\OOP Java>java Box
```

```
Volume of Box: 6000
```

```
Area of Box: 2200
```

```
Volume of Box: 210
```

```
Area of Box: 214
```

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:

```
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) {
        return new ComplexNumber(this.real + other.real,
this.imaginary + other.imaginary);
    }

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

```

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

```

```

public class ComplexNumberDemo {
    public static void main(String[] args) {
        ComplexNumber c1 = new ComplexNumber(3, 2);
        ComplexNumber c2 = new ComplexNumber(4, -2);

        ComplexNumber sum = c1.add(c2);
        ComplexNumber product = c1.multiply(c2);

        System.out.print("Sum: ");
        sum.display();

        System.out.print("Product: ");
        product.display();
    }
}

```

```
D:\CDAC\OOP Java>javac ComplexNumberDemo.java
```

```
D:\CDAC\OOP Java>java ComplexNumberDemo
```

```
Sum: 7.0 + 0.0i
```

```
Product: 16.0 + 2.0i
```

```
D:\CDAC\OOP Java>|
```

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.

Ans:

```
class BMICalculator{
    private double height,weight;

    public BMICalculator(double height,double weight){
        this.height=height;
        this.weight=weight;
    }
    public void setHeight(double height){
        this.height=height;
    }
    public void setWeight(double weight){
        this.weight=weight;
    }
    public void calculateBMI(){
        double bmi=weight/(height*height);
        System.out.println("BMI: "+bmi);
    }
}

class BMICalculatorDemo{
    public static void main(String[] args){
        BMICalculator b1=new BMICalculator(1.75,70);
        b1.calculateBMI();
    }
}
```

```
D:\CDAC\OOP Java>javac BMICalculatorDemo.java
```

```
D:\CDAC\OOP Java>java BMICalculatorDemo
BMI: 22.857142857142858
```

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:

```
class ElectricityBill{
    String customerName;
    double unitsConsumed;
    double billAmount;

    public ElectricityBill(String customerName,double
unitsConsumed){
        this.customerName=customerName;
        this.unitsConsumed=unitsConsumed;
    }
    double calculateBillAmount(){
        if(unitsConsumed<=100){
            return (5*unitsConsumed) ;
        }
        else if(unitsConsumed>100 && unitsConsumed<=300){
            return ((100*5)+((unitsConsumed-100)*7)) ;
        }
        else{
            return ((100*5)+(200*7)+(unitsConsumed-300)*10)
;
        }
    }
}
```

```
    }  
}  
class ElectricityBillDemo{  
    public static void main(String[] args){  
        ElectricityBill b1=new ElectricityBill("arya",600);  
        System.out.println(b1.unitsConsumed+" Units consumed "+"  
by "+b1.customerName+". so bill amount is:  
"+b1.calculateBillAmount());  
    }  
}
```

```
D:\CDAC\OOP Java>javac ElectricityBillDemo.java  
  
D:\CDAC\OOP Java>java ElectricityBillDemo  
600.0 Units consumed by arya. so bill amount is: 4900.0  
  
D:\CDAC\OOP Java>|
```