**Nishant Patil**

**PG-DAC**

**Java**

**Assignment 4**

**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 {

double height, width, breadth;

Room(double height, double width, double breadth) {

this.height = height;

this.width = width;

this.breadth = breadth;

}

double volume() {

return height \* width \* breadth;

}

}

public class Main {

public static void main(String[] args) {

Room room1 = new Room(10, 12, 15);

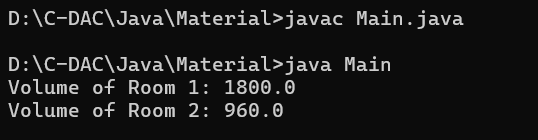
Room room2 = new Room(8, 10, 12);

System.out.println("Volume of Room 1: " + room1.volume());

System.out.println("Volume of Room 2: " + room2.volume());

}

}



**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.**

class Student {

String name;

int marks1, marks2, marks3;

void setValues(String name, int m1, int m2, int m3) {

this.name = name;

this.marks1 = m1;

this.marks2 = m2;

this.marks3 = m3;

}

int getTotalMarks() {

return marks1 + marks2 + marks3;

}

double getAverageMarks() {

return getTotalMarks() / 3.0;

}

void display() {

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

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

}

}

public class Main {

public static void main(String[] args) {

Student s1 = new Student();

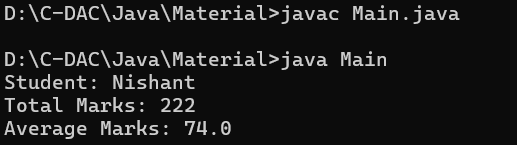
s1.setValues("Nishant", 75, 80, 67);

s1.display();

System.out.println("Average Marks: " + s1.getAverageMarks());

}

}



**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;

Box(double height, double width, double breadth) {

this.height = height;

this.width = width;

this.breadth = breadth;

}

double getVolume() {

return height \* width \* breadth;

}

double getArea() {

return 2 \* (width \* breadth + breadth \* height + height \* width);

}

}

public class BoxDemo {

public static void main(String[] args) {

Box box1 = new Box(5, 6, 7);

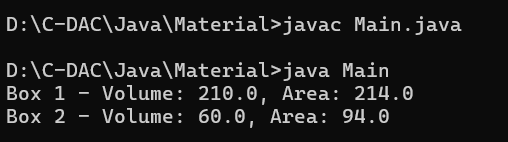
Box box2 = new Box(3, 4, 5);

System.out.println("Box 1 - Volume: " + box1.getVolume() + ", Area: " + box1.getArea());

System.out.println("Box 2 - Volume: " + box2.getVolume() + ", Area: " + box2.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 {

double real, imaginary;

Complex() {

this.real = 0;

this.imaginary = 0;

}

Complex(double real) {

this.real = real;

this.imaginary = 0;

}

Complex(double real, double imaginary) {

this.real = real;

this.imaginary = imaginary;

}

Complex add(Complex c) {

return new Complex(this.real + c.real, this.imaginary + c.imaginary);

}

Complex multiply(Complex c) {

return new Complex(this.real \* c.real - this.imaginary \* c.imaginary,

this.real \* c.imaginary + this.imaginary \* c.real);

}

void display() {

System.out.println(real + " + " + imaginary + "i");

}

}

public class ComplexDemo {

public static void main(String[] args) {

Complex c1 = new Complex(3, 2);

Complex c2 = new Complex(4, -2);

Complex sum = c1.add(c2);

Complex product = c1.multiply(c2);

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

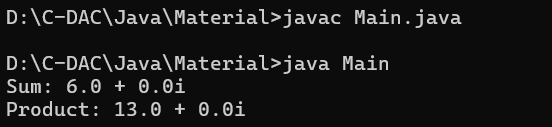
sum.display();

System.out.print("Product: ");

product.display();

}

}



**Q.6 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.**

import java.util.Scanner;

class BMICalculator {

private double height, weight;

BMICalculator(double height, double weight) {

this.height = height;

this.weight = weight;

}

void setHeight(double height) {

this.height = height;

}

void setWeight(double weight) {

this.weight = weight;

}

double calculateBMI() {

return weight / (height \* height);

}

}

public class BMIDemo {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter height (m): ");

double height = sc.nextDouble();

System.out.print("Enter weight (kg): ");

double weight = sc.nextDouble();

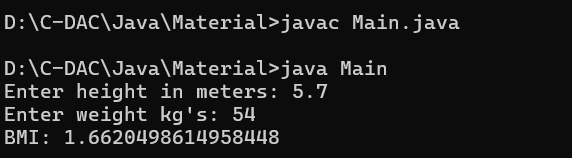
BMICalculator bmi = new BMICalculator(height, weight);

System.out.println("BMI: " + bmi.calculateBMI());

sc.close();

}

}



**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 {

String customerName;

double unitsConsumed, billAmount;

ElectricityBill(String customerName, double unitsConsumed) {

this.customerName = customerName;

this.unitsConsumed = unitsConsumed;

}

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);

}

}

void displayBill() {

System.out.println("Customer: " + customerName);

System.out.println("Units Consumed: " + unitsConsumed);

System.out.println("Bill Amount: Rs. " + billAmount);

}

}

public class ElectricityBillDemo {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.print("Enter customer name: ");

String name = sc.nextLine();

System.out.print("Enter units consumed: ");

double units = sc.nextDouble();

ElectricityBill bill = new ElectricityBill(name, units);

bill.calculateBillAmount();

bill.displayBill();

sc.close();

}

}

