

# Rajalakshmi Engineering College

Name: RAKSHITHA R  
Email: 240701418@rajalakshmi.edu.in  
Roll no: 240701418  
Phone: 7305274265  
Branch: REC  
Department: CSE - Section 8  
Batch: 2028  
Degree: B.E - CSE

Scan to verify results



## 2024\_28\_III\_OOPS Using Java Lab

### REC\_2028\_OOPS using Java\_Week 7\_CY

Attempt : 1  
Total Mark : 40  
Marks Obtained : 40

#### **Section 1 : Coding**

##### **1. Problem Statement**

Alex and Bob are designing a control system for household appliances, and one of the appliances is a washing machine. You want to create a program to help them that models the washing machine as a motor and calculates its electricity consumption based on its capacity.

Define an interface named Motor with the following methods:

`void run() double consume(double capacity)`

Create a class called WashingMachine that implements the Motor interface.

In the WashingMachine class:

Implement the run() method to print "Washing machine is

running."Implement a consume() method to print "Washing machine is consuming electricity."Implement the consume(double capacity) method to calculate the electricity consumption (in kWh) of the washing machine based on its capacity. The formula for electricity consumption is (capacity \* 0.05).

### ***Input Format***

The input consists of a double value representing the capacity of the washing machine in kW.

### ***Output Format***

The first line of output prints "Washing machine is running."

The second line prints "Washing machine is consuming electricity."

The third line prints "Electricity consumption: X kWh" where X is a double value, rounded off to two decimal places, representing the electricity consumption.

Refer to the sample output for formatting specifications.

### ***Sample Test Case***

Input: 2.5

Output: Washing machine is running.

Washing machine is consuming electricity.

Electricity consumption: 0.13 kWh

### ***Answer***

```
import java.util.Scanner;
```

```
interface Motor {  
    void run();  
    double consume(double capacity);  
}
```

```
class WashingMachine implements Motor {
```

```
    public void run() {  
        System.out.println("Washing machine is running.");  
    }
```

```

public void consume() {
    System.out.println("Washing machine is consuming electricity.");
}

public double consume(double capacity) {
    double consumption = capacity * 0.05;
    return consumption;
}

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        WashingMachine washingMachine = new WashingMachine();

        double capacity = scanner.nextDouble();

        washingMachine.run();
        washingMachine.consume();

        double consumption = washingMachine.consume(capacity);
        System.out.printf("Electricity consumption: %.2f kWh", consumption);

        scanner.close();
    }
}

```

**Status :** Correct

**Marks :** 10/10

## 2. Problem Statement

Maria, an online store owner, is looking to implement a pricing system that calculates the final price of products after applying discounts. She needs a program that takes the original price of a product and the discount percentage as input and computes the final discounted price. The discount is applied as a percentage of the original price. Maria wants to ensure that the final price is formatted to display exactly two decimal places.

Implement this functionality using the PriceCalculator interface and the DiscountCalculator class.

### ***Input Format***

The first line of input consists of the original price (a double value).

The second line of input consists of a discount percentage (a double value).

### ***Output Format***

The output displays the final price after the discount, adhering to the following format: "Final Price after discount: \$[final\_price]".

Here, [final\_price] should be replaced with the calculated final price, formatted as a currency value with two decimal places.

Refer to the sample output for the formatting specifications.

### ***Sample Test Case***

Input: 100.0

10.0

Output: Final Price after discount: \$90.00

### ***Answer***

```
import java.util.Scanner;

interface PriceCalculator {
    double calculatePrice(double originalPrice, double discount);
}

class DiscountCalculator implements PriceCalculator {
    public double calculatePrice(double originalPrice, double discount) {
        double discountedPrice = originalPrice - (originalPrice * discount / 100);
        return discountedPrice;
    }
}

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

```
Scanner scanner = new Scanner(System.in);
double originalPrice = scanner.nextDouble();
double discount = scanner.nextDouble();
PriceCalculator calculator = new DiscountCalculator();
double finalPrice = calculator.calculatePrice(originalPrice, discount);
System.out.printf("Final Price after discount: $%.2f%n", finalPrice); //
Formats output to 2 decimal places
}
```

**Status :** Correct

**Marks :** 10/10

### 3. Problem Statement:

Rathish is planning a road trip and needs a program to convert speeds between miles per hour (MPH) and kilometers per hour (KPH).

Create an interface, SpeedConverter, with a method convertSpeed(double mph). Implement the interface with MPHtoKPHConverter class, allowing Rathish to input MPH and receive the converted speed in KPH, rounded to two decimal points.

Formula: Speed in KPH = 1.60934 \* Speed in MPH.

#### **Input Format**

The input consists of a single double-point number representing the speed in miles per hour (MPH).

#### **Output Format**

The output displays the converted speed (double-point number) in kilometers per hour (KPH) rounded off to two decimal points in the following format:

"Speed in KPH: <<converted speed>>".

Refer to the sample output for the formatting specifications.

#### **Sample Test Case**

Input: 1.0

Output: Speed in KPH: 1.61

**Answer**

```
import java.util.Scanner;

interface SpeedConverter {
    double convertSpeed(double mph);
}

class MPHtoKPHConverter implements SpeedConverter {
    public double convertSpeed(double mph) {
        return mph * 1.60934;
    }
}

class SpeedConversionApp {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        double speedInMPH = scanner.nextDouble();

        SpeedConverter converter = new MPHtoKPHConverter();

        double speedInKPH = converter.convertSpeed(speedInMPH);

        System.out.printf("Speed in KPH: %.2f\n", speedInKPH);

        scanner.close();
    }
}
```

**Status :** Correct

**Marks :** 10/10

#### 4. Problem Statement:

Sam is developing a geometry application and needs a class for trapezoid calculations. Create a "Trapezoid" class implementing a "ShapeInput" interface with a method to input trapezoid dimensions.

Also, implement a "ShapeCalculator" interface with methods to compute

area and perimeter. In the "Main" class, instantiate Trapezoid, gather user input, and display the calculated area and perimeter with two decimal places.

#### Note

Area of Trapezoid =  $(1/2) * (base1 + base2) * height$

Perimeter of Trapezoid =  $base1 + base2 + side1 + side2$

#### ***Input Format***

The first line of input is a double-point value representing base1 of the trapezoid.

The second line of input is a double-point value representing base2 of the trapezoid.

The third line of input is a double-point value representing the height of the trapezoid.

The fourth line of input is a double-point value representing side1 of the trapezoid.

The fifth line of input is a double-point value representing side2 of the trapezoid.

#### ***Output Format***

The output displays the two lines of the calculated area (double type) and perimeter (double type) of the trapezoid, each rounded to two decimal places in the following format:

"Area of the Trapezoid: <<calculated area>>".

Perimeter of the Trapezoid: <<calculated perimeter>>".

Refer to the sample output for the formatting specifications.

#### ***Sample Test Case***

Input: 1.0

2.0

1.0

3.0

1.0

Output: Area of the Trapezoid: 1.50

Perimeter of the Trapezoid: 7.00

### **Answer**

```
import java.util.Scanner;
```

```
interface ShapeInput {  
    void getInput();  
}
```

```
interface ShapeCalculator {  
    double calculateArea();  
    double calculatePerimeter();  
}
```

```
class Trapezoid implements ShapeInput, ShapeCalculator {  
    private double base1, base2, height, side1, side2;
```

```
    public void getInput() {  
        Scanner scanner = new Scanner(System.in);
```

```
        base1 = scanner.nextDouble();  
        base2 = scanner.nextDouble();  
        height = scanner.nextDouble();  
        side1 = scanner.nextDouble();  
        side2 = scanner.nextDouble();  
    }
```

```
    public double calculateArea() {  
        return 0.5 * height * (base1 + base2);  
    }
```

```
    public double calculatePerimeter() {  
        return base1 + base2 + side1 + side2;  
    }  
}
```

```
public class Main {
```



```
public static void main(String[] args) {  
    Trapezoid trapezoid = new Trapezoid();  
    trapezoid.getInput();  
  
    double area = trapezoid.calculateArea();  
    double perimeter = trapezoid.calculatePerimeter();  
  
    System.out.println("Area of the Trapezoid: " + String.format("%.2f", area));  
    System.out.println("Perimeter of the Trapezoid: " + String.format("%.2f",  
perimeter));  
    }  
}
```

**Status :** Correct

**Marks : 10/10**