

Rajalakshmi Engineering College

Name: Anu shalini B
Email: 240701043@rajalakshmi.edu.in
Roll no: 240701043
Phone: 9486176983
Branch: REC
Department: CSE - Section 10
Batch: 2028
Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 7_Q1

Attempt : 1
Total Mark : 10
Marks Obtained : 0

Section 1 : Coding

1. Problem Statement:

Rajiv is analyzing the energy consumption in his household and wants to calculate the total cost based on the daily energy usage. He is given the rate per unit of electricity and the energy consumed for multiple days. To structure this calculation efficiently, he decides to use an interface-based approach.

Implement an interface CostCalculator with the necessary methods to retrieve energy details and compute the cost. The calculations should be handled in the EnergyConsumptionTracker class, while the EnergyConsumptionApp class should only handle input and output.

Formula

Energy Cost for one day = Energy Consumed per day * Rate Per Unit

Input Format

The first line of input consists of the rate per unit as an 'R' (a double value).

The second line of input consists of the number of days 'N' (an integer).

The third line of input consists of the daily energy consumption values for each day 'D' (double values), separated by space.

Output Format

The first line of the output prints: "Day-wise Energy Cost:"

The next N lines of the output print the day-wise energy costs(double type) and the total energy cost (double type) in Indian Rupees in the following format: "Day [day_number]: Rs. [energy_cost]"

The last line of the output prints: "Total Energy Cost: Rs. [total_cost]"

Note: energy_cost and total_cost are rounded off to two decimal points

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 0.01

3

10.0 20.0 30.0

Output: Day-wise Energy Cost:

Day 1: Rs. 0.10

Day 2: Rs. 0.20

Day 3: Rs. 0.30

Total Energy Cost: Rs. 0.60

Answer

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

```
import java.util.*;
```

```
interface CostCalculator {  
    void getEnergyDetails(double rate, int days, double[] consumption);  
    void computeCost();  
}
```

```
class EnergyConsumptionTracker implements CostCalculator {  
    private double rate;  
    private int days;  
    private double[] consumption;  
    private double[] dailyCosts;  
    private double totalCost;
```

```
    public void getEnergyDetails(double rate, int days, double[] consumption) {  
        this.rate = rate;  
        this.days = days;  
        this.consumption = consumption;  
        this.dailyCosts = new double[days];  
    }
```

```
    public void computeCost() {  
        totalCost = 0.0;  
        System.out.println("Day-wise Energy Cost:");  
        for (int i = 0; i < days; i++) {  
            dailyCosts[i] = consumption[i] * rate;  
            totalCost += dailyCosts[i];  
            System.out.printf("Day %d: Rs. %.2f\n", (i + 1), dailyCosts[i]);  
        }  
        System.out.printf("Total Energy Cost: Rs. %.2f", totalCost);  
    }  
}
```

```
class EnergyConsumptionApp {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
  
        double ratePerUnit = scanner.nextDouble();  
        int numDays = scanner.nextInt();  
  
        CostCalculator tracker = new EnergyConsumptionTracker(ratePerUnit,  
numDays);
```

```
        tracker.getEnergyDetails(scanner);  
        tracker.calculateAndDisplayCost();  
        scanner.close();  
    }  
}
```

Status : Wrong

Marks : 0/10

Rajalakshmi Engineering College

Name: Anu shalini B
Email: 240701043@rajalakshmi.edu.in
Roll no: 240701043
Phone: 9486176983
Branch: REC
Department: CSE - Section 10
Batch: 2028
Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

REC_2028_OOPS using Java_Week 7_MCQ

Attempt : 1
Total Mark : 15
Marks Obtained : 11

Section 1 : MCQ

1. If a class implements two interfaces that have the same default method, what must the class do?

Answer

The class must override the method to resolve ambiguity.

Status : Correct

Marks : 1/1

2. What is the output of the following code?

```
interface A {  
    default void show() {  
        System.out.println("A's Default Method");  
    }  
}
```

```
class B {  
    public void show() {  
        System.out.println("B's Method");  
    }  
}
```

```
class C extends B implements A {  
}
```

```
public class Main {  
    public static void main(String[] args) {  
        C obj = new C();  
        obj.show();  
    }  
}
```

Answer

B's Method

Status : Correct

Marks : 1/1

3. Which of the following statements about Java interfaces is true?

Answer

A class can implement multiple classes and interfaces together.

Status : Wrong

Marks : 0/1

4. What is the primary purpose of static methods in Java interfaces?

Answer

They allow an interface to provide helper methods without requiring an implementing class.

Status : Correct

Marks : 1/1

5. What is the output of the following code?

```
interface A {  
    static void display() {  
        System.out.println("Static method in A");  
    }  
}
```

```
class B implements A {  
    static void display() {  
        System.out.println("Static method in B");  
    }  
}
```

```
public class Main {  
    public static void main(String[] args) {  
        B.display();  
    }  
}
```

Answer

Static method in B

Status : Correct

Marks : 1/1

6. Which of the following statements is true regarding default methods in Java interfaces?

Answer

A default method can be overridden in a class implementing the interface.

Status : Correct

Marks : 1/1

7. What happens when an implementing class does not override a default method from an interface?

Answer

The default method's implementation from the interface will be used.

Status : Correct

Marks : 1/1

8. Consider a class implementing an interface and extending a class, both having a method with the same name. Which method gets called?

Answer

The method from the interface

Status : Wrong

Marks : 0/1

9. How do you call a static method from an interface MyInterface?

Answer

MyInterface.staticMethod();

Status : Correct

Marks : 1/1

10. Which of the following is the correct way to declare an interface in Java?

Answer

public interface Vehicle { public void start() {};

Status : Wrong

Marks : 0/1

11. What is the output of the following code?

```
interface X {  
    default void show() {  
        System.out.println("X's Default Method");  
    }  
}
```

```
interface Y {  
    default void show() {  
        System.out.println("Y's Default Method");  
    }  
}
```



```
class Z implements X, Y {  
    public void show() {  
        System.out.println("Z's Method");  
    }  
}
```

```
public class Main {  
    public static void main(String[] args) {  
        Z obj = new Z();  
        obj.show();  
    }  
}
```

Answer

Z's Method

Status : Correct

Marks : 1/1

12. What is the output of the following code?

```
interface A {  
    default void show() {  
        System.out.println("A's Default Method");  
    }  
}
```

```
interface B {  
    default void show() {  
        System.out.println("B's Default Method");  
    }  
}
```

```
class C implements A, B {  
    public void show() {  
        A.super.show();  
    }  
}
```

```
public class Main {
```

```
public static void main(String[] args) {  
    C obj = new C();  
    obj.show();  
}  
}
```

Answer

A's Default Method

Status : Correct

Marks : 1/1

13. Can a Java interface contain both default and static methods?

Answer

Yes, an interface can have both default and static methods.

Status : Correct

Marks : 1/1

14. How can a class explicitly call a default method from an interface if there is a naming conflict?

Answer

Using InterfaceName.methodName();

Status : Wrong

Marks : 0/1

15. What is the output of the following code?

```
interface MathOperations {  
    static int square(int x) {  
        return x * x;  
    }  
}  
  
public class Main {  
    public static void main(String[] args) {  
        System.out.println(MathOperations.square(5));  
    }  
}
```

}

Answer

25

Status : Correct

Marks : 1/1

Rajalakshmi Engineering College

Name: Anu shalini B
Email: 240701043@rajalakshmi.edu.in
Roll no: 240701043
Phone: 9486176983
Branch: REC
Department: CSE - Section 10
Batch: 2028
Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 7_Q2

Attempt : 1
Total Mark : 10
Marks Obtained : 0

Section 1 : Coding

1. Problem Statement

Jaheer is working on a health monitoring system to help individuals calculate their Body Mass Index (BMI). He has implemented a basic BMI calculator and an interface called HealthCalculator. It should have a method called calculateBMI.

You are tasked with creating a program that takes weight and height as input, calculates the BMI using the BMICalculator class, and displays the result. If the height or weight is less than or equal to zero, then return -1.

Formula: $BMI = \text{weight} / (\text{height} * \text{height})$

Input Format

The first line of input consists of a double value W, the person's weight in kilograms.

The second line consists of a double value H, the height of the person in meters.

Output Format

The output displays "BMI: " followed by a double value, representing the calculated BMI, rounded off to two decimal places.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 70.0

175

Output: BMI: 22.86

Answer

```
import java.util.Scanner;

interface HealthCalculator{
    double calculateBMI(double weight,double height);
}

class BMICalculator implements HealthCalculator{
    public double BMICalculate (double weight,double height){
        if (weight <=0 || height <=0)return -1;
        return weight/(height * height);
    }
}

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

        double weight = scanner.nextDouble();
        double height = scanner.nextDouble();

        BMICalculator bmiCalculator = new BMICalculator();
        double bmi = bmiCalculator.calculateBMI(weight, height);

        System.out.printf("BMI: %.2f\n", bmi);
    }
}
```

240701043

```
    scanner.close();  
  }  
}
```

240701043

240701043

240701043

Status : Wrong

Marks : 0/10

240701043

240701043

240701043

240701043

240701043

240701043

240701043

240701043

240701043

240701043

240701043

240701043

Rajalakshmi Engineering College

Name: Anu shalini B
Email: 240701043@rajalakshmi.edu.in
Roll no: 240701043
Phone: 9486176983
Branch: REC
Department: CSE - Section 10
Batch: 2028
Degree: B.E - CSE

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 7_Q3

Attempt : 1
Total Mark : 10
Marks Obtained : 0

Section 1 : Coding

1. Problem Statement

A financial analyst, Alex, needs a program to calculate simple interest for various financial transactions. He requires a straightforward tool that takes in the principal amount, interest rate, and time in years and computes the interest.

The formula to be used is: $\text{Interest} = \text{Principal} \times \text{Rate} \times \text{Time} / 100$

Implement this functionality using the InterestCalculator interface and the SimpleInterestCalculator class.

Input Format

The first line of input consists of the principal amount P as a double value.

The second line of input consists of the annual interest rate r as a double value.

The third line of input consists of the number of years t as a positive integer, which is an integer value.

Output Format

The output displays the calculated simple interest in the following format: "Simple Interest: [interest_value]", Here, [interest_value] should be replaced with the actual interest value calculated by the program.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 1000.00

5.00

2

Output: Simple Interest: 100.0

Answer

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

```
interface InterestCalculator {  
    double calculateInterest(double principal, double rate, int time);  
}
```

```
class SimpleInterestCalculator implements InterestCalculator {  
    public double calculateInterest(double principal, double rate, int time) {  
        return (principal * rate * time) / 100;  
    }  
}
```

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



```
double rate = scanner.nextDouble();  
int time = scanner.nextInt();  
InterestCalculator calculator = new SimpleInterestCalculator();  
double interest = calculator.simpleInterest(principal, rate, time);  
System.out.println("Simple Interest: " + interest);  
}  
}
```

Status : Wrong

Marks : 0/10