Rajalakshmi Engineering College

Name: Punith Kumar.S
Email: 241501155@rajalakshmi.edu.in
Roll no: 241501155
Phone: 9600149411
Branch: REC
Department: AI & ML - Section 3
Batch: 2028
Degree: B.E - AI & ML

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 : 10

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: Interest = Principal × Rate × 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 simpleInterest(double principal, double rate, int time);
}

class SimpleInterestCalculator implements InterestCalculator {
   public double simpleInterest(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 : Correct ***Marks : 10/10***

Rajalakshmi Engineering College

Name: Punith Kumar.S
Email: 241501155@rajalakshmi.edu.in
Roll no: 241501155
Phone: 9600149411
Branch: REC
Department: AI & ML - Section 3
Batch: 2028
Degree: B.E - AI & ML

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 : 10

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 = weight / (height * 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
1.75

Output: BMI: 22.86

Answer

```
import java.util.Scanner;


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

class BMICalculator implements HealthCalculator {
   public double calculateBMI(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);


        scanner.close();
    }
}
```

Status : Correct ***Marks : 10/10***

Rajalakshmi Engineering College

Name: Punith Kumar.S
Email: 241501155@rajalakshmi.edu.in
Roll no: 241501155
Phone: 9600149411
Branch: REC
Department: AI & ML - Section 3
Batch: 2028
Degree: B.E - AI & ML

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 : 10

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;
```

```
interface CostCalculator {
    void getEnergyDetails(Scanner scanner);
    void calculateAndDisplayCost();
}


class EnergyConsumptionTracker implements CostCalculator {
    private double ratePerUnit;
    private int numDays;
    private double[] energyConsumed;

    public EnergyConsumptionTracker(double ratePerUnit, int numDays) {
        this.ratePerUnit = ratePerUnit;
        this.numDays = numDays;
        this.energyConsumed = new double[numDays];
    }

    @Override
    public void getEnergyDetails(Scanner scanner) {
        for (int i = 0; i < numDays; i++) {
            energyConsumed[i] = scanner.nextDouble();
        }
    }

    @Override
    public void calculateAndDisplayCost() {
        double totalCost = 0.0;

        System.out.println("Day-wise Energy Cost:");
        for (int i = 0; i < numDays; i++) {
            double dailyCost = energyConsumed[i] * ratePerUnit;
            totalCost += dailyCost;
            System.out.printf("Day %d: Rs. %.2f%n", i + 1, dailyCost);
        }

        System.out.printf("Total Energy Cost: Rs. %.2f%n", 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 : Correct ***Marks : 10/10***