

Rajalakshmi Engineering College

Name: samyuktha ss

Email: 241001217@rajalakshmi.edu.in

Roll no:

Phone: 7539908242

Branch: REC

Department: IT - Section 2

Batch: 2028

Degree: B.E - IT

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 : 14

Section 1 : MCQ

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

Answer

MyInterface.staticMethod();

Status : Correct

Marks : 1/1

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

Answer

interface Vehicle { void start();}

Status : Correct

Marks : 1/1

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

Answer

A class can implement multiple interfaces.

Status : Correct

Marks : 1/1

4. 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

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

Answer

Using InterfaceName.super.methodName();

Status : Correct

Marks : 1/1

6. 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

7. 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

8. 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

9. 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

10. 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

11. 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

12. 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 superclass

Status : Correct

Marks : 1/1

13. 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

14. 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

15. 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

A's Default Method

Status : Wrong

Marks : 0/1

Rajalakshmi Engineering College

Name: samyuktha ss

Email: 241001217@rajalakshmi.edu.in

Roll no:

Phone: 7539908242

Branch: REC

Department: IT - Section 2

Batch: 2028

Degree: B.E - IT

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 7_Q5

Attempt : 1

Total Mark : 10

Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Raj is curious about how old he is in the current year.

He has asked you to create a simple program that calculates a person's age based on their birth year. You decide to implement this functionality using the AgeCalculator interface and the HumanAgeCalculator class.

Note: The current year is 2024. Calculate the current age by using the formula: current year - birth year.

Input Format

The input consists of an integer representing the birth year.

Output Format

The output displays "You are X years old." where X is an integer representing the calculated age based on the entered birth year.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 1934

Output: You are 90 years old.

Answer

```
import java.util.Scanner;

interface AgeCalculator {
    int calculateAge(int birthYear);
}

class HumanAgeCalculator implements AgeCalculator {
    public int calculateAge(int birthYear) {
        return 2024 - birthYear;
    }
}

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

        AgeCalculator ageCalculator = new HumanAgeCalculator();

        int birthYear = scanner.nextInt();
        int age = ageCalculator.calculateAge(birthYear);

        System.out.println("You are " + age + " years old.");
    }
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: samyuktha ss

Email: 241001217@rajalakshmi.edu.in

Roll no:

Phone: 7539908242

Branch: REC

Department: IT - Section 2

Batch: 2028

Degree: B.E - IT

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

John is developing a car loan calculator and has structured his program using two interfaces, Principal and InterestRate, defining methods for principal and interest rate retrieval.

The Loan class implements these interfaces, taking principal and annual interest rates as parameters. User input is solicited for these values, and the program ensures their validity before performing calculations. If input values are invalid (less than or equal to zero), an error message is displayed.

Note: Total interest = principal * interest rate * years

Input Format

The first line of input consists of a double value P, representing the principal.

The second line consists of a double value R, representing the annual interest rate.

The third line consists of an integer value N, representing the loan duration in years.

Output Format

If the input values are valid, print "Total interest paid: Rs. " followed by a double value, representing the total interest paid, rounded off to two decimal places.

If the input values are invalid (negative or zero values for principal, annual interest rate, or loan duration), print "Invalid input values!".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 20000.00
0.05
5

Output: Total interest paid: Rs.5000.00

Answer

```
import java.util.Scanner;

interface Principal {
    double getPrincipal();
}

interface InterestRate {
    double getInterestRate();
}

class Loan implements Principal, InterestRate {

    double principal;
    double rate;
```

```

Loan(double principal, double rate) {
    this.principal = principal;
    this.rate = rate;
}

public double getPrincipal() {
    return principal;
}

public double getInterestRate() {
    return rate;
}

public double calculateTotalInterest(int years) {
    return principal * rate * years;
}
}

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

        double carPrice = scanner.nextDouble();

        double annualInterestRate = scanner.nextDouble();

        int loanDuration = scanner.nextInt();

        if (carPrice <= 0 || annualInterestRate <= 0 || loanDuration <= 0) {
            System.out.println("Invalid input values!");
            return;
        }

        Loan carLoan = new Loan(carPrice, annualInterestRate);
        double totalInterest = carLoan.calculateTotalInterest(loanDuration);

        System.out.printf("Total interest paid: Rs.%2f%n", totalInterest);
    }
}

```

Status : Correct

Marks : 10/10

2. 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.print("Washing machine is running. ");
    }

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

    public double consume(double capacity) {
        // the main prints the returned value, so do NOT print here
        return capacity * 0.05;
    }
}

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

3. 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 discountPercentage);
}

class DiscountCalculator implements PriceCalculator {

    public double calculatePrice(double originalPrice, double discountPercentage)
    {
        double discountAmount = originalPrice * (discountPercentage / 100.0);
        return originalPrice - discountAmount;
    }
}

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

4. Problem Statement

Jeevan is developing a fitness-tracking application to monitor daily physical activity.

The application incorporates a FitnessTracker class that implements two interfaces: StepCounter for tracking the number of steps taken and CalorieCalculator for estimating total calories burned based on total steps.

Jeevan needs your help creating a program.

Note

The calorie calculation formula is: Total caloriesBurned = (total steps / 100.0) * 20.0.

Input Format

The first line of input is an integer n, representing the number of days Jeevan wants to input data.

The second line consists of space-separated integers, representing the number of steps Jeevan took on each day.

Output Format

The first line of output prints: "Total Steps: <totalSteps>", where '<totalSteps>' is the sum of steps (integer) taken over 'n' days.

The second line prints: "Calories Burned: <caloriesBurned>", where '<caloriesBurned>' is the estimated total calories (double-point number) burned based on the total steps taken rounded off to two decimal places.

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 3
340 234 987

Output: Total Steps: 1561
Calories Burned: 312.20

Answer

```
import java.util.Scanner;  
  
interface StepCounter {  
    void countSteps(int steps);  
    int getTotalSteps();  
}
```

```
interface CalorieCalculator {
    double calculateCaloriesBurned(int totalSteps);
}

class FitnessTracker implements StepCounter, CalorieCalculator {

    int total = 0;

    public void countSteps(int steps) {
        total += steps;
    }

    public int getTotalSteps() {
        return total;
    }

    public double calculateCaloriesBurned(int totalSteps) {
        return (totalSteps / 100.0) * 20.0;
    }
}

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

        FitnessTracker tracker = new FitnessTracker();

        int n = scanner.nextInt();

        for (int i = 0; i < n; i++) {
            int steps = scanner.nextInt();
            tracker.countSteps(steps);
        }

        int totalSteps = tracker.getTotalSteps();
        System.out.println("Total Steps: " + totalSteps);

        double caloriesBurned = tracker.calculateCaloriesBurned(totalSteps);
        System.out.printf("Calories Burned: %.2f%n", caloriesBurned);
    }
}
```

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

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: samyuktha ss

Email: 241001217@rajalakshmi.edu.in

Roll no:

Phone: 7539908242

Branch: REC

Department: IT - Section 2

Batch: 2028

Degree: B.E - IT

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 sc);  
    void calculateAndDisplayCost();  
}  
  
class EnergyConsumptionTracker implements CostCalculator {  
    private double ratePerUnit;  
    private int numDays;  
    private double[] consumptions;  
  
    public EnergyConsumptionTracker(double rate, int days) {  
        this.ratePerUnit = rate;  
        this.numDays = days;  
        this.consumptions = new double[days];  
    }  
  
    @Override  
    public void getEnergyDetails(Scanner sc) {  
        for (int i = 0; i < numDays; i++) {  
            consumptions[i] = sc.nextDouble();  
        }  
    }  
  
    @Override  
    public void calculateAndDisplayCost() {  
        System.out.println("Day-wise Energy Cost:");  
        double total = 0.0;  
        for (int i = 0; i < numDays; i++) {  
            double cost = consumptions[i] * ratePerUnit;  
            total += cost;  
            System.out.printf("Day %d: Rs. %.2f%n", i + 1, cost);  
        }  
        System.out.printf("Total Energy Cost: Rs. %.2f", total);  
    }  
}  
  
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

Rajalakshmi Engineering College

Name: samyuktha ss

Email: 241001217@rajalakshmi.edu.in

Roll no:

Phone: 7539908242

Branch: REC

Department: IT - Section 2

Batch: 2028

Degree: B.E - IT

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: samyuktha ss

Email: 241001217@rajalakshmi.edu.in

Roll no:

Phone: 7539908242

Branch: REC

Department: IT - Section 2

Batch: 2028

Degree: B.E - IT

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

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

1.75

Output: BMI: 22.86

Answer

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

```
interface HealthCalculator {  
    double calculateBMI(double weight, double height);  
}  
  
class BMICalculator implements HealthCalculator {  
    @Override  
    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