

# Rajalakshmi Engineering College

Name: Gnaana Kumaar  
Email: 241001062@rajalakshmi.edu.in  
Roll no: 241001062  
Phone: 9790249960  
Branch: REC  
Department: IT - Section 4  
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{
    public double calculateBMI(double weight,double height){
        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: Gnaana Kumaar  
Email: 241001062@rajalakshmi.edu.in  
Roll no: 241001062  
Phone: 9790249960  
Branch: REC  
Department: IT - Section 4  
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:  $\text{Interest} = \text{Principal} \times \text{Rate} \times \text{Time} / 100$

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;
```

```
// You are using Java
```

```
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: Abil kumar  
Email: 241001003@rajalakshmi.edu.in  
Roll no: 241001003  
Phone: 6369113570  
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 1\_PAH

Attempt : 1  
Total Mark : 40  
Marks Obtained : 40

#### Section 1 : Coding

##### 1. PROBLEM STATEMENT:

Maria, a software developer, is working on a project to create a simple program to determine which of two integers is closest to zero. The integers can be either positive or negative. The program needs to take two integer inputs and calculate which one is closer to zero. If both integers are equidistant from zero, the program should return 0.

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

The input contains two lines:

The first line of the input contains an integer, which can be either a positive or a negative integer.

The second line of the input contains an integer, which can be either a positive or a negative integer.

### **Output Format**

The output displays the integer that is closest to zero in the following format:

"The integer closest to zero is: [closest\_integer]"

Here, [closest\_integer] should be replaced with the integer that is closer to zero based on its absolute value.

Refer to the sample output for the formatting specifications.

### **Sample Test Case**

Input: 5

8

Output: The integer closest to zero is: 5

### **Answer**

// You are using Java

import java.io.\*;

import java.util.Scanner;

class Main {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int num1 = sc.nextInt();

int num2 = sc.nextInt();

int closest;

if (Math.abs(num1) < Math.abs(num2)) {

closest = num1;

} else if (Math.abs(num2) < Math.abs(num1)) {



```
        closest = num2;
    } else {
        closest = 0;
    }

    System.out.println("The integer closest to zero is: " + closest);

    sc.close();
}
}
```

**Status :** Correct

**Marks :** 10/10

## 2. PROBLEM STATEMENT:

Maria, a software developer, is working on a program to determine if two given integers which can be either positive or negative integers have the same parity (both even or both odd). She needs your help in writing this program.

Write a program that takes two integers as input and checks if both integers are either even or odd.

### ***Input Format***

The input consists of two lines:

The first line consists of an integer (input1) which can be either positive or negative.

The second line consists of an integer (input2) which can be either positive or negative.

### ***Output Format***

The output is displayed in the following format:

If both integers have the same parity (i.e., both even or both odd), print:

"Both integers are either even or odd"

Otherwise, print:

"The integers have different parities"

Refer to the sample output for the formatting specifications.

### **Sample Test Case**

Input: 2  
-4

Output: Both integers are either even or odd

### **Answer**

// You are using Java

import java.io.\*;

import java.util.Scanner;

class Main {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int I1 = sc.nextInt();

int I2 = sc.nextInt();

if ((I1 % 2 == 0 && I2 % 2 == 0) || (I1 % 2 != 0 && I2 % 2 != 0)) {

System.out.println("Both integers are either even or odd");

} else {

System.out.println("The integers have different parities");

}

sc.close();

}

}

**Status : Correct**

**Marks : 10/10**

### 3. Problem Statement

In the Kingdom of Delivery Logistics, there is a giant truck used for transporting packages across the kingdom. The truck has a maximum capacity represented by an integer, and each package also has a specific weight. The truck's efficiency and safety depend on whether the weight of the package is below a certain threshold.

The kingdom's delivery service has a rule: if the weight of a package is less than one-third of the truck's total capacity, the package is eligible for quick processing and dispatch. However, if the weight is too heavy, the package will require special handling.

As a logistics manager, you need to check whether the weight of the package is less than one-third of the truck's total capacity.

Write a program using a ternary operator that helps determine whether the package weight meets the requirement for quick processing or if it needs special handling.

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

The first line of input consists of an integer  $p$ , representing the weight of the package.

The second line consists of an integer  $w$ , representing the total weight capacity of the truck.

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

The first line of output prints "One-third of Truck: X," where X is one-third of the truck's total weight capacity as a double value with two decimal places.

The second line of output displays one of the following:

1. If  $p$  is less than one-third of the truck's total weight capacity, print "Package weight is less than one-third of the truck's capacity".
2. Otherwise, print "Package weight is not less than one-third of the truck's capacity".

Refer to the sample output for the formatting specifications.

### **Sample Test Case**

Input: 13

60

Output: One-third of Truck: 20.00

Package weight is less than one-third of truck's capacity

### **Answer**

// You are using Java

```
import java.io.*;
```

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

```
class Main {
```

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

```
        Scanner sc = new Scanner(System.in);
```

```
        int p = sc.nextInt();
```

```
        int w = sc.nextInt();
```

```
        double oneThird = w / 3.0;
```

```
        System.out.printf("One-third of Truck: %.2f\n", oneThird);
```

```
        String result = (p < oneThird)
```

```
            ? "Package weight is less than one-third of truck's capacity"
```

```
            : "Package weight is not less than one-third of truck's capacity";
```

```
        System.out.println(result);
```

```
    }  
}
```

**Status : Correct**

**Marks : 10/10**

## **4. Problem Statement**

Mickey and Miney are walking through a magical forest. The forest is full

of enchanted stones, each with a unique number. There is a legend that says the magic power of the stones can be revealed by using a special operation. To determine the magic power of a given stone, you need to perform a bitwise AND operation with the number 15.

Each stone's number is represented by an integer, and Mickey needs to find the magic power of each stone by applying this operation.

Your task is to help Mickey compute the result of the bitwise AND operation of the given stone number with 15, and print the result.

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

The input consists of a single integer.

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

The output should display a single integer, which is the result of the bitwise AND operation between input and 15.

Refer to the sample output for format specifications.

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

Input: 25

Output: 9

#### ***Answer***

```
// You are using Java
import java.io.*;

import java.util.Scanner;

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

        int num = sc.nextInt();
        int result = num & 15;
```

```
        System.out.println(result);
    }
    sc.close();
}
```

**Status :** Correct

**Marks :** 10/10

# Rajalakshmi Engineering College

Name: Gnaana Kumaar  
Email: 241001062@rajalakshmi.edu.in  
Roll no: 241001062  
Phone: 9790249960  
Branch: REC  
Department: IT - Section 4  
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;
// You are using Java
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: Gnaana Kumaar

Email: 241001062@rajalakshmi.edu.in

Roll no: 241001062

Phone: 9790249960

Branch: REC

Department: IT - Section 4

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 : 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, double[] consumption);  
    void calculateCost();  
}
```

```
class EnergyConsumptionTracker implements CostCalculator {  
    private double ratePerUnit;  
    private double[] dailyConsumption;  
    private double[] dailyCosts;  
    private double totalCost;  
  
    @Override  
    public void getEnergyDetails(double rate, double[] consumption) {  
        this.ratePerUnit = rate;  
        this.dailyConsumption = consumption;  
        this.dailyCosts = new double[consumption.length];  
    }  
  
    @Override  
    public void calculateCost() {  
        totalCost = 0.0;  
        for (int i = 0; i < dailyConsumption.length; i++) {  
            dailyCosts[i] = dailyConsumption[i] * ratePerUnit;  
            totalCost += dailyCosts[i];  
        }  
  
        System.out.println("Day-wise Energy Cost:");  
        for (int i = 0; i < dailyCosts.length; i++) {  
            System.out.printf("Day %d: Rs. %.2f%n", (i + 1), dailyCosts[i]);  
        }  
        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 :** Wrong

**Marks : 0/10**

# Rajalakshmi Engineering College

Name: Gnaana Kumaar

Email: 241001062@rajalakshmi.edu.in

Roll no: 241001062

Phone: 9790249960

Branch: REC

Department: IT - Section 4

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 : 2

Total Mark : 15

Marks Obtained : 9

#### Section 1 : MCQ

1. 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**

2. 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**

3. 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**

4. 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**

5. 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**

6. 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**

7. 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**

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

**Answer**

```
MyInterface.staticMethod();
```

**Status : Correct**

**Marks : 1/1**

9. 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**

10. 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**

11. 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**

12. 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**

Compilation Error

**Status : Wrong**

**Marks : 0/1**

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

**Answer**

The class cannot implement two interfaces with identical default methods.

**Status : Wrong**

**Marks : 0/1**

14. 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

15. 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 A

**Status :** Wrong

**Marks : 0/1**