

# Rajalakshmi Engineering College

Name: Aldrine Linjoe.s

Email: 240701033@rajalakshmi.edu.in

Roll no: 240701033

Phone: 7092049029

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\_CY**

Attempt : 1

Total Mark : 40

Marks Obtained : 40

### **Section 1 : Coding**

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

Ray is developing a tax calculation program in Java. The program includes an interface named TaxCalculator with a method to calculate tax based on salary. The SimpleTaxCalculator class implements this interface and determines the tax to be paid based on the salary amount using progressive tax slabs.

Your task is to implement this system. The program first takes an integer T representing the number of test cases, followed by T salary values. For each salary, calculate the total tax to be paid based on the following progressive tax rules:

For the first 50,000 of salary, the tax rate is 5%. For the next 50,000 (i.e., from 50,001 to 1,00,000), the tax rate is 10%. For any amount above 1,00,000, the tax rate is 20%. (That is, only the amount above 1,00,000 is

taxed at 20%).

Example

Input

3

78000

110000

23000

Output

5300

9500

1150

Explanation

For Salary Rs. 78,000

$$\text{Tax} = 0.1 * (78,000 - 50,000) + 0.05 * 50,000 = 5,300$$

For Salary Rs. 1,10,000

$$\text{Tax} = 0.2 * (110000 - 100000) + 0.1 * 50,000 + 0.05 * 50,000 = 9,500$$

For Salary Rs. 23,000

$$\text{Tax} = 0.05 * 23,000 = 1,150$$

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

The first line of the input consists of an integer, T, representing the number of test cases.

The next T lines of the input consist of a single integer, representing the annual salary of an individual, separated by a line.

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

The output displays the calculated tax as an integer for each test case, separated by a line.

Refer to the sample output for the formatting specifications.

### **Sample Test Case**

Input: 2

100

300

Output: 5

15

### **Answer**

```
import java.util.Scanner;
// You are using Java
interface TaxCalculator{
    public int calculateTax(int n);
}
class SimpleTaxCalculator implements TaxCalculator{
    public int calculateTax(int n){
        double t;
        if(n<=50000){
            t=0.05*n;
            int d=(int)t;
            return d;
        }else if(n>50000 && n<=100000){
            t=0.1*(n-50000)+0.05*50000;
            int d=(int)t;
            return d;
        }else if(n>100000){
            t=0.2*(n-100000)+0.1*50000+0.05*50000;
            int d=(int)t;
            return d;
        }
        return 1;
    }
    class Main {
        public static void main(String[] args) {
            Scanner scanner = new Scanner(System.in);
```

```
int T = scanner.nextInt();
TaxCalculator taxCalculator = new SimpleTaxCalculator();

for (int i = 0; i < T; i++) {
    int salary = scanner.nextInt();
    int tax = taxCalculator.calculateTax(salary);
    System.out.println(tax);
}

scanner.close();
}
```

Status : Correct

Marks : 10/10

## 2. 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;  
  
// You are using Java  
//Type your code here  
interface Principal{  
    public double getPrincipal();  
}  
interface InterestRate{  
    public double getInterestRate();  
}  
class Loan implements Principal,InterestRate{  
    double p,r;  
    Loan(double p,double r){  
        this.p=p;  
        this.r=r;  
    }  
    public double getPrincipal(){  
        return p;  
    }  
    public double getInterestRate(){  
        return r;  
    }  
    public double calculateTotalInterest(int n){
```

```

        return p*r*n;
    }

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

### 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;  
  
// You are using Java  
interface SpeedConverter{  
    public double convertSpeed(double s);  
}  
class MPHtoKPHConverter implements SpeedConverter{  
    public double convertSpeed(double s){  
        return s*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

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;  
// You are using Java  
interface PriceCalculator{  
    public double calculatePrice(double p,double d);  
}  
class DiscountCalculator implements PriceCalculator{  
    public double calculatePrice(double p,double d){  
        return p-p*(d/100);  
    }  
}  
  
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**