Rajalakshmi Engineering College

Name: UMAR FAROOK
Email: 240701573@rajalakshmi.edu.in
Roll no: 240701573
Phone: 9791730398
Branch: REC
Department: CSE - Section 5
Batch: 2028
Degree: B.E - CSE

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

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

```java
import java.util.Scanner;

interface PriceCalculator{
    double calculatePrice(double originalPrice,double discount);
}

class DiscountCalculator implements PriceCalculator{
    public double calculatePrice(double originalPrice,double discount){
        return originalPrice - (originalPrice * (discount / 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
    }
}
```

2. 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

```java
import java.util.Scanner;

interface StepCounter{

    void countSteps(int steps);
    int getTotalSteps();
}
interface CalorieCalculator{
    double calculateCaloriesBurned(int totalSteps);
}

class FitnessTracker implements StepCounter, CalorieCalculator{
    int s = 0;
    public void countSteps(int steps){
     s += steps;
    }

    public int getTotalSteps(){
        return s;
    }

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

}
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*

3.   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

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

For Salary Rs. 1,10,000

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

For Salary Rs. 23,000

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

```java
import java.util.Scanner;

interface TaxCalculator{
    int calculateTax(int salary);
}

class SimpleTaxCalculator implements TaxCalculator{
    public int calculateTax(int salary){
        int tax = 0;
        if(salary <= 50000){
            tax = (int)(0.05 * salary);
        }
        else if(salary > 50000 && salary <= 100000){
            tax = (int)(0.1 * (salary - 50000)+ 0.05 * 50000);
        }
        else if(salary > 100000){
            tax = (int)(0.2 * (salary - 100000) + 0.1 * 50000 + 0.05*50000);
        }
        return tax;
    }
}
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();
```

```
    }
}
```

4.   Problem Statement:

Sam is developing a geometry application and needs a class for trapezoid calculations. Create a "Trapezoid" class implementing a "ShapeInput" interface with a method to input trapezoid dimensions.

Also, implement a "ShapeCalculator" interface with methods to compute area and perimeter. In the "Main" class, instantiate Trapezoid, gather user input, and display the calculated area and perimeter with two decimal places.

Note

Area of Trapezoid = (1/2) * (base1 + base2) * height

Perimeter of Trapezoid = base1 + base2 + side1 + side2

Input Format

The first line of input is a double-point value representing base1 of the trapezoid.

The second line of input is a double-point value representing base2 of the trapezoid.

The third line of input is a double-point value representing the height of the trapezoid.

The fourth line of input is a double-point value representing side1 of the trapezoid.

The fifth line of input is a double-point value representing side2 of the trapezoid.

Output Format

The output displays the two lines of the calculated area (double type) and perimeter (double type) of the trapezoid, each rounded to two decimal places in the following format:

"Area of the Trapezoid: <<calculated area>>".

Perimeter of the Trapezoid: <<calculated perimeter>>".

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 1.0
2.0
1.0
3.0
1.0
Output: Area of the Trapezoid: 1.50
Perimeter of the Trapezoid: 7.00

Answer

```java
import java.util.Scanner;

interface ShapeInput{
    void getInput();
}
interface ShapeCalculator{
    double calculateArea();
    double calculatePerimeter();
}

class Trapezoid implements ShapeInput, ShapeCalculator{
    double b1,b2,h,s1,s2;
    public void getInput(){
        Scanner sc = new Scanner(System.in);
        b1 = sc.nextDouble();
        b2 = sc.nextDouble();
        h = sc.nextDouble();
        s1 = sc.nextDouble();
        s2 = sc.nextDouble();
```

```java
    }

    public double calculateArea(){
        double area = (0.5) * (b1 + b2) * h;
        return area;
    }

    public double calculatePerimeter(){
        double perimeter = b1+b2+s1+s2;
        return perimeter;
    }

}
public class Main {
    public static void main(String[] args) {
        Trapezoid trapezoid = new Trapezoid();
        trapezoid.getInput();

        double area = trapezoid.calculateArea();
        double perimeter = trapezoid.calculatePerimeter();

        System.out.println("Area of the Trapezoid: " + String.format("%.2f", area));
        System.out.println("Perimeter of the Trapezoid: " + String.format("%.2f",
perimeter));
    }
}
```

Status : Correct   *Marks : 10/10*