

Rajalakshmi Engineering College

Name: SONASREE RP

Email: 240701521@rajalakshmi.edu.in

Roll no: 240701521

Phone: 7305340666

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 6_CY

Attempt : 1

Total Mark : 40

Marks Obtained : 40

Section 1 : Coding

1. Problem Statement

Teena's retail store has implemented a Loyalty Points System to reward customers based on their spending. The program calculates and displays the loyalty points based on whether the customer is a regular or a premium customer.

For regular customers (class Customer), the loyalty points are calculated as:

Loyalty points = amount spent / 10

For premium customers (class PremiumCustomer, which inherits from Customer), the loyalty points are calculated as:

Loyalty points = 2 * (amount spent / 10)

The program should use method overriding for premium customers to

calculate their loyalty points. The method that needs to be overridden is calculateLoyaltyPoints in the Customer class.

Input Format

The first line of input consists of an integer representing the amount spent by the customer.

The second line consists of a string representing the premium customer status:

- "yes" if the customer is a premium customer.
- "no" if the customer is not a premium customer.

Output Format

The output should display the loyalty points earned based on the amount spent and the customer type.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 50

yes

Output: 10

Answer

```
import java.util.Scanner;

class Customer {
    public int calculateLoyaltyPoints(int amountSpent) {
        return amountSpent / 10;
    }
}

class PremiumCustomer extends Customer {
    public int calculateLoyaltyPoints(int amountSpent) {
        return 2 * (amountSpent / 10);
    }
}
```

```
public class Main {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
  
        int amountSpent = scanner.nextInt();  
  
        String isPremium = scanner.next().toLowerCase();  
  
        Customer customer;  
  
        if (isPremium.equals("yes")) {  
            customer = new PremiumCustomer();  
        } else {  
            customer = new Customer();  
        }  
  
        int loyaltyPoints = customer.calculateLoyaltyPoints(amountSpent);  
  
        System.out.println(loyaltyPoints);  
    }  
}
```

Status : Correct

Marks : 10/10

2. Problem Statement

Bob has been tasked with creating a program using CircleUtils class to calculate and display the circumference and area of the circle.

The program should allow Bob to input the radius of a circle as both an integer and a double and compute both the circumference and area of the circle using separate overloaded methods:

calculateCircumference- To calculate the circumference using the formula
 $2 * 3.14 * \text{radius}$
calculateArea- To calculate the area $3.14 * \text{radius} * \text{radius}$

Write a program to help Bob.

Input Format

The first line of input consists of an integer m, representing the radius of the circle as a whole number.

The second line consists of a double value n , representing the radius of the circle as a decimal number.

Output Format

The first line of output displays two space-separated double values, rounded to two decimal places, representing the circumference of the circle with the integer radius and the double radius, respectively.

The second line displays two space-separated double values, rounded to two decimal places, representing the area of the circle with the integer radius and the double radius, respectively.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 5

3.50

Output: 31.40 21.98

78.50 38.47

Answer

```
import java.util.Scanner;

class CircleUtils {

    public double calculateCircumference(int radius) {
        return 2 * 3.14 * radius;
    }

    public double calculateCircumference(double radius) {
        return 2 * 3.14 * radius;
    }

    public double calculateArea(int radius) {
        return 3.14 * radius * radius;
    }
}
```

```

    }

    public double calculateArea(double radius) {
        return 3.14 * radius * radius;
    }
}

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

        int radiusInt = scanner.nextInt();
        double radiusDouble = scanner.nextDouble();

        CircleUtils circleUtils = new CircleUtils();

        double circumferenceInt = circleUtils.calculateCircumference(radiusInt);
        double circumferenceDouble =
        circleUtils.calculateCircumference(radiusDouble);
        double areaInt = circleUtils.calculateArea(radiusInt);
        double areaDouble = circleUtils.calculateArea(radiusDouble);

        System.out.format("%.2f %.2f\n", circumferenceInt, circumferenceDouble);
        System.out.format("%.2f %.2f", areaInt, areaDouble);

        scanner.close();
    }
}

```

Status : Correct

Marks : 10/10

3. Problem Statement

A painter needs to determine the cost to paint different shapes based on their surface area. The program should be designed to handle the area of a sphere and calculate the total painting cost using the following formulas:

Area of sphere: $\text{Area} = 4 * \pi * r^2$ where $\pi = 3.14$
 Total painting cost: Cost = cost per square meter * area of sphere

The program will consist of three classes:

Shape class: This class should set the shape type and radius.Area class: This class should extend Shape to calculate the area.Cost class: This class should extend Area to calculate the total painting cost.

Input Format

The input consists of a string representing the shape type, a double value representing the radius, and another double value representing the cost per square meter on each line.

Output Format

For a valid shape type of "Sphere":

- The first line prints: "Area of Sphere is: <calculated_area>" rounded to two decimal places.
- The second line prints: "Cost to paint the shape is: <total_painting_cost>" rounded to two decimal places.

For any other shape types, print: "Invalid type".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: Sphere

3.4

5.8

Output: Area of Sphere is: 145.19

Cost to paint the shape is: 842.12

Answer

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

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

```
// Base class
```

```
class Shape {
```

```
protected String shapeType;
```

```
protected double radius;
```

```
public void setShape(String shapeType, Scanner scanner) {
    this.shapeType = shapeType;
    if (shapeType.equals("Sphere")) {
        this.radius = scanner.nextDouble();
    }
}

// Intermediate class extending Shape
class Area extends Shape {
    protected double area;

    public void calculateArea() {
        if (shapeType.equals("Sphere")) {
            this.area = 4 * 3.14 * radius * radius;
        }
    }

    public double getArea() {
        return area;
    }
}

// Derived class extending Area
class Cost extends Area {
    private double costPerSquare;
    private double totalCost;

    public void setCost(double costPerSquare) {
        this.costPerSquare = costPerSquare;
    }

    public void calculateCost() {
        if (shapeType.equals("Sphere")) {
            this.totalCost = area * costPerSquare;

            System.out.printf("Area of Sphere is: %.2f\n", area);
            System.out.printf("Cost to paint the shape is: %.2f\n", totalCost);
        } else {
            System.out.println("Invalid type");
        }
    }
}
```

```
        }  
    }  
  
public class Main {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        String s = scanner.next();  
        Cost shape = new Cost();  
        shape.setShape(s, scanner);  
        double costToPaint = scanner.nextDouble();  
        shape.calculateArea();  
        shape.setCost(costToPaint);  
        shape.calculateCost();  
    }  
}
```

Status : Correct

Marks : 10/10

4. Problem Statement

Mary is managing a business and wants to analyze its profitability. She operates both a regular business model and a seasonal business model. To assess profitability, she uses a program that calculates and compares the profit margins for both models based on revenue and cost.

The program defines:

BusinessUtility class with a method calculateMargin(double revenue, double cost).SeasonalBusinessUtility (inherits from BusinessUtility) and overrides calculateMargin(double revenue, double cost), adding a seasonal adjustment of 10% to the base margin.ProtectionChecker class with a method checkProfitability(double regularMargin), which prints "Business is profitable." if the regular margin is 10% or more, otherwise prints "Business is not profitable.".

Mary inputs revenue and cost, and the program compute and display the regular and seasonal margins using:

$$\text{Margin} = ((\text{Revenue} - \text{Cost}) / \text{Revenue}) \times 100$$

Seasonal Margin = Margin + 10

Input Format

The first line of input consists of a double value r , representing the revenue.

The second line consists of a double value c , representing the cost.

Output Format

The first line prints a double value, representing the regular profit margin, rounded to two decimal places, in the format: "Regular Margin: X. XX%", where X.XX denotes the calculated regular margin.

The second line prints a double value, representing the seasonal profit margin, rounded to two decimal places, in the format: "Seasonal Margin: X. XX%", where X.XX denotes the calculated seasonal margin.

The third line prints a string, indicating whether the business is profitable or not profitable, based on the regular margin.

If the regular margin is less than 10, print "Business is not profitable.". If it is 10 or greater, print "Business is profitable."

Refer to the sample output for the formatting specifications.

Sample Test Case

Input: 1000.0

800.0

Output: Regular Margin: 20.00%

Seasonal Margin: 30.00%

Business is profitable.

Answer

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

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

```
class BusinessUtility {  
    public double calculateMargin(double revenue, double cost) {  
        return ((revenue - cost) / revenue) * 100;  
    }  
}  
  
class SeasonalBusinessUtility extends BusinessUtility {  
    @Override  
    public double calculateMargin(double revenue, double cost) {  
        double baseMargin = super.calculateMargin(revenue, cost);  
        return baseMargin + 10;  
    }  
}  
  
class ProfitabilityChecker {  
    public void checkProfitability(double regularMargin) {  
        if (regularMargin >= 10) {  
            System.out.println("Business is profitable.");  
        } else {  
            System.out.println("Business is not profitable.");  
        }  
    }  
}  
  
class Main {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        double revenue = scanner.nextDouble();  
        double cost = scanner.nextDouble();  
        BusinessUtility business = new BusinessUtility();  
        SeasonalBusinessUtility seasonalBusiness = new  
        SeasonalBusinessUtility();  
        double regularMargin = business.calculateMargin(revenue, cost);  
        double seasonalMargin = seasonalBusiness.calculateMargin(revenue,  
cost);  
  
        System.out.printf("Regular Margin: %.2f%%\n", regularMargin);  
        System.out.printf("Seasonal Margin: %.2f%%\n", seasonalMargin);  
  
        ProfitabilityChecker checker = new ProfitabilityChecker();  
        checker.checkProfitability(regularMargin);  
        scanner.close();  
    }  
}
```

}

Status : Correct

Marks : 10/10