Rajalakshmi Engineering College

Name: RANNESH KHUMAR B R
Email: 240701422@rajalakshmi.edu.in
Roll no: 2116240701
Phone: 9042350670
Branch: REC
Department: CSE - Section 8
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

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.ProfitabilityChecker 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:

Margin = ((Revenue − Cost) / Revenue) × 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;

```java
class BusinessUtility {
    public double calculateMargin(double revenue, double cost) {
        double margin = ((revenue - cost) / revenue) * 100;
        return margin;
    }
}

class SeasonalBusinessUtility extends BusinessUtility {
    public double calculateMargin(double revenue, double cost) {
        double baseMargin = super.calculateMargin(revenue, cost);
        double seasonalMargin = baseMargin + 10;
        return seasonalMargin;
    }
}

class ProfitabilityChecker {
    public void checkProfitability(double regularMargin) {
        if (regularMargin < 10) {
            System.out.println("Business is not profitable.");
        } else {
            System.out.println("Business is 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();
```

```
    }
}
```

2. 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: Area = 4 * pi * r² where pi = 3.14Total 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

```java
import java.util.Scanner;



class Shape {
    String shapeName = "";
    double radius;
    void setShape(String x, Scanner scanner) {
        shapeName = x;
        if (x.equals("Sphere")) {
            radius = scanner.nextDouble();
        } else {
            System.out.println("Invalid type");
            return;
        }
    }
}

class Area extends Shape {
    double area;
    void calculateArea() {
        if (shapeName.equals("Sphere")) {
            area = 4 * 3.14 * radius * radius;
            System.out.printf("Area of Sphere is: %.2f%n", area);
        }
    }
}

class Cost extends Area {
    double costPerMtrSquare;

    void setCost(double x) {
        costPerMtrSquare = x;
    }
    void calculateCost() {
```

```
        double totalCost = costPerMtrSquare * area;
        if (area > 0) {
            System.out.printf("Cost to paint the shape is: %.2f%n", totalCost);
        }
    }
}
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*

3.   Problem Statement

Adams has a reputation company with a great number of employees. He must calculate the salary weekly according to the hourly rate and working hours. Create a program to define a class Employee with attributes name and hourly rate. Create a subclass HourlyEmployee that calculates the weekly salary based on the number of hours worked.

(The first 40 hours are based on the regular hour rate. If the work hours are greater than 40 then the work wage is 1.5 times the hourly rate)

Note: Use Math(Math.max, Math.min) functions .

Example

Input:

Chris

10

Output:

Weekly Salary: Rs.475.00

Explanation:

Calculation:

The first 40 hours are paid normally: 40 × 10 = 400.00The extra 5 hours are paid at 1.5 times the hourly rate: 5 × (10×1.5) = 5 × 15 = 75.00Total salary: 400.00 + 75.00 = 475.00

Input Format

The first line of input consists of a string that represents the name of the employee.

The second line consists of a double value that represents the rate for an hour.

The last line consists of an integer that represents the total hours worked.

Output Format

The output displays the total salary of the employee, where salary is rounded to two decimal places in the format: "Weekly Salary: Rs.<double value>".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: Dave
10.0
40
Output: Weekly Salary: Rs.400.00

Answer

```
import java.util.Scanner;
import java.text.DecimalFormat;
```

```java
class Employee {
    private String name;
    private double hourlyRate;

    public Employee(String name, double hourlyRate) {
        this.name = name;
        this.hourlyRate = hourlyRate;
    }

    public String getName() {
        return name;
    }

    public double getHourlyRate() {
        return hourlyRate;
    }
}

class HourlyEmployee extends Employee {
    private int hoursWorked;

    public HourlyEmployee(String name, double hourlyRate, int hoursWorked) {
        super(name, hourlyRate);
        this.hoursWorked = hoursWorked;
    }

    public int getHoursWorked() {
        return hoursWorked;
    }

    public double calculateWeeklySalary() {
        int regularHours = Math.min(hoursWorked, 40);
        int overtimeHours = Math.max(0, hoursWorked - 40);
        double regularPay = regularHours * getHourlyRate();
        double overtimePay = overtimeHours * (getHourlyRate() * 1.5);
        return regularPay + overtimePay;
    }
}

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

```
        String name = scanner.nextLine();
        double hourlyRate = scanner.nextDouble();
        int hoursWorked = scanner.nextInt();

        HourlyEmployee employee = new HourlyEmployee(name, hourlyRate,
hoursWorked);

        double weeklySalary = employee.calculateWeeklySalary();
        DecimalFormat df = new DecimalFormat("#.00");
        String formattedSalary = df.format(weeklySalary);
        System.out.println("Weekly Salary: Rs." + formattedSalary);
        scanner.close();
    }
}
```

Status : Correct   *Marks : 10/10*

4. 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 * radiuscalculateArea- To calculate the area 3.14 * radius * 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

```java
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);
```

```java
        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*