Rajalakshmi Engineering College

Name: Shreenidhi T
Email: 240701503@rajalakshmi.edu.in
Roll no: 240701503
Phone: 9150942326
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

Arun wants to calculate the age gap between the grandfather and the son and determine the father's age after 5 years.

Your task is to assist him in developing a program using three classes: GrandFather, Father, and Son, where the GrandFather stores the grandfather's age, the Father extends GrandFather to include the father's age and calculates his age after 5 years, and Son extends Father to include the son's age and calculate the age difference between the grandfather and the son.

Input Format

The input consists of three integers representing the ages of the grandfather, father, and son, one per line.

Output Format

The first line of output prints "Grandfather and son's age gap:" followed by an integer representing the age gap between the grandfather and the son, ending with "years".

The second line prints "Father's Age:" followed by an integer representing the father's age after 5 years, ending with "years".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 50
30
3

Output: Grandfather and son's age gap: 47 years
Father's Age: 35 years

Answer

```
import java.util.Scanner;

// You are using Java
import java.util.Scanner;

class GrandFather {
   int grandfatherAge;
   public void setGrandfatherAge(int age) {
      grandfatherAge = age;
   }
   public int getGrandfatherAge() {
      return grandfatherAge;
   }
}

class Father extends GrandFather {
   int fatherAge;
   public void setFatherAge(int age) {
      fatherAge = age;
   }
   public int getFatherAge() {
      return fatherAge;
   }
   public int calculateFatherAgeAfter5Years() {
      return fatherAge + 5;
   }
}

class Son extends Father {
   int sonAge;
   public void setSonAge(int age) {
      sonAge = age;
   }
   public int getSonAge() {
      return sonAge;
   }
   public int calculateGrandfatherSonAgeDifference() {
      return grandfatherAge - sonAge;
   }
}


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

      int grandfatherAge = scanner.nextInt();
      son.setGrandfatherAge(grandfatherAge);

      int fatherAge = scanner.nextInt();
      son.setFatherAge(fatherAge);

      int sonAge = scanner.nextInt();
      son.setSonAge(sonAge);

      System.out.println("Grandfather and son's age gap: "+
son.calculateGrandfatherSonAgeDifference() + " years");

      int fatherAgeAfter5Years = son.calculateFatherAgeAfter5Years();
      System.out.println("Father's Age: " + fatherAgeAfter5Years + " years");
   }
}
```

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 * 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

```
import java.util.Scanner;

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

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

45

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;

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

class Employee {
   protected String name;
   protected double hourlyRate;

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

class HourlyEmployee extends Employee {
   private int hoursWorked;

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

```
    public double calculateWeeklySalary() {
        int regularHours = Math.min(40, hoursWorked);
        int overtimeHours = Math.max(0, hoursWorked - 40);
        double regularPay = regularHours * hourlyRate;
        double overtimePay = overtimeHours * hourlyRate * 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

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 * r^2 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

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

```
import java.util.Scanner;
import java.lang.Math;
import java.text.DecimalFormat;
class Shape {
   protected String shapeType;
   protected double radius;
   public void setShape(String type, Scanner scanner) {
      this.shapeType = type;
```

```
        if (scanner.hasNextDouble()) {
            this.radius = scanner.nextDouble();
        } else {
            scanner.next();
            this.radius = 0.0;
        }
    }
    public String getShapeType() {
        return shapeType;
    }
}
class Area extends Shape {
    protected double area;
    private static final double PI = 3.14;
    public void calculateArea() {
        if ("Sphere".equalsIgnoreCase(shapeType)) {
            this.area = 4 * PI * radius * radius;
        } else {
            this.area = 0.0;
        }
    }
    public double getArea() {
        return area;
    }
}
class Cost extends Area {
    private double costPerSquareMeter;
    private double totalPaintingCost;
    public void setCost(double costPerSquareMeter) {
        this.costPerSquareMeter = costPerSquareMeter;
    }
    public void calculateCost() {
        DecimalFormat df = new DecimalFormat("0.00");

        if ("Sphere".equalsIgnoreCase(shapeType)) {
            this.totalPaintingCost = costPerSquareMeter * area;
            System.out.println("Area of Sphere is: " + df.format(area));
            System.out.println("Cost to paint the shape is: " +
df.format(totalPaintingCost));

        } 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***