Rajalakshmi Engineering College

Name: Adithya B
Email: 240701018@rajalakshmi.edu.in
Roll no: 240701018
Phone: 9444117405
Branch: REC
Department: CSE - Section 10
Batch: 2028
Degree: B.E - CSE

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

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;

```java
import java.text.DecimalFormat;

// You are using Java
import java.util.Scanner;
import java.text.DecimalFormat;
import java.lang.Math;

class Employee {
    protected String name;
    protected 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;
    private static final int REGULAR_HOURS_LIMIT = 40;
    private static final double OVERTIME_RATE_MULTIPLIER = 1.5;

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

    public double calculateWeeklySalary() {
        double regularHours = Math.min(hoursWorked, REGULAR_HOURS_LIMIT);
        double overtimeHours = Math.max(0, hoursWorked -
REGULAR_HOURS_LIMIT);

        double regularPay = regularHours * hourlyRate;
        double overtimePay = overtimeHours * (hourlyRate *
OVERTIME_RATE_MULTIPLIER);
```

```java
        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*

2. 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

```java
import java.util.Scanner;

import java.util.Scanner;


class Customer {

    public int calculateLoyaltyPoints(int amountSpent) {

        return amountSpent / 10;
    }
}
```

```java
class PremiumCustomer extends Customer {

    @Override
    public int calculateLoyaltyPoints(int amountSpent) {

        return 2 * super.calculateLoyaltyPoints(amountSpent);

    }
}

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*

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: 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

import java.util.Scanner;

// You are using Java
import java.util.Scanner;
class Shape {

```java
    protected String shapeType;
    protected double radius;

    // main calls: setShape(s, scanner);
    public void setShape(String type, Scanner sc) {
        this.shapeType = type;

        if (type.equals("Sphere")) {
            this.radius = sc.nextDouble();   // read radius from scanner
        } else {
            this.radius = -1; // invalid shape
        }
    }
}
class Area extends Shape {
    protected double area;

    public void calculateArea() {
        if (!shapeType.equals("Sphere")) {
            area = -1;
            return;
        }

        area = 4 * 3.14 * radius * radius;
    }
}
class Cost extends Area {
    private double costPerSqMeter;
    private double totalCost;

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

    public void calculateCost() {
        if (area == -1) {
            System.out.println("Invalid type");
        } else {
            totalCost = area * costPerSqMeter;
            System.out.printf("Area of Sphere is: %.2f ", area);
            System.out.printf("Cost to paint the shape is: %.2f ", 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*

4. 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

```java
import java.util.Scanner;

// You are using Java
class GrandFather {
    protected int grandfatherAge;

    public void setGrandfatherAge(int age) {
        this.grandfatherAge = age;
    }
}
class Father extends GrandFather {
    protected int fatherAge;

    public void setFatherAge(int age) {
        this.fatherAge = age;
    }

    public int calculateFatherAgeAfter5Years() {
        return fatherAge + 5;
    }
}
class Son extends Father {
    private int sonAge;

    public void setSonAge(int age) {
        this.sonAge = age;
    }
```

```java
    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*