

Rajalakshmi Engineering College

Name: Monica B
Email: 240701330@rajalakshmi.edu.in
Roll no: 240701330
Phone: 6385195950
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

Teena is launching a new airline, Boeing747, and needs to calculate the total revenue generated from ticket sales based on the ticket cost and seat availability. Teena's airline offers two types of seats: regular and premium. The ticket cost and seat availability for both types of seats need to be considered for revenue calculation.

To help with this, Teena wants to implement a system using multilevel inheritance with three classes:

Airline: This class will have the ticket cost as an attribute and defines the method setCost(double cost) and double getCost(). Indigo: This class will extend Airline and add the seat availability attribute and defines the method getSeatAvailability() and setSeatAvailability(int seatAvailability). Boeing747: This class will extend Indigo and include a

method `calculateTotalRevenue()` based on the ticket cost and seat availability .

Teena needs to calculate the total revenue using the formula:

$\text{Total Revenue} = \text{ticket cost} * \text{seat availability}$

Help Teena implement this system for calculating the revenue of her airline.

Input Format

The first line of input consists of a double value, representing the flight's ticket cost.

The second line consists of an integer, representing seat availability.

Output Format

The first line of output prints "Ticket Cost: Rs. " followed by a double value representing the ticket cost rounded to one decimal place.

The second line of output prints "Seat Availability: X seats" where X is an integer value representing the seat availability.

The third line of output prints "Total Revenue: Rs. " followed by a double value representing the total revenue rounded to one decimal place.

Refer to the sample output for the exact text and format.

Sample Test Case

Input: 1000.0
100

Output: Ticket Cost: Rs. 1000.0
Seat Availability: 100 seats
Total Revenue: Rs. 100000.0

Answer

```
import java.util.Scanner;  
  
class Airline {  
    private double ticketCost;
```

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

    public double getCost() {
        return ticketCost;
    }
}

class Indigo extends Airline {
    private int seatAvailability;

    public void setSeatAvailability(int seatAvailability) {
        this.seatAvailability = seatAvailability;
    }

    public int getSeatAvailability() {
        return seatAvailability;
    }
}

class Boeing747 extends Indigo {
    public double calculateTotalRevenue() {
        return getCost() * getSeatAvailability();
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Boeing747 plane = new Boeing747();

        double ticketCost = scanner.nextDouble();
        plane.setCost(ticketCost);
        int seatAvailability = scanner.nextInt();
        plane.setSeatAvailability(seatAvailability);

        System.out.printf("Ticket Cost: Rs. %.1f\n", plane.getCost());
        System.out.println("Seat Availability: " + plane.getSeatAvailability() + "
seats");
        System.out.printf("Total Revenue: Rs. %.1f\n",
plane.calculateTotalRevenue());
    }
}
```

}

Status : Correct

Marks : 10/10

2. 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: $40 \times 10 = 400.00$ The extra 5 hours are paid at 1.5 times the hourly rate: $5 \times (10 \times 1.5) = 5 \times 15 = 75.00$ Total 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;
```

```
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 = (int) Math.min(hoursWorked, 40);
        int overtimeHours = (int) 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

3. 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 = $2 * (\text{amount spent} / 10)$

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 {  
    protected int amountSpent;
```

```
    public Customer() {  
        this.amountSpent = 0;  
    }
```

```
    public void setAmountSpent(int amountSpent) {
```

```

        this.amountSpent = amountSpent;
    }

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

class PremiumCustomer extends Customer {

    public PremiumCustomer() {
        super();
    }

    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

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

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;

class Shape {
    String type;
    double radius;
    void setShape(String type, Scanner sc) {
        this.type = type;
        this.radius = sc.nextDouble();
    }
}

class Area extends Shape {
    double area;
    void calculateArea() {
        if (type.equals("Sphere")) {
            area = 4 * 3.14 * radius * radius;
        }
    }
}

class Cost extends Area {
    double costPerSqMeter;
    double totalCost;
    void setCost(double cost) {
        costPerSqMeter = cost;
    }
    void calculateCost() {
        if (type.equals("Sphere")) {
            totalCost = costPerSqMeter * area;
            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