

Rajalakshmi Engineering College

Name: SANTHOSH S

Email: 241801251@rajalakshmi.edu.in

Roll no: 241801251

Phone: 9790911586

Branch: REC

Department: AI & DS - Section 3

Batch: 2028

Degree: B.E - AI & DS

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

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;
// You are using Java
class Employee{
    String name;
    double rate;
}
class HourlyEmployee extends Employee{
    int hours;
    double salary=0;
    HourlyEmployee(String name,double rate,int hours)
    {
        this.name=name;
        this.rate=rate;
        this.hours=hours;
    }
    public double calculateWeeklySalary()
    {
        if(hours>40)
        {
            salary+=40*rate;
            hours-=40;
            salary+=hours*rate*1.5;
        }
        else
            salary=hours*rate;
        return salary;
    }
}
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 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:

Total Revenue = ticket cost * 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;  
  
// You are using Java  
class Airline{  
    double cost;  
    void setCost(double cost){  
        this.cost=cost;  
    }  
    double getCost()  
    {  
        return cost;  
    }  
}  
class Indigo extends Airline{  
    int seats;  
    int getSeatAvailability()  
    {  
        return seats;  
    }  
    void setSeatAvailability(int Availability)  
    {
```

```

        seats=Availability;
    }
}

class Boeing747 extends Indigo{

    double calculateTotalRevenue()
    {
        return seats*cost;
    }
}

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

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 = $4 * \pi * r^2$ 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;  
  
class Shape{  
    String Shape;  
    double radius;  
    void setShape(String s,Scanner ss)  
    {  
        this.Shape=s;  
    }
```

```

        radius=ss.nextDouble();
    }
}
class Area extends Shape{
    public double calculateArea(){
        return 4*radius*radius*3.14;
    }
}
class Cost extends Area{
    double cost;
    void setCost(double cost)
    {
        this.cost=cost;
    }
    void calculateCost()
    {
        if(Shape.equals("Sphere"))
        {
            System.out.printf("Area of Sphere is: %.2f\n",calculateArea());
            System.out.printf("Cost to paint the shape is: %.2f
\n",calculateArea()*cost);
        }
        else
            System.out.printf("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

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  
class GrandFather{
```

```
int Gage;
void setGrandfatherAge(int age)
{
    this.Gage=age;
}
}
class Father extends GrandFather{
    int Fage;
    void setFatherAge(int age)
    {
        this.Fage=age;
    }
    int calculateFatherAgeAfter5Years()
    {
        return Fage+5;
    }
}
class Son extends Father{
    int Sage;
    void setSonAge(int age){
        this.Sage=age;
    }
    int calculateGrandfatherSonAgeDifference()
    {
        return Gage-Sage;
    }
}
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