Rajalakshmi Engineering College

Name: Nandhini Velmurugan
Email: 241901063@rajalakshmi.edu.in
Roll no: 241901063
Phone: 9043367699
Branch: REC
Department: CSE (CS) - Section 1
Batch: 2028
Degree: B.E - CSE (CS)

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;
```

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

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

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

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

```
import java.util.Scanner;

class Customer {
   public Customer() {
   }

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

class PremiumCustomer extends Customer {
   public PremiumCustomer() {
   }

   @Override
   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***

3. 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;

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

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

class Son extends Father {
   private int sonAge;
   public void setSonAge(int age) {
      this.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_**

4. 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;

class Airline {
   private double cost;

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

   public double getCost() {
      return cost;
   }
}
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***