

Rajalakshmi Engineering College

Name: samyuktha ss

Email: 241001217@rajalakshmi.edu.in

Roll no:

Phone: 7539908242

Branch: REC

Department: IT - Section 2

Batch: 2028

Degree: B.E - IT

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:

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 this.cost;
}
}

class Indigo extends Airline {
    private int seatAvailability;

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

    public int getSeatAvailability() {
        return this.seatAvailability;
    }
}

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

    public void displayRevenueDetails() {
        System.out.printf("Ticket Cost: Rs. %.1f ", getCost());
        System.out.printf("Seat Availability: %d seats ", getSeatAvailability());
        System.out.printf("Total Revenue: Rs. %.1f", calculateTotalRevenue());
    }
}

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 {
    String name;
    double hourlyRate;

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

class HourlyEmployee extends Employee {
    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

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:

$$\text{Loyalty points} = \text{amount spent} / 10$$

For premium customers (class PremiumCustomer, which inherits from Customer), the loyalty points are calculated as:

$$\text{Loyalty points} = 2 * (\text{amount spent} / 10)$$

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 {
    public Customer() {

    }

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

class PremiumCustomer extends Customer {
    public PremiumCustomer() {
        super();
    }

    @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

4. 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:

$$\text{Margin} = ((\text{Revenue} - \text{Cost}) / \text{Revenue}) \times 100$$

$$\text{Seasonal Margin} = \text{Margin} + 10$$

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) {
        double margin = ((revenue - cost) / revenue) * 100;
        return margin;
    }
}

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 >= 10.0) {
            System.out.print("Business is profitable.");
        }
    }
}
```

```
        } else {
            System.out.print("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

Rajalakshmi Engineering College

Name: samyuktha ss

Email: 241001217@rajalakshmi.edu.in

Roll no:

Phone: 7539908242

Branch: REC

Department: IT - Section 2

Batch: 2028

Degree: B.E - IT

Scan to verify results



2024_28_III_OOPS Using Java Lab

REC_2028_OOPS using Java_Week 6_MCQ

Attempt : 1

Total Mark : 15

Marks Obtained : 15

Section 1 : MCQ

1. What will be the output of the following Java program?

```
class A {  
    int value = 10;  
    void display() {  
        System.out.println("A's display: " + value);  
    }  
}  
class B extends A {  
    int value = 20;  
    void display() {  
        System.out.println("B's display: " + value);  
    }  
}  
class Test {  
    public static void main(String[] args) {
```

```
A obj = new B();
obj.display();
System.out.println("Value: " + obj.value);
}
}
```

Answer

B's display: 20 Value: 10

Status : Correct

Marks : 1/1

2. What will be the output of the following program?

```
class A {
    public int i;
    private int j;
}
class B extends A {
    void display() {
        super.j = super.i + 1;
        System.out.println(super.i + " " + super.j);
    }
}
class inheritance {
    public static void main(String args[]) {
        B obj = new B();
        obj.i=1;
        obj.j=2;
        obj.display();
    }
}
```

Answer

Compile Time Error

Status : Correct

Marks : 1/1

3. What will be the output of the following Java program?

```
class Test {  
    void show(int a) {  
        System.out.println("Integer method");  
    }  
    void show(String s) {  
        System.out.println("String method");  
    }  
    public static void main(String[] args) {  
        Test obj = new Test();  
        obj.show(null);  
    }  
}
```

Answer

String method

Status : Correct

Marks : 1/1

4. What will be the output of the following Java program?

```
class Test {  
    void display(int a, int b) {  
        System.out.println("Method 1");  
    }  
    void display(double a, double b) {  
        System.out.println("Method 2");  
    }  
    public static void main(String[] args) {  
        Test obj = new Test();  
        obj.display(10, 10.0);  
    }  
}
```

Answer

Method 2

Status : Correct

Marks : 1/1

5. What will be the output of the following Java program?

```
class Vehicle {  
    void start() {  
        System.out.println("Vehicle starts");  
    }  
}  
class Car extends Vehicle {  
  
    void start() {  
        System.out.println("Car starts");  
    }  
}  
class ElectricCar extends Car {  
    void start() {  
        System.out.println("Electric Car starts silently");  
    }  
}  
class Test {  
    public static void main(String[] args) {  
        Vehicle v = new ElectricCar();  
        v.start();  
    }  
}
```

Answer

Electric Car starts silently

Status : Correct

Marks : 1/1

6. What will be the output of the following code?

```
class A {  
    void display() {  
        System.out.println("Display A");  
    }  
}
```

```
class B extends A {
```

```

void display() {
    System.out.println("Display B");
}
}

class C extends B {
    void display() {
        super.display();
    }
}

class Test {
    public static void main(String[] args) {
        C obj = new C();
        obj.display();
    }
}

```

Answer

Display B

Status : Correct

Marks : 1/1

7. What will be the output of the following Java program?

```

class Vehicle {
    void startEngine() {
        System.out.println("Vehicle engine started");
    }
}

```

```

class Car extends Vehicle {
    void startEngine() {
        System.out.println("Car engine started");
    }
}

```

```

class Main {
    public static void main(String[] args) {

```

```
        Vehicle myVehicle = new Car();
        myVehicle.startEngine();
    }
}
```

Answer

Car engine started

Status : Correct

Marks : 1/1

8. What will be the output of the following code?

```
class A {
    int sum(int x) {
        return x + 2;
    }
}

class B extends A {
    int sum(int x) {
        return super.sum(x) * 2;
    }
}

class C extends B {
    int sum(int x) {
        return super.sum(x) - 3;
    }
}

class Test {
    public static void main(String[] args) {
        C obj = new C();
        System.out.println(obj.sum(4));
    }
}
```

Answer

Status : Correct

Marks : 1/1

9. What will be the output of the following Java program?

```
class Parent {  
    void show() {  
        System.out.println("Parent class");  
    }  
}  
class Child extends Parent {  
    void show() {  
        System.out.println("Child class");  
    }  
}  
class Test {  
    public static void main(String[] args) {  
        Parent obj = new Child();  
        obj.show();  
    }  
}
```

Answer

Child class

Status : Correct

Marks : 1/1

10. Which of the following is the correct way for class B to inherit from class A?

Answer

class B extends A {}

Status : Correct

Marks : 1/1

11. Select the correct keyword for implementing inheritance through the class.

Answer

extends

Status : Correct

Marks : 1/1

12. What will be the output of the following program?

```
class Vehicle {  
    String type = "Vehicle";  
}
```

```
class Car extends Vehicle {  
    String type = "Car";  
}
```

```
class Test {  
    public static void main(String[] args) {  
        Car c = new Car();  
        System.out.println(c.type);  
    }  
}
```

Answer

Car

Status : Correct

Marks : 1/1

13. Which of the following is true about method overriding in Java?

Answer

The method must have the same name, same parameters, and must be in different classes with an inheritance relationship

Status : Correct

Marks : 1/1

14. What will be the output of the following program?

```
class A {  
    int x = 10;
```

```
}

class B extends A {
    int x = 20;
}

class C extends B {
    int x = 30;

    void display() {
        System.out.println(x);
        System.out.println(super.x);
    }
}
```

```
class Test {
    public static void main(String[] args) {
        C obj = new C();
        obj.display();
    }
}
```

Answer

3020

Status : Correct

Marks : 1/1

15. What will be the output of the following Java program?

```
class A {
    void display() {
        System.out.println("Class A");
    }
}
```

```
class B extends A {
    void show() {
        System.out.println("Class B");
    }
}
```

```
}
```

```
class C extends B {
```

```
    void print() {
```

```
        System.out.println("Class C");
```

```
    }
```

```
}
```

```
class Test {
```

```
    public static void main(String[] args) {
```

```
        C obj = new C();
```

```
        obj.display();
```

```
        obj.show();
```

```
        obj.print();
```

```
    }
```

```
}
```

Answer

Class A Class B Class C

Status : Correct

Marks : 1/1

Rajalakshmi Engineering College

Name: samyuktha ss

Email: 241001217@rajalakshmi.edu.in

Roll no:

Phone: 7539908242

Branch: REC

Department: IT - Section 2

Batch: 2028

Degree: B.E - IT

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 6_Q1

Attempt : 1

Total Mark : 10

Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Elsa subscribes to a premium service with a base monthly cost, a service tax and an extra feature cost. Assist her in writing an inheritance program that takes input for these values and calculates the total monthly cost.

Refer to the below class diagram:

Input Format

The first line of input consists of a double value, representing the base monthly cost.

The second line consists of a double value, representing the service tax.

The third line consists of a double value, representing the extra feature cost.

Output Format

The output prints "Rs. X" where X is a double value, rounded off to two decimal places.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 10.0

2.5

5.0

Output: Rs. 17.50

Answer

```
import java.util.Scanner;  
  
import java.util.Scanner;  
  
class Subscription {  
    protected double baseCost;  
  
    public Subscription(double baseCost) {  
        this.baseCost = baseCost;  
    }  
}  
  
class PremiumSubscription extends Subscription {  
    private double serviceTax;  
    private double extraFeatureCost;  
  
    public PremiumSubscription(double baseCost, double serviceTax, double  
extraFeatureCost) {  
        super(baseCost);  
        this.serviceTax = serviceTax;  
        this.extraFeatureCost = extraFeatureCost;  
    }  
  
    public double calculateMonthlyCost() {  
        return baseCost + serviceTax + extraFeatureCost;  
    }  
}
```

```
}

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

        double baseMonthlyCost = scanner.nextDouble();
        double serviceTax = scanner.nextDouble();
        double extraFeatureCost = scanner.nextDouble();

        PremiumSubscription premiumSubscription = new
PremiumSubscription(baseMonthlyCost, serviceTax, extraFeatureCost);

        double totalMonthlyCost = premiumSubscription.calculateMonthlyCost();

        System.out.printf("Rs. %.2f%n", totalMonthlyCost);

        scanner.close();
    }
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: samyuktha ss

Email: 241001217@rajalakshmi.edu.in

Roll no:

Phone: 7539908242

Branch: REC

Department: IT - Section 2

Batch: 2028

Degree: B.E - IT

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 6_Q2

Attempt : 1

Total Mark : 10

Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Alice is managing an online store and wants to implement a program using inheritance to calculate the selling price of products after applying discounts.

Guide her by following the instructions:

Create a base class called Product with a public double attribute price. Create a subclass called DiscountedProduct, which extends Product and includes a private double attribute discount rate. This subclass has a method called calculateSellingPrice() to determine the final selling price after applying the discount.

Formula: Discounted selling price = price * (1 - discount rate)

Input Format

The first line of input consists of a double value p , the initial price of the product.

The second line consists of a double value d , the discount rate.

Output Format

The output prints "Rs. X", where X is a double value, representing the calculated discounted selling price, rounded off to two decimal places.

If the discount rate is greater than 1, print "Not applicable".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 50.00

0.20

Output: Rs. 40.00

Answer

```
import java.util.Scanner;  
import java.util.Scanner;  
  
class Product {  
    public double price;  
  
    public Product(double price) {  
        this.price = price;  
    }  
}  
  
class DiscountedProduct extends Product {  
    private double discountRate;  
  
    public DiscountedProduct(double price, double discountRate) {  
        super(price);  
        this.discountRate = discountRate;  
    }  
  
    public double calculateSellingPrice() {
```

```
        return price * (1 - discountRate);
    }

    public boolean isDiscountApplicable() {
        return discountRate <= 1.0;
    }
}

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

        double initialPrice = scanner.nextDouble();
        double discountRate = scanner.nextDouble();
        DiscountedProduct discountedProduct = new
DiscountedProduct(initialPrice, discountRate);
        double sellingPrice = discountedProduct.calculateSellingPrice();

        if (sellingPrice >= 0) {
            System.out.printf("Rs. %.2f%n", sellingPrice);
        } else {
            System.out.println("Not applicable");
        }
        scanner.close();
    }
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: samyuktha ss

Email: 241001217@rajalakshmi.edu.in

Roll no:

Phone: 7539908242

Branch: REC

Department: IT - Section 2

Batch: 2028

Degree: B.E - IT

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 6_Q3

Attempt : 1

Total Mark : 10

Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Preethi is working on a project to automate sales tax calculations for items in a store. She wants to create a program that takes the price of an item and the sales tax rate as input and calculates the final price of the item after applying the sales tax.

Write a program using the class SalesTaxCalculator, which contains an overloaded method named calculateFinalPrice to handle both integer and double inputs. The program should also include a Main class that takes user input, calls the appropriate method from SalesTaxCalculator, and prints the final price of the item.

Formula Used: Final price = price + ((price * sales tax rate) / 100)

Input Format

The first line of input consists of an integer price (the price of the item for integer inputs).

The second line of input consists of an integer taxRate (the sales tax rate for integer inputs).

The third line of input consists of a double price (the price of the item for double inputs).

The fourth line of input consists of a double taxRate (the sales tax rate for double inputs).

Output Format

The first line of output prints an integer, representing the final price of the item after applying the sales tax for integer inputs (a and b).

The second line prints a double value, representing the final price of the item after applying the sales tax for double-value inputs (m and n), rounded to two decimal places.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 100

10

100.0

5.0

Output: 110

105.00

Answer

```
import java.util.Scanner;  
class SalesTaxCalculator {
```

```
    public static int calculateFinalPrice(int price, int taxRate) {  
        return price + (price * taxRate) / 100;  
    }
```

```
public static double calculateFinalPrice(double price, double taxRate) {  
    return price + ((price * taxRate) / 100);  
}  
}  
  
class Main {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        int intPrice = scanner.nextInt();  
        int intTaxRate = scanner.nextInt();  
        double doublePrice = scanner.nextDouble();  
        double doubleTaxRate = scanner.nextDouble();  
  
        int finalPriceInt = SalesTaxCalculator.calculateFinalPrice(intPrice,  
intTaxRate);  
        double finalPriceDouble =  
SalesTaxCalculator.calculateFinalPrice(doublePrice, doubleTaxRate);  
  
        System.out.println(finalPriceInt);  
        System.out.format("%.2f", finalPriceDouble);  
    }  
}
```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: samyuktha ss

Email: 241001217@rajalakshmi.edu.in

Roll no:

Phone: 7539908242

Branch: REC

Department: IT - Section 2

Batch: 2028

Degree: B.E - IT

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 6_Q4

Attempt : 1

Total Mark : 10

Marks Obtained : 10

Section 1 : Coding

1. Problem Statement

Mr.Kapoor wants to create a program to calculate the volume of a Cuboid and a Cube using method overriding.

Implements a base class Cuboid with attributes for length, width, and height. Include a method calculateVolume() that computes the volume of the cuboid.

Extends the base class with a subclass Cube representing a cube, where all sides are equal. Override the calculateVolume() method in the Cube class to compute the volume of the cube.

The program should take user input for the dimensions of the cuboid and the side length of the cube and display the calculated volumes with two decimal places.

Input Format

The first line of input consists of 3 space-separated double values, representing the cuboid length, width, and height, respectively.

The second line consists of a double value, representing the side length of the cube.

Output Format

The first line of output prints the volume of the cuboid, rounded off to two decimal places.

The second line prints the volume of the cube, rounded off to two decimal places.

Refer to the sample output for formatting specifications.

Sample Test Case

Input: 60.0 60.0 60.0
50.0

Output: Volume of Cuboid: 216000.00
Volume of Cube: 125000.00

Answer

```
import java.util.Scanner;  
  
import java.util.Scanner;  
  
class Cuboid {  
    double length, width, height;  
  
    Cuboid(double length, double width, double height) {  
        this.length = length;  
        this.width = width;  
        this.height = height;  
    }  
}
```

```

        double calculateVolume() {
            return length * width * height;
        }
    }

class Cube extends Cuboid {

    Cube(double side) {
        super(side, side, side);
    }

    @Override
    double calculateVolume() {
        return length * length * length;
    }
}

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

        double cuboidLength = scanner.nextDouble();
        double cuboidWidth = scanner.nextDouble();
        double cuboidHeight = scanner.nextDouble();

        // Regular object instantiation for Cuboid
        Cuboid cuboid = new Cuboid(cuboidLength, cuboidWidth, cuboidHeight);
        System.out.printf("Volume of Cuboid: %.2f\n", cuboid.calculateVolume());

        double cubeSide = scanner.nextDouble();

        // Upcasting - Using superclass reference for subclass object (DMD)
        Cuboid cube = new Cube(cubeSide); // Upcasting
        System.out.printf("Volume of Cube: %.2f", cube.calculateVolume()); // Calls
        Cube's method dynamically

        scanner.close();
    }
}

```

Status : Correct

Marks : 10/10

Rajalakshmi Engineering College

Name: samyuktha ss

Email: 241001217@rajalakshmi.edu.in

Roll no:

Phone: 7539908242

Branch: REC

Department: IT - Section 2

Batch: 2028

Degree: B.E - IT

Scan to verify results



2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 6_Q5

Attempt : 1

Total Mark : 10

Marks Obtained : 10

Section 1 : Coding

1. Problem statement:

Tim was tasked with developing a grocery shopping app. You have a class hierarchy that includes Item, Produce, and OrganicProduce. Your goal is to calculate the total cost of a shopping list, which may contain a mix of regular produce and organic produce items. Additionally, you need to apply discounts to organic items. Apply a 10% discount on organic produce items

Class Hierarchy:

Item: Base class for all items.

Produce: Subclass of Item for regular produce items.

OrganicProduce: Subclass of Produce for organic produce items.

Input Format

The first line of input consists of an integer, 'n'.

For each 'n' item, the user will provide:

- A string 'type' representing the item type ('Regular' or 'Organic').
- A string 'name' represents the item name.
- A double 'price' represents the item price.

Output Format

The output will display the total cost of the shopping list, including discounts on organic items.

Refer to the sample output for format specifications.

Sample Test Case

Input: 1

Regular Banana 1.99

Output: 1.99

Answer

```
import java.util.Scanner;  
import java.util.Scanner;  
  
class Item {  
    String name;  
    double price;  
  
    Item(String name, double price) {  
        this.name = name;  
        this.price = price;  
    }  
  
    double calculateCost() {  
        return price;
```

```
        }
    }

class Produce extends Item {
    Produce(String name, double price) {
        super(name, price);
    }
}
```

```
class OrganicProduce extends Produce {
    OrganicProduce(String name, double price) {
        super(name, price);
    }
}
```

```
@Override
double calculateCost() {

    return price * 0.9;
}
}
```

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

```
        int n = sc.nextInt();
        sc.nextLine(); // Consume newline
```

```
        double totalCost = 0.0;
```

```
        for (int i = 0; i < n; i++) {
            String type = sc.next();
            String name = sc.next();
            double price = sc.nextDouble();

            if (type.equals("Regular")) {
```

```
                Item item = new Produce(name, price);
                totalCost += item.calculateCost();
            } else if (type.equals("Organic")) {
```

```
                Item item = new OrganicProduce(name, price);
                totalCost += item.calculateCost();
```

```
        }  
    }  
    System.out.printf("%.2f%n", totalCost);  
}
```

Status : Correct

Marks : 10/10