

# Rajalakshmi Engineering College

Name: Roshan S

Email: 241501170@rajalakshmi.edu.in

Roll no: 241501170

Phone: 9677031330

Branch: REC

Department: AI & ML - Section 1

Batch: 2028

Degree: B.E - AI & ML

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 {  
    protected String name;  
    protected double price;  
  
    public Item(String name, double price) {  
        this.name = name;  
        this.price = price;  
    }  
  
    public double calculateCost() {  
        return price;  
    }  
}
```

```
class Produce extends Item {  
    public Produce(String name, double price) {  
        super(name, price);  
    }  
  
    @Override  
    public double calculateCost() {  
        return price; // no discount  
    }  
}  
  
class OrganicProduce extends Produce {  
    public OrganicProduce(String name, double p  
        super(name, price);  
    }  
  
    @Override  
    public double calculateCost() {  
        return price * 0.90; // 10% discount  
    }  
}
```

```
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