12214994_SaurabhRana

Case Study: Sugar Bliss Bakery

Amit owns a newly opened bakery named **Sugar Bliss**, which sells different varieties of chocolates. To attract customers during festive seasons, the bakery offers **discounts on selected chocolate flavours**. Amit wants to automate the billing process by developing a **C# console-based application** that calculates the final payable amount after applying the appropriate discount.

You are required to help Amit by developing this application using **C# classes and objects**.

Requirements

Class Specification

Create a class named **Chocolate** with the following **public properties**:

Data Type	Property Name
string	Flavour
int	Quantity
int	PricePerUnit
double	TotalPrice
double	DiscountedPrice

Method in Chocolate Class (5 Marks)

```
public bool ValidateChocolateFlavour()
```

- This method validates the chocolate flavour.
- If the flavour is **Dark**, **Milk**, or **White**, return `true`.
- Otherwise, return `false`.
- **Note:** Flavour is case-sensitive.

Discount Details

Flavour	**Discount Percentage**
Dark	18%
Milk	12%
White	6%

Price Calculation Formula

- **Total Price** = Quantity × PricePerUnit
- **Discounted Price** = Total Price − (Total Price × Discount Percentage / 100)

Method in Program Class (10 Marks)

```
public Chocolate CalculateDiscountedPrice(Chocolate chocolate)
```

This method should:

- Calculate the total price
- Apply the discount based on the selected flavour
- Return the updated Chocolate object

Main Method Instructions

Accept user input for:

  - Chocolate flavour
  - Quantity
  - Price per unit

2. Call `ValidateChocolateFlavour()`:
    - If the method returns `false`, display **“Invalid flavour”**
3. If the flavour is valid:
    - Call `CalculateDiscountedPrice()`
    - Display the following details:
        - Flavour
        - Quantity
        - Price per unit
        - Total price
        - Discounted price

Notes

- All classes, properties, and methods must be **public**
- Do **not** use `Environment.Exit()`
- Do **not** modify method names or return types
- Follow proper **object-oriented programming principles**
- Output must match the sample format exactly

Sample Input & Output

Sample Input 1

```
Enter the flavour
Dark
Enter the quantity
5
Enter the price per unit
20
```

Sample Output 1

```
Flavour : Dark
Quantity : 5
Price Per Unit : 20
Total Price : 100
Discounted Price : 82
```

Sample Input 2

```
Enter the flavour
Fruit
Enter the quantity
8
Enter the price per unit
12
```

Sample Output 2

```
Invalid flavour
```

```
public class Chocolate

{
    public string Flavour { get; set; }

    public int Quantity { get; set; }

    public int PricePerUnit { get; set; }

    public double TotalPrice { get; set; }

    public double DiscountedPrice { get; set; }


    public bool ValidateChocolateFlavour()

    {

        if (Flavour == "Dark" || Flavour == "Milk" || Flavour == "White")

        {

            return true;

        }

        return false;

    }

}

public class Program

{

    public Chocolate CalculateDiscountedPrice(Chocolate chocolate)

    {

        chocolate.TotalPrice = chocolate.Quantity * chocolate.PricePerUnit;


        double discountPercentage = 0;


        if (chocolate.Flavour == "Dark")

            discountPercentage = 18;

        else if (chocolate.Flavour == "Milk")

            discountPercentage = 12;

        else if (chocolate.Flavour == "White")

            discountPercentage = 6;
        chocolate.DiscountedPrice =
            chocolate.TotalPrice - (chocolate.TotalPrice * discountPercentage / 100);

        return chocolate;
    }

    public static void Main()
    {
        Program program = new Program();
        Chocolate chocolate = new Chocolate();

        Console.WriteLine("Enter the flavour");
        chocolate.Flavour = Console.ReadLine();

        Console.WriteLine("Enter the quantity");
        chocolate.Quantity = Convert.ToInt32(Console.ReadLine());

        Console.WriteLine("Enter the price per unit");
        chocolate.PricePerUnit = Convert.ToInt32(Console.ReadLine());

        if (!chocolate.ValidateChocolateFlavour())
        {
            Console.WriteLine("Invalid flavour");
        }
        else
        {
            chocolate = program.CalculateDiscountedPrice(chocolate);
```

```
            Console.WriteLine("Flavour : " + chocolate.Flavour);

            Console.WriteLine("Quantity : " + chocolate.Quantity);

            Console.WriteLine("Price Per Unit : " + chocolate.PricePerUnit);

            Console.WriteLine("Total Price : " + chocolate.TotalPrice);

            Console.WriteLine("Discounted Price : " + chocolate.DiscountedPrice);

        }

    }

}
```