

UNiDAYS Discounts Coding Challenge

1. Build and Run the Code

The code was written in C++ using Visual Studio 2017. To build the code go to **Build | Build Solution** in the menu bar or press **F7**. To run the code go to **Debug | Start Without Debugging** or press **Ctrl+F5**.

2. User Input

The user is asked to add an item to the basket. The input should be upper or lowercase A, B, C, D or E. The program checks the input validity. If it is not valid, the program asks the user for another input. The loop will continue until the user enters the correct input.

3. Approach

The file containing the class is separated from the file containing the rest of the program, making it easier to read and understand the code.

For this problem, I decided to use just the two mandatory public methods, *AddToBasket(item)* and *CalculateTotalPrice()*.

- *AddToBasket(item)* method:
 - The variable *item[ItemName]Number* is incremented after each item added to the basket. The variable will store the total number of items with the same name added to the basket.
- *CalculateTotalPrice()* method:
 - When calculating the discount price for each item, first, the program has to check the total number of items with the same name with the pricing rules. After that, it will divide that number to either 2 or 3 (depending on the pricing rule) and it will multiply it by the discount price (set by the pricing rule). Then, if the pricing rule is not valid for the total number of items with the same name, the program will add the normal price of the items that do not follow the pricing rule.

4. Pricing Rules

- Pricing rule for item B: 2 for £20
 - If the total number of items named B is a multiple of 2, the discount can be applied. The final price, with discount, for item B is calculated using the following formula:

*price for item B (with discount) = (total number of items named B)/2 * the discount price, where the discount price = £20.*

- If the number of items named B is not a multiple of 2, the program will subtract 1 from the total number of items named B when checking the condition. Next, the above formula will have a slight change:

*price for item B (with discount) = (total number of items named B)/2 * the discount price + initial price for item B, where the discount price = £20 and initial price for item B = £12.*

- Pricing rule for item C: 3 for £10
 - If the total number of items named C is a multiple of 3, the discount can be applied. The final price, with discount, for item C is calculated using the same formula as above except that the values of *the discount price* and *initial price for item C* will change accordingly.
 - If the total number of items named C is not a multiple of 3, the program will subtract 1 or 2 from the total number of items named C when checking the condition, and then in the formula, the *initial price for item C* will be multiplied by 1 or 2.
- Pricing rule for item D: Buy 1 get 1 free
 - The exact same formula applies as for item B. The *discount price* and the *initial price for item D* are going to be £7.
- Pricing rule for item E: 3 for the price of 2
 - The same concept as for item C applies. This change was made in the formula: the *discount price* will be multiplied by 2.