



MALMÖ HÖGSKOLA

**Malmö University
School of Technology**

Programming Using C#, Basic Course

Grocery Store - Console Application

Assignment 1 Mandatory

[Farid Naisan](#)

University Lecturer
Department of Computer Science

Assignment 1: VAT Calculator

1. Objectives

- To work with simple classes and objects
- To create a first C# program as a Console Application
- To work with primitive data types such as text, numbers and logical values.
- To use simple variables to store values and write methods to manipulate them.
- Write data to a console window by using the **Console.Write** and **Console.WriteLine**.
- Read data from a console using the **Console.Read** and **Console.ReadLine**.

You are expected to have studied the lecture documents published on the module, done the optional exercises and even have checked the recommended Internet links before starting this assignment. Trial-and-error is not a good method. It will be a waste of your as well as your teachers' precious time. Key to success is that you understand every step you take in your solution. Take your time and be patient!! As soon as you realize that you are stuck with a problem, go to the forum in the module and start a thread. There will many who would like to help you. Again, no question is dumb; it is dumb not to question!

2. Description

This assignment consists of two parts, one mandatory part required for a passing grade (Pass, G) and a second part that is optional but required if you aim at receiving a higher grade (Pass with Distinction, VG). In the first part, you are given a problem with some instructions to solve it. In the next part, you are asked to find and implement a task similar to Part 1 by yourself.

Apu has opened a gas station including a little shop for selling fast food and groceries. Your job in Part 1 is to write a console application (a program) for Apu to help him calculate a governmental tax known as Value Added Tax (VAT). Different rate percent applies depending on whether the product is categorized as food or a non-food item. The program should read the user input, name, price, quantity and type of a product and then calculate the total amount payable by the customer and the included amount of VAT. To accomplish this task, you need to write two classes, one **Product** class that handles all data and operations related to the product objects and another class, **GroceryStore** that serves as a start class and uses the operation that you program in the Product class.

In Part 2 (optional), you are given the opportunity to choose any object type, such TV, House, Address, etc and write a class for it just as the **Product** class. You write then a start class that tests the objects as in Part 1.

3. Work Plan

- Do the Part 1 first and then Part 2
- In both of these parts, you write two classes, one with the method Main and one that programs an object, for example Product in Part 1
- Create a C# Console Application project using Visual Studio (VS). VS will create a solution for you and put the project into this solution. In this document we refer to this project as Assignmen1.
- When creating the project, select a new folder for this assignment.
- Upon creation of the application, VS's IDE creates also a default start class, **program.cs**, for you in which it has prepared the Main method.
- Rename the **program.cs** to **GroceryStore.cs** in the Solution Explorer (right-click on the project name). VS will ask you if you will also change the name of the class in the code accordingly, something that you should happily accept. It will be inside the Main method where we will then create an instance of the **Product** class to start the program. Don't forget to save your project often.
- Save your work often when coding (every 5 minutes). Make sure you always keep a backup of your files.

4. Assessment

In order to get a passing grade, Part 1 must be completed. To get a higher grade, even Part 2 must be carried out with good quality considerations. Unsatisfactorily done projects will be returned for completion. You have to consider (to the extent applicable to this assignment) the terms explained in the document "Quality Standards and Guidelines", available on the module.

Mandatory for both parts: Two classes, one start class with the Main method (for ex **GroceryStore.cs**) and one for solving the main problem (for ex Product.cs). The program class should not contain any instance variables or methods other than the Main Method.

5. New Terms

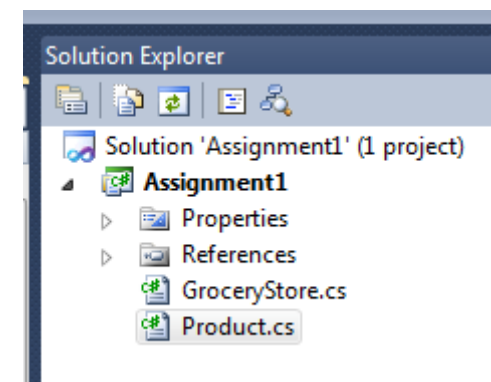
User	A person that runs the program. You are the user when you run and test your program.
Read Input	The user gives input through the keyboard (or mouse) to your program. You will be using the Console.ReadLine to fetch the user's input. This method will read a whole line of input after the user has pressed the Enter key as line of text. If the user feeds a number like 5,67, the ReadLine method will transport it to your program as "5,67", i.e. a value of the string type. You must convert it the text to the desired type (int, double, string, etc), and save them in variables in your class. Do not use Console.Read as it works quite differently.
Print or show results	Show somehow the results to the user, using the Console.Write or Console.WriteLine
Application	All components that make your program.
Console Application	An application that is run on a command prompt window (see the figure below).

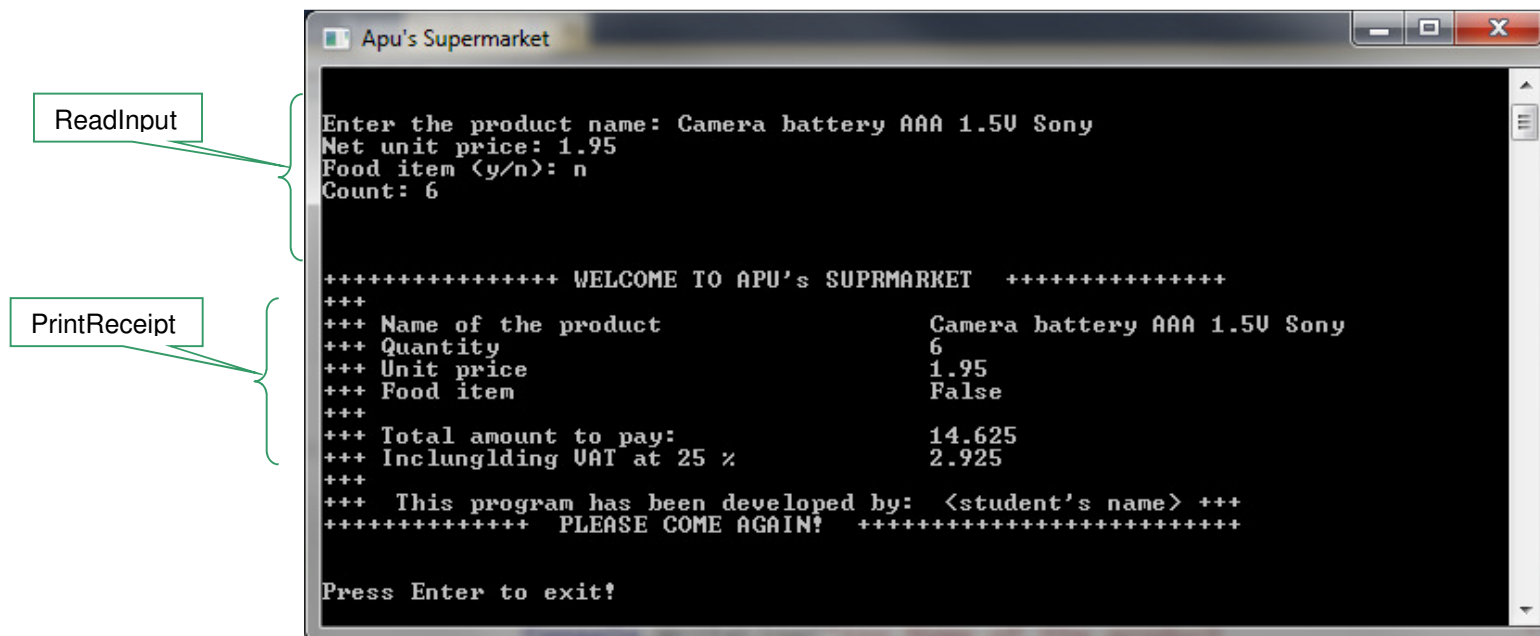
6. Part 1: GroceryStore

In Europe, consumers are subject to pay a governmental tax called Value-Added Tax (VAT) directly at purchase time. This tax is due for both goods and services. The prices are usually tagged with a final price to pay that includes the prevailing VAT, so the consumer knows exactly how much to pay when checking the price of a product.

There are two types of taxes: food items having a 12.0 % tax rate and other items having a 25% tax rate.

Write a program that asks for the **name** of the **product**, the **price**, **quantity** and the **type** of item (food or non-food) and then Calculates and Prints out a receipt showing the total cost of the item and the amount of VAT included in the total price. The figure here shows a sample project for Part 1. A solution can have more than one project. You can create part 2 also in the same solution; right-click on the Solution name and select Add, New Project. (In VS Ultimate version, you have to set on the Solution option via the Tools menu). The figure below demonstrates a sample program execution.





```

Enter the product name: Camera battery AAA 1.5V Sony
Net unit price: 1.95
Food item (y/n): n
Count: 6

+++++ WELCOME TO APU'S SUPRMARKET +++++
+++
+++ Name of the product           Camera battery AAA 1.5V Sony
+++ Quantity                     6
+++ Unit price                   1.95
+++ Food item                    False
+++
+++ Total amount to pay:         14.625
+++ Inclunglding VAT at 25 %    2.925
+++
+++ This program has been developed by: <student's name> +++
+++++ PLEASE COME AGAIN! +++++

Press Enter to exit!
  
```

Note: Replace the <student's name> by your own name!

6.1 The Product class

Create a new class in VS and save as **Prouct.cs** in your project directory. Use this file to write the class Product according to the following guidelines:

- 6.1.1 **Instance Variables:** Declare fields (variables) in your class for saving input: name (**string**), count (**int**), unitPrice (**double**), foodItem (**bool**). Declare also variables for storing output: totalVAT (**double**) and totalNetValue (**double**). You can hard-code the values of VAT rate using constants as follows:

```
private const double foodVATRate = 0.12, otherVATRate = 0.25;
```

Note: In later assignments, we will learn to reduce the number of output variables as much as possible by replacing them with methods, in which the value is calculated and returned every time the method is called.

6.1.2 **Methods:** The following methods are to be written:

6.1.2.1 Write a method, **Start** that encapsulates the following steps:

- Read input from the keyboard, (name, count, unit price, type of item, (food or non-food) and save them into instance variables.
- Calculate the total tax and the total amount to pay.
- Print a receipt using the calculated values (output) as shown in the example above.
- To help you here, the skeleton of some of the methods you might need to write are provided below.

```
public void Start()
{
    //Read input
    ReadInput();

    //Calculate total tax
    CalculateValues();

    //Calculate totalNetPrice, total price
    PrintReceipt();
}

private void ReadInput()
{
    //1. Read name of the product
    ReadName();

    //2. Read price without VAT
    ReadNetUnitPrice();

    //3. Ask the user if the item is a food item
    ReadIfFoodItem();

    //4. Read number of items (quantity)
    ReadCount();
}
```

6.1.2.2 Write the following methods into your class and write the methods called in the **ReadInput** method. You may of course write your own methods as long as you maintain a good code structure.

6.1.2.3 The method **ReadInput** requires four methods as listed in the this image. Write these methods by yourself. The methods are used internally in the class and therefore should be declared as private.

To help you with reading the type of product (food or non-food), you may copy and use the following code to your class:

```
68 private void ReadIfFoodItem()  
69 {  
70     Console.Write("Food item (y/n): ");  
71  
72     char response = char.Parse(Console.ReadLine());  
73  
74     if ( (response == 'y') || (response == 'Y') )  
75         foodItem = true;  
76     else  
77         foodItem = false;  
78 }  
--
```

The statement on line 74 is a conditional statement (covered in the next module) and if the value of **response** begins with a y or a Y, the **foodItem** variable will get a value **true** and otherwise **false**.

When you are done with above, you are done with the ReadInput. You now have to write the two other methods contained in the **Start** method above.

6.1.2.4 **CalculateValues** that computes the total net price (count * net price) and total amount of VAT and saves the results in the output variables that you have declared for them.

6.1.2.5 **PrintReceipt**: that prints to the console window a receipt as shown earlier, using the input and output variables.

```
//Calculate totalNetValue TotalVat  
private void CalculateValues()  
{  
    //Your implementation here...  
}
```

```
// Print results  
private void PrintReceipt()  
{  
    //Your implementation here...  
}
```

6.2 The GroceryStore Class

- 6.2.1 In order to test the Product class, we need to complete the program by a program class with a method Main. This method will be the one the CLR (the Common Language Runtime) will look for to start your program. Although you can have methods and instance variables in this class, just like any other class, it is recommended that you keep this class as small as possible. It should only contain the **Main** method and shall not have any instance variables.
- 6.2.2 Create an object of the Product class (see below) in the Main method and then call its Start method. This method, if programmed as above, reads the input from the user, calculates output using the input data and save the results. It also prints the results to the screen.

Three steps to remember always:

- Read and save input,
- Perform the required calculations (process the data), and
- Display results

Most of the solutions fit into this model.

Note: In case you wonder why the code snippets given in this document are images and not text so you could copy and paste, it should be clarified that based on experience, it is more effective to learn the details when you copy the code by rewriting it.

```
/// <summary>
/// This program computes total VAT and the total amount for a product,
/// given the net price, type of product and the corresponding tax rate.
/// The program then displays the results in form of a receipt.
///
/// This class creates an object (instance) of the Product class which
/// encapsulates all data and operations related to a product.
/// </summary>
class GroceryStore
{
    static void Main(string[] args)
    {
        Console.Title = "Apu's Supermarket";

        //Declare and create an instance of the Product class.
        Product product = new Product(); //object created with keyword new

        //Call the product-object's Start method to perform all
        //steps required
        product.Start();

        //The command prompt window will close quickly after the program has executed
        //the instructions under the Start method. To cause the window to stay, put
        //the application in a reading mode, using the ReadLine method.
        Console.WriteLine("Press Enter to exit!");
        Console.ReadLine();
    }
}
```




7. Part 2: Program Your Object (for Grade VG)

- 7.1 Look around at home, at your work or where you are now. You will find numerous objects, a chair, a baby, a friend, a car, etc around you. Choose your **one** favorite object and program it as in Part 1.
- 7.2 For the object you have chosen, determine at least three attributes (fields) that best describes the objects of this type. Determine then a data type ([string](#), [int](#), [double](#) etc) that the attribute can be represented by. For an Address class, the street, city, and zip code could be good fields. These all can be declared as [string](#) as all the three items contain text.
- 7.3 Think about at least two operations (methods) that can be performed on the object using the fields.
- 7.4 The object does not have to be a physical thing; it may be a conceptual object, a calculation object or an object like "Recipe", "Movie", etc.
- 7.5 Create a new Project in your Solution in VS and follow the same procedure as in Part 1. To avoid resubmission, have in mind the following:
 - 7.5.1 You must create two classes as in Part 1.
 - 7.5.2 All instance variables must be declared as [private](#). No variable should be declared [public](#).

8. Submission

Compress all your files including the Properties folder using a **zip** or **rar** format (as explained in the document "Quality Standards and Guidelines"). Go to the Assignment page (where you downloaded this document) in Its L, click the Submit Answer button and upload the compressed file. Do not upload single files and do not send your solution by email even when Its L is not down (doesn't happen often). Its L is the only way to make your submission and it is there, your teachers can record your grade.

Good Luck!

Programming is fun. Never give up. Ask for help!

Farid Naisan,

Course Responsible and Instructor