



MALMÖ HÖGSKOLA
Centrum för teknikstudier

Malmö University
School of Technology

Programmering med C#, grundkurs
Programming Using C#, Basic Course

VAT Calculator

Assignment 1 – Console Application Mandatory

[Farid Naisan](#)
University Lecturer
Department of Computer Science



Introduction:

In this assignment you will create a tax calculator program and define one of your own objects in order to get more training in getting started.

The following section will discuss the Quality Standards and Guidelines for this assignment, some of the topics there may be hard to grasp as you haven't read the description for the rest of this assignment. Please skim through this section to start with, if there are things you don't understand, read the rest of the assignment and then, after you have completed the assignment, read this section again in order to see if your solution achieves the Quality Standards and Guidelines required..."

Quality Standards and Guidelines

Requirements:

- ✓ The project must contain at least two classes, for example **Product** and **TaxProgram** in both Part 1 and Part 2 of this assignment, and every class is to be saved in a separate file, for example **Product.cs** and **TaxProgram.cs** respectively. If you are using Visual Studio, it will prepare a default class for you.
- ✓ You can use a standard text editor such as Windows Notepad.exe to write the source code, compile and run from the command line, or use other development environments (known as IDE, Integrated Development Environment) such as Visual Studio to create the projects, write code, build and run from the IDE.
- ✓ All instance variables must be declared as private.
- ✓ Make your own assumptions whenever you find instructions unclear. Document your assumptions directly in the code or as an extra note to your instructor.
- ✓ In view of the fact that this and other assignments are designed for beginners, you may practice improvements, optimizations and enhancements to the coding structure and the user interface if you are not a beginner. However, certain instructions may be marked as mandatory in which case you are expected to follow the instructions.
- ✓ Write your name and the date in the top of all your source files (as comments).
- ✓ Do not forget to document your source code by writing comments explaining your code above or in front of your code's statements

Quality

- ✓ The application must have been compiled, tested and run satisfactorily before it is submitted. It is very important to maintain a good style and code structure and a normally functioning program, rather than presenting a well functioning program, but with a poor code quality. Higher grades are given to projects containing well-structured and reasonably documented code, and a satisfactorily working program.
- ✓ All identifiers (class names, variable names and method names) should be chosen carefully so that they express their purpose. The suggested solutions to the exercises can be taken as a guide. Short names that are not expressive (ex a, fnc, etc.) and should be strictly avoided.

Assessment:

- ✓ The assignment receives a letter grade A-F which can be then translated to a grade type of your choice. For students with a Swedish personal number, the grades will be automatically translated to the Swedish system, VG, G and U. Projects that do not meet the minimum criteria for a passing grade will be returned for complementary work. The final result for all assignments will be determined as a weighed average of all assignments plus the instructor's judgment of the progress you make throughout the course.

Submission

- ✓ The files are to be packed into a ZIP or RAR file and submitted inside It's Learning (ITS). Make sure to include all the files that are part of your project. If you are using Visual Studio for this assignment, include all your project's files and subfolders in your ZIP or RAR file. Sending only the **sln** or **cproj** file is not enough. Include even the folder **Properties**.

Although it is allowed to discuss solutions and ideas with other classmates, the assignment is done and submitted individually. Make good use of the forum on Its for this assignment, but when helping others do not put more than a few (4-5) lines of code. You must apply your own solution and you are not allowed to copy and submit same code as any other. It is expected that you have gone through all the recommended Readings, done exercises and eventual quizzes before starting with this assignment. If you understand and can follow the optional exercises, and the code examples available for this module, you shouldn't face any problem in completing this assignment.

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**.

2. Description

This assignment consists of two parts. In the first part you are given a problem with some instructions to solve it. In the next part you find and implement a task by yourself.

3. Work Plan

- Do the Part 1 first and then Part 2. In Part 1, you program the **Tax Calculator**, and in Part 2, you choose an object by yourself and program it just as in Part 1.
- In both of these parts, you write two classes, one with the method Main and one that programs an object, for example taxes in Part 1 and something that you choose in Part 2.
- Create a C# Console Application project using Visual Studio. The IDE will create a solution for you and put the project into this solution. In this document we refer to this project as Assignmen1.
- The IDE has also prepared a default **start class**, program.cs, for you in which the IDE has prepared a Main method.
- Rename the **program.cs** to **MainProgram.cs** in the Solution Explorer. 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.

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.

Mandatory for both parts: Two classes, one start class with the Main method (for ex TaxProgram.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. The input can be a text or a numerical value but is received and interpreted by your

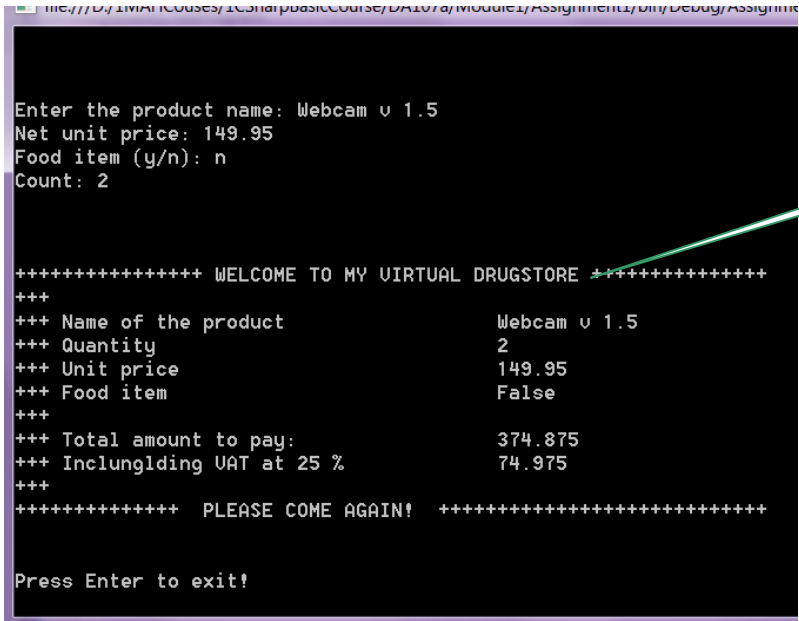
	program as text. The text needs to be converted to the correct data type (int, double, string, etc). The values are then saved in variables inside your class for manipulation.
Print or show results	Show somehow the results to the user, using the Console.Write or Console.WriteLine
Application	All components that make your program.
Console Application	An application that is run on a command prompt window (see the figure below).

6. Part 1: Drugstore

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.

A drugstore would like to have a program that computes the final sell prices for its products by applying the related VAT rate to the original net price. There are two types of taxes: food items having a 12.0 % tax rate and other items having a 25% rate.

Write a program that asks for the **name** of the **product**, the **price**, 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 below demonstrates a sample program execution.



```

file:///D:/1/MAI/Courses/1/CS/naibasiccourse/DA107a/module1/Assignment1/01/Debug/Assignme
Enter the product name: Webcam v 1.5
Net unit price: 149.95
Food item (y/n): n
Count: 2

+++++ WELCOME TO MY VIRTUAL DRUGSTORE +++++
+++
+++ Name of the product      Webcam v 1.5
+++ Quantity                2
+++ Unit price              149.95
+++ Food item               False
+++
+++ Total amount to pay:    374.875
+++ Including VAT at 25 %   74.975
+++
+++++ PLEASE COME AGAIN! +++++

Press Enter to exit!
  
```

ReadInput

PrintResults

6.1 The Product class

Create a new folder on your computer for this assignment. Create a file and save it as **Prouct.cs** in this folder. Use this file to write and store 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 in a couple of 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 sell price.
- 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 are provided below.

6.1.2.2 Write the following methods into your class and complete the internal methods used. You may of course write more of your own methods whenever you have a task that needs to be done.

```
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();
}
```

The method **ReadInput** requires four methods as listed in the above image. Write these methods by yourself. The methods are used internally in the class and therefore should be declared as `private`. As an example, the code for the `ReadIfFoodItem` is provided below.

```
private void ReadIfFoodItem()
{
    Console.Write("Food item (y/n): ");

    char response = char.Parse(Console.ReadLine());

    if ( (response == 'y') || (response == 'Y') )
        foodItem = true;
    else
        foodItem = 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.

- 6.1.2.3 **CalculateValues** that computes the total net price (count * net price) and total amount of VAT and saves the results in output variables..

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

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

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

6.2 The TaxProgram 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 to keep this class as small as possible.
- 6.2.2 Based on the above recommendation, this class should only contain the **Main** method and shall not have any instance variable.
- 6.2.3 Create an object of the Product class (see below) in the Main method and then call its Start method to so it reads the input from the user, calculate or manipulate the input and save results. Print out the results. Three steps to remember always: Read and save input, Perform the required calculations, and Print out the results,

```
/// <summary>
/// This program computes the final sell prices for products in a drugstore.
/// Given the net price, type of product and the corresponding tax rate,
/// The program calculates the final amount to be paid by a customer.
/// This class creates an object (instance) of the Product class which
/// encapsulates all data and operations about a product.
/// </summary>
class TaxProgram
{
    static void Main(string[] args)
    {
        //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
        product.Start();

        //The command prompt will close quickly after the program has executed
        //the instructions under the Start method.
        //In order to see the results, let the prompt window wait for an input (return)
        Console.WriteLine("Press Enter to exit!");
        Console.ReadLine();
    }
}
```

7. Part 2: Program Your Object

- 7.1 Look around at home, at your work or where you are now. You may 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 your 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.
- 7.3 Think about at least two operations (methods) that can be performed on the object using the attributes.
- 7.4 The object does not have to be a physical thing; it may be any of objects like "Address", "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 No variable should be declared **public**.

8. Submission

As instructed on the assignment page in It's learning.

Good Luck!

Programming is fun. Never give up. Ask for help!
Farid Naisan, Course Coordinator and Instructor