

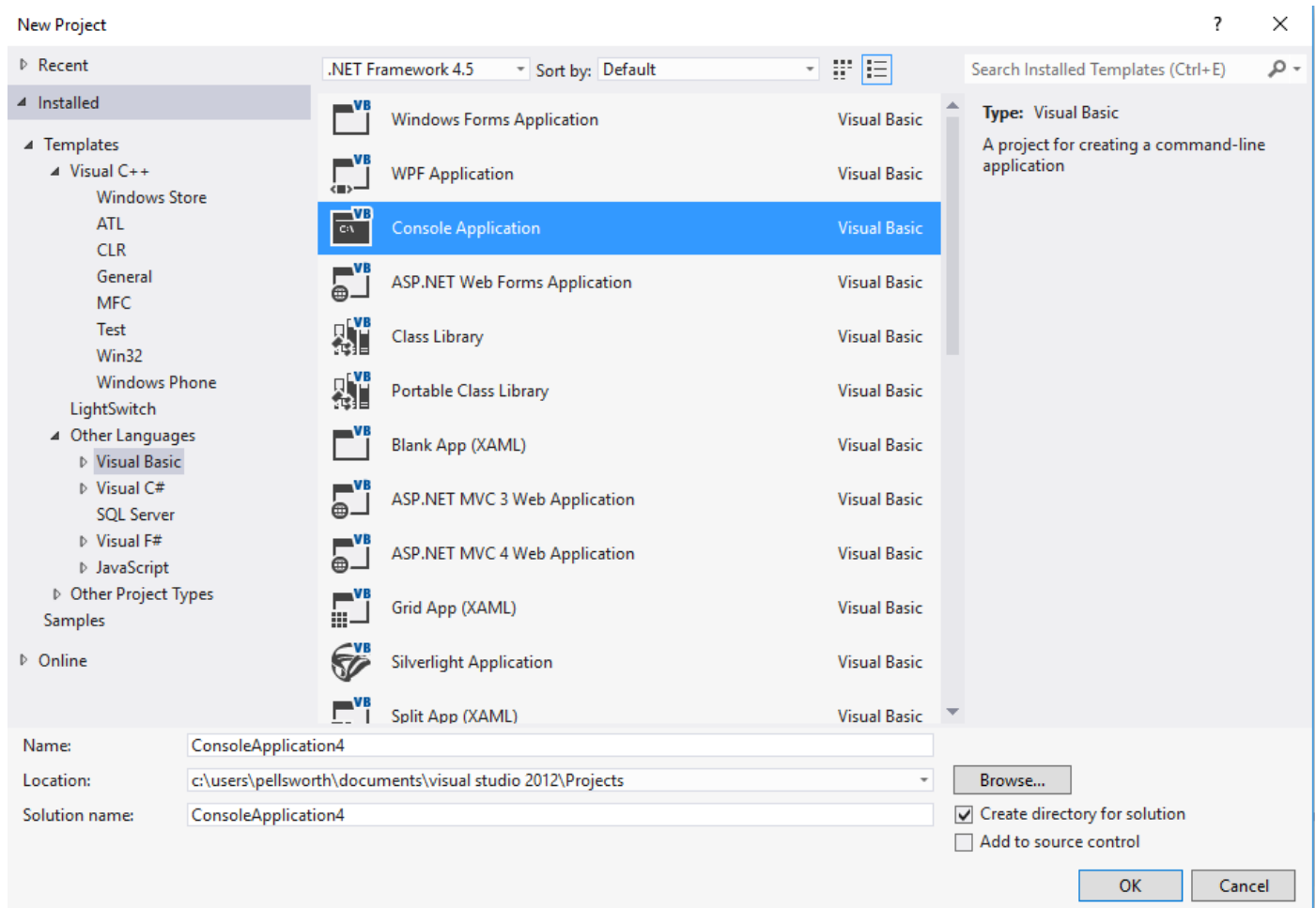
Objective: The student will be able to write a simple program to perform basic arithmetic operations.

Directions: Perform the following steps.

Step 1: Start Visual Studio.

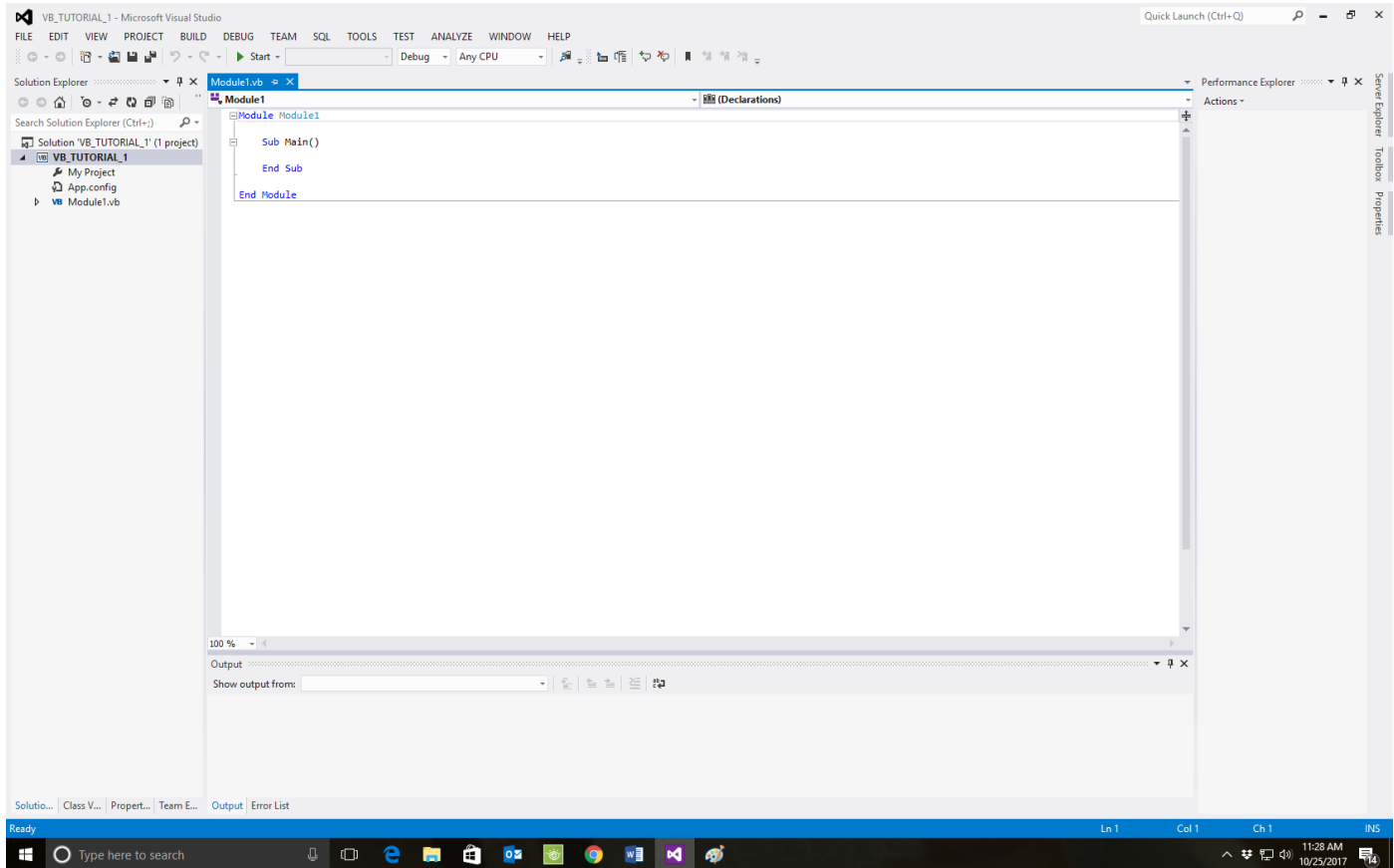
Step 2: Select Visual Basic.

Step 3: Select Empty Project.

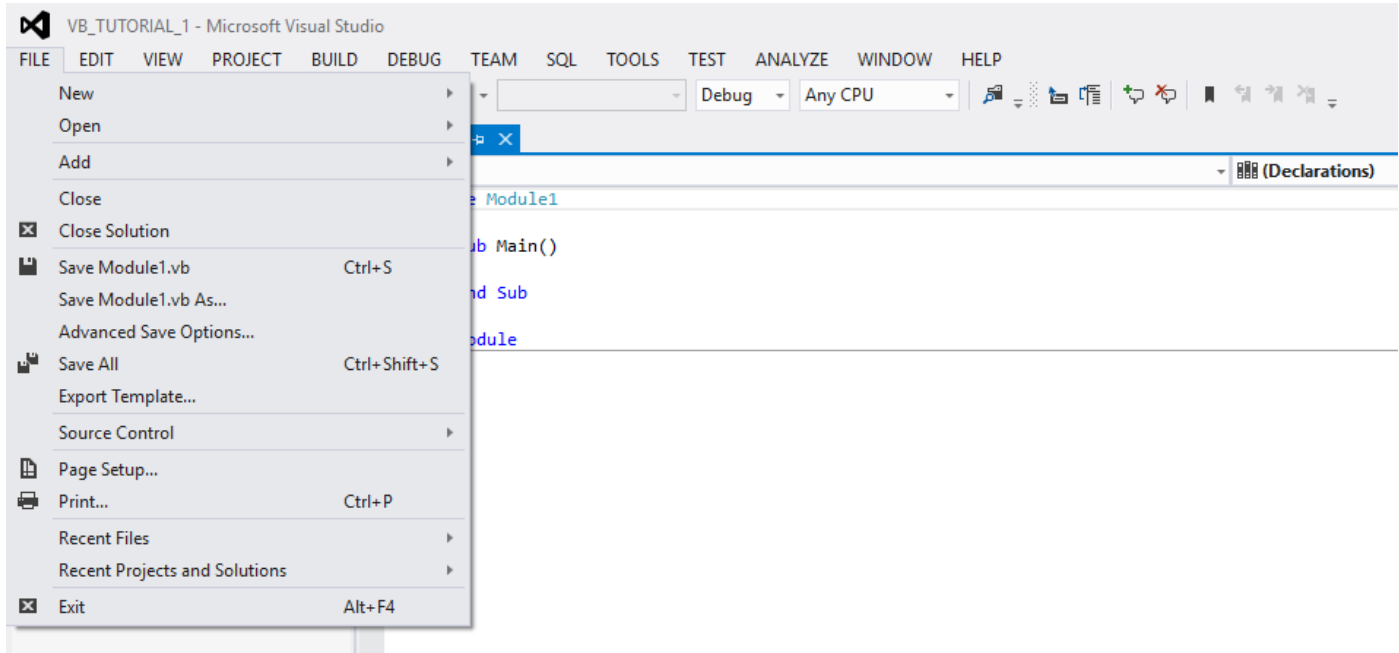


Step 4: Give the project a meaningful name.

Step 5: Click on OK.

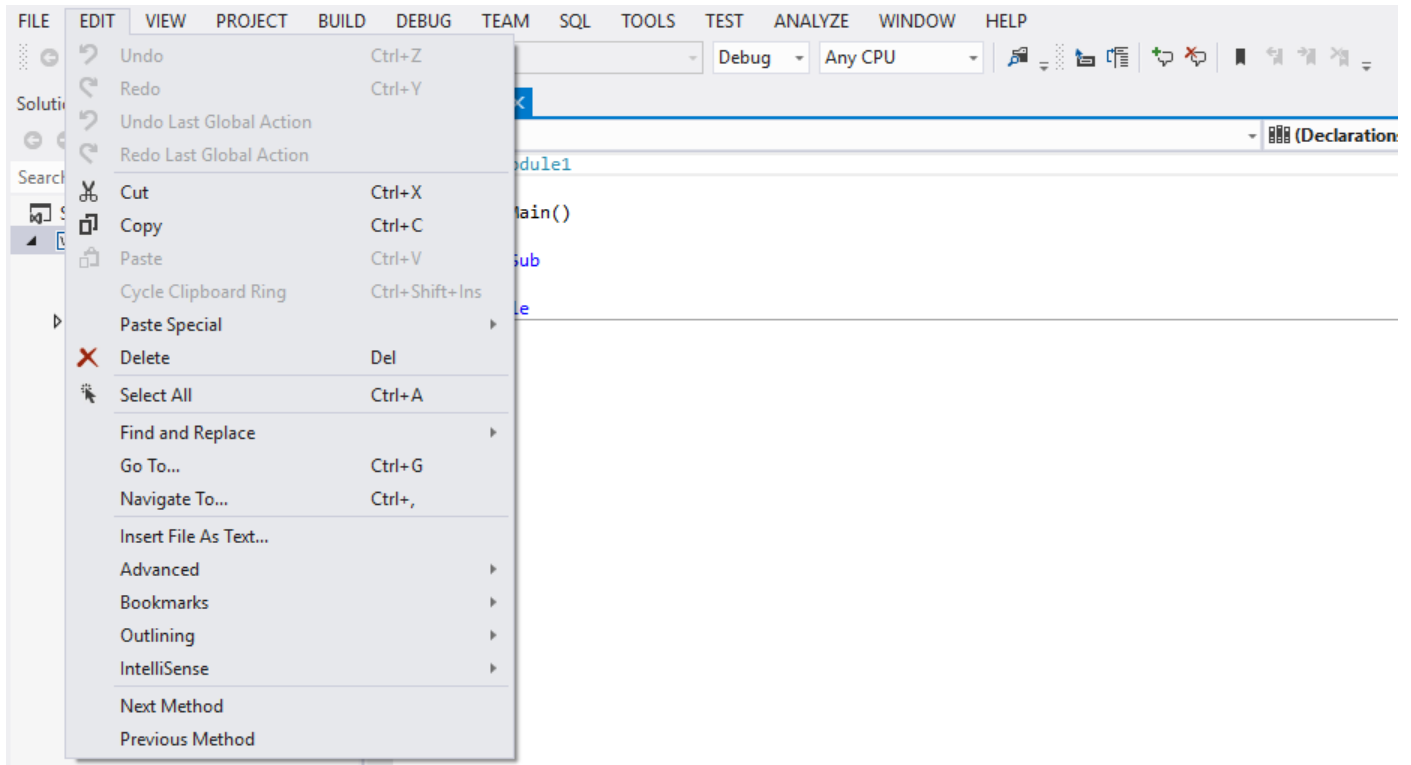


Step 6: This is the Visual Studio Integrated Development Environment or IDE for short. This is the place where you will write and edit your programs. Let's take a closer look at the IDE. At the top of the screen is a task bar with several drop down tool selections.

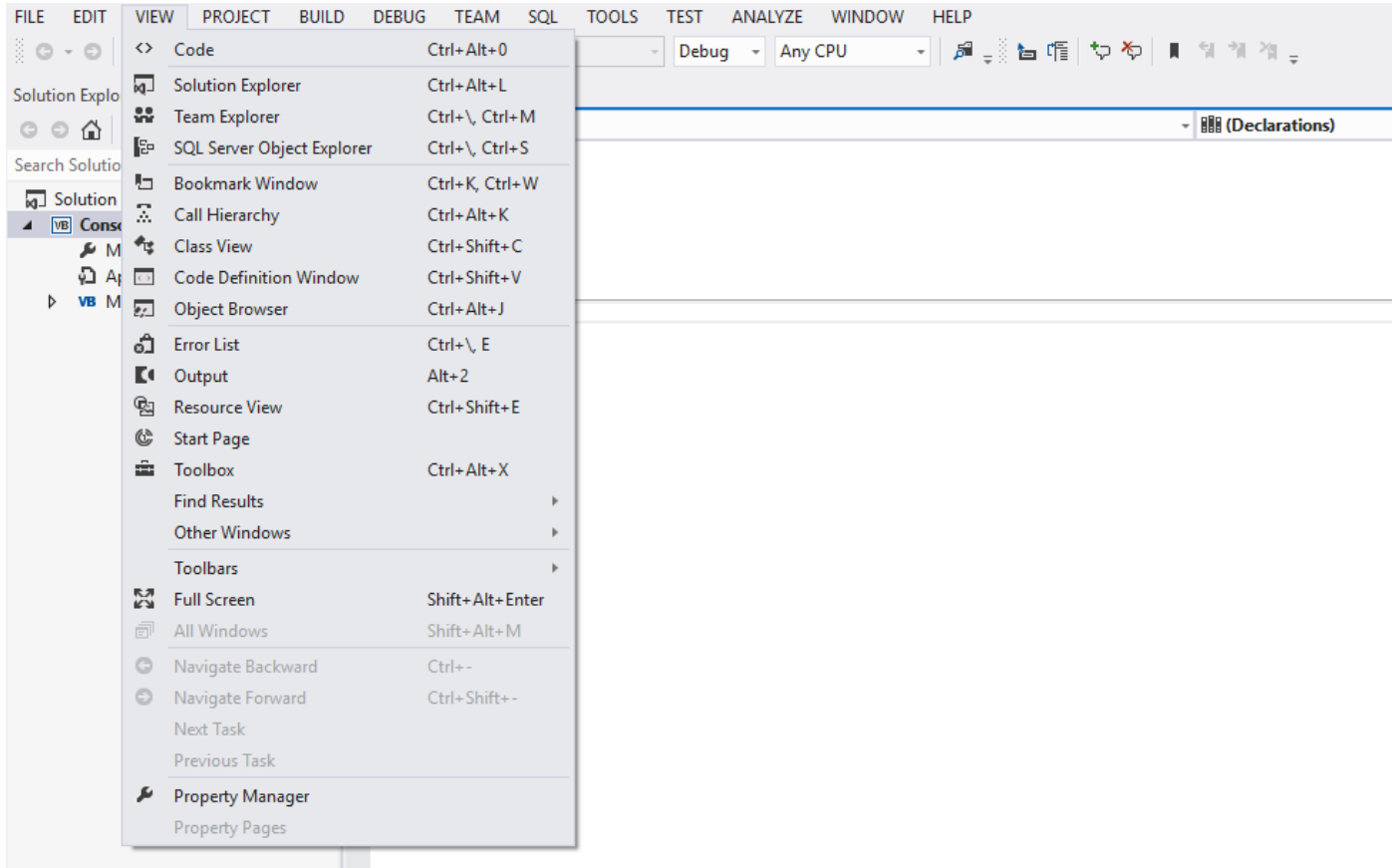


Step 7: This is the File tab drop down menu with several options that are similar to the standard Windows File Drop down menu. From here we can save our current project by selecting the Save All option (preferred choice).

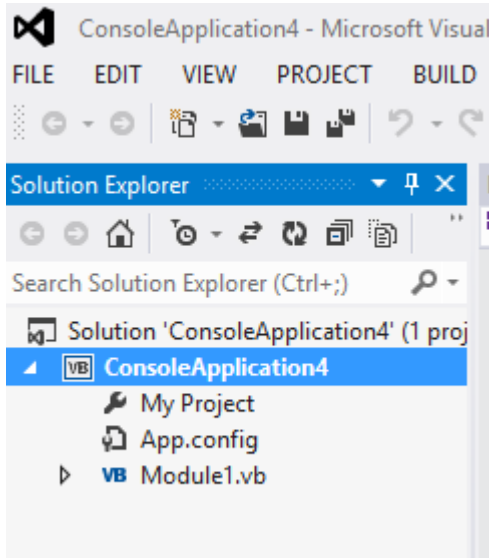
We also can start a new project by select the New option or open a previously saved project by selecting the Open option. We can also print the source code of our project by selecting the Print option.



Step 8: This is the Edit tab allows us to Copy blocks of code to paste into other parts of our program, which can be a huge time saver. There is also a Find and Replace function that we can use to quickly find every location of a variable in our program or where a method is located and used.



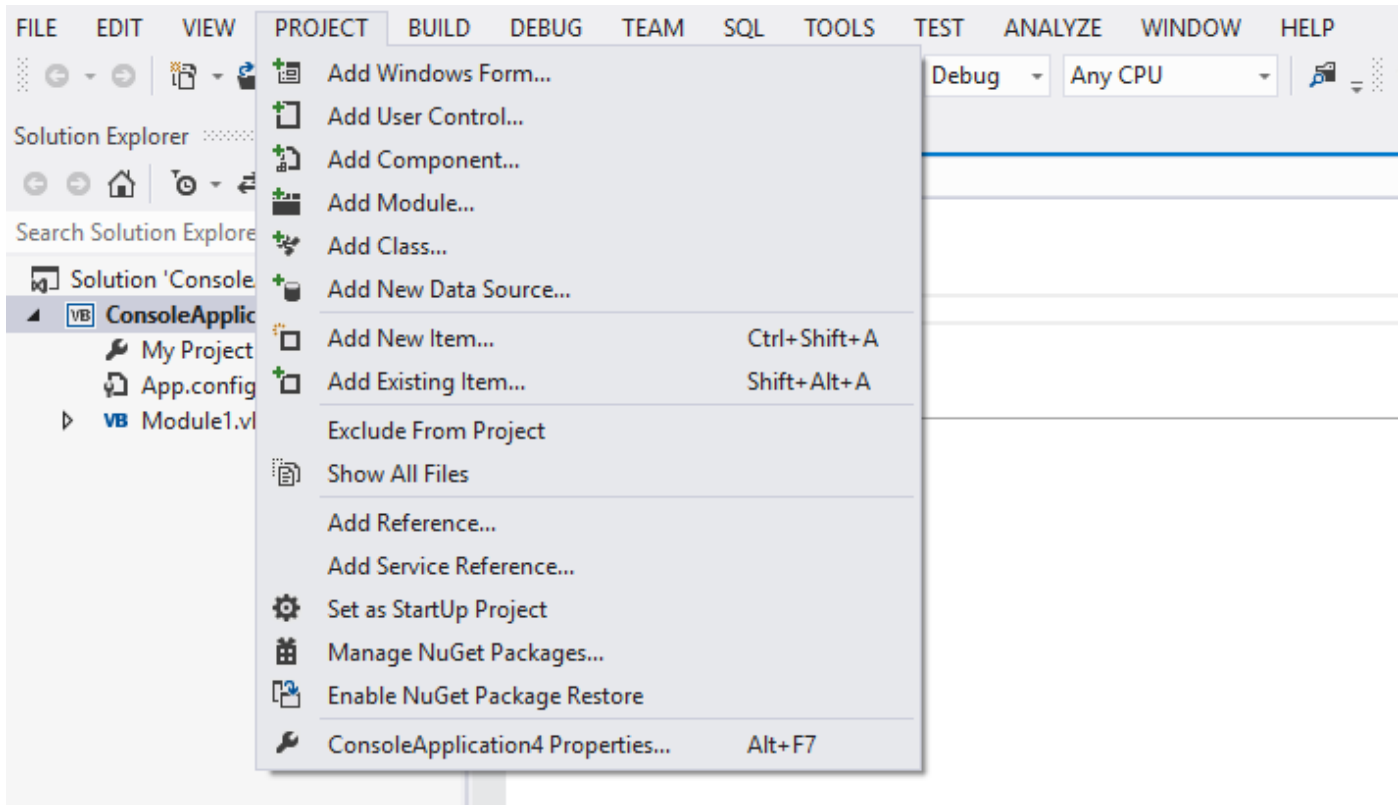
Step 9: This View tab is used to select many different windows such as the Solution Explorer. Although many of these options will not be used by us there are a few important ones. The Solution Explorer opens a small window on either the left side or the right side of the screen depending on the configuration setup. This window holds a hierarchy of our project. In a large project with many files it allows us to quickly navigate to the required file. Another important feature is that when nothing is displayed on the screen we can use the Solution Explorer to open the editing window to our program.



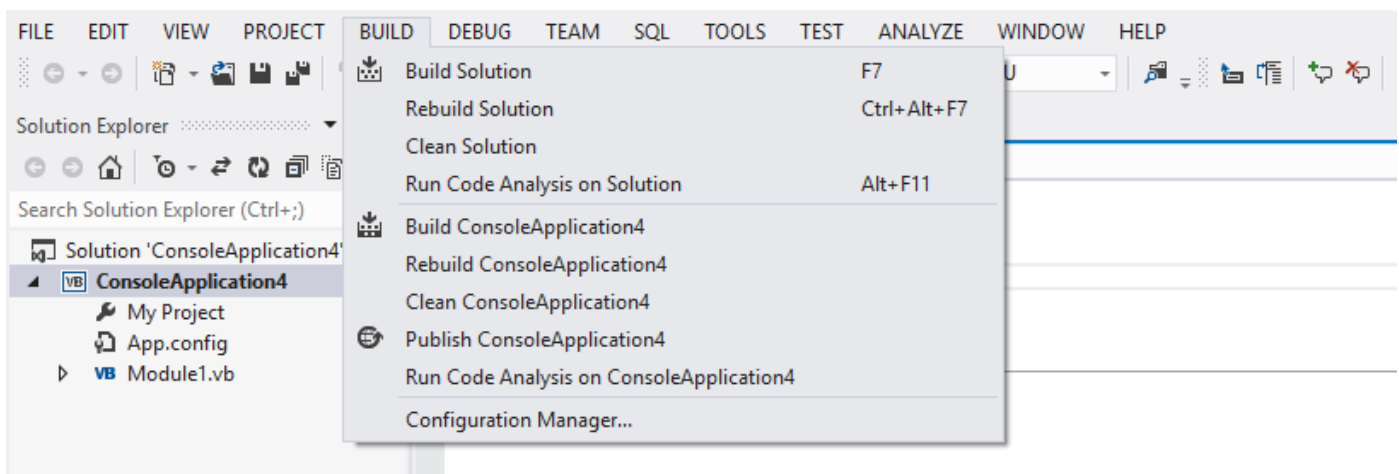
Step 10: From the View tab we can select the Error List. After we compile a program we can use the error list to quickly navigate to each of the errors in the program. This allows us to quickly correct the errors. The Error List usually opens at the bottom of the screen.

The Output selection will allow us to see the output of the program. It usually is a long window traversing the length of the screen and is located at the bottom of the screen.

There are many options on the View tab that are not important to us at this time since we are just beginning to learn about computer programming and they are helpful for much larger projects.

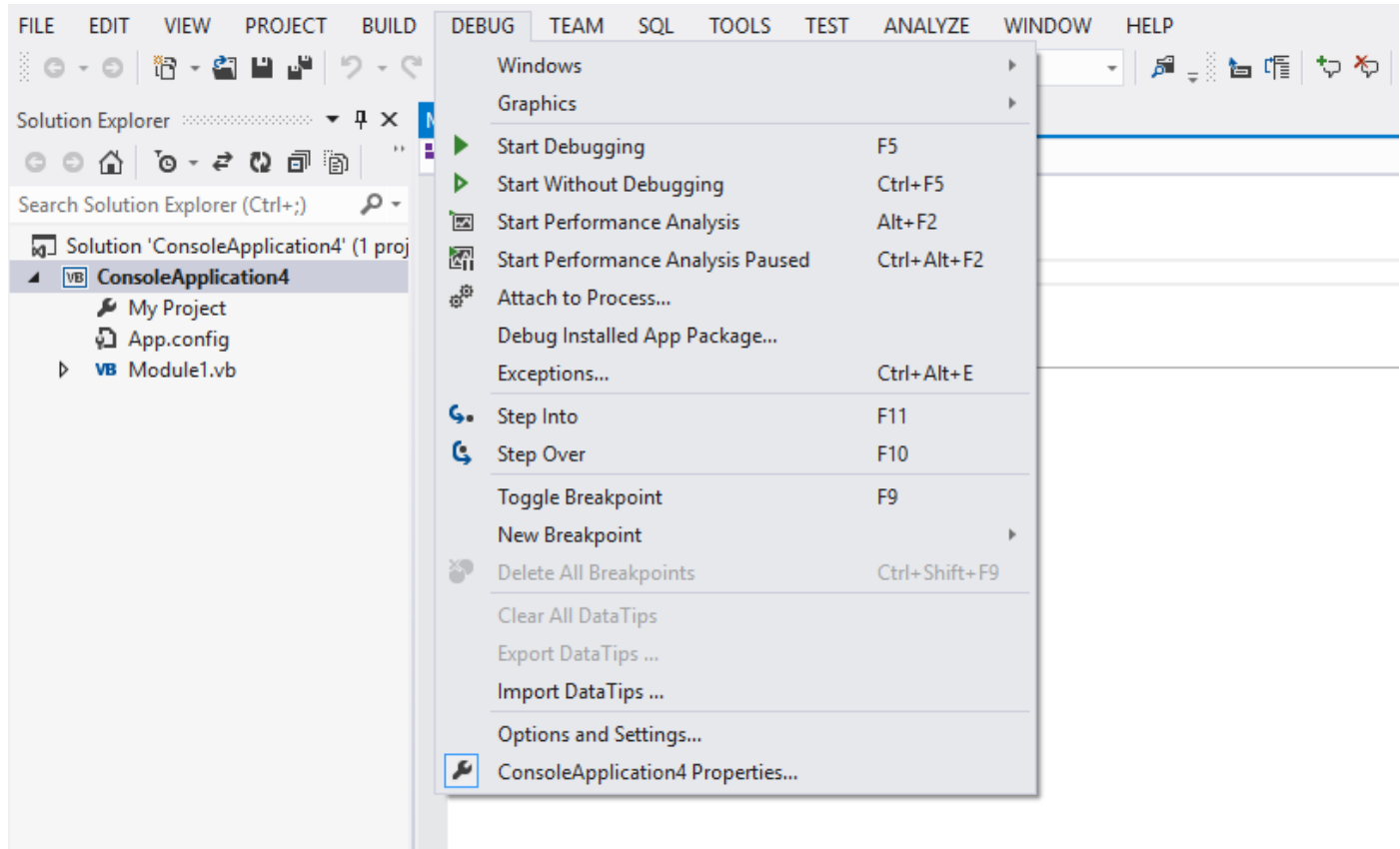


Step 11: The Project tab allow us to add files to our project. If the program, we are writing starts to become large and cumbersome we can add a new source file with the Add New Item selection so that we can break up the program into manageable pieces. On larger projects I will add a source file for Input Methods, Processing Methods, and Output Methods.



Step 12: The Build tab gives us the option to build or compile our program. The Rebuild option forces the program to be recompiled. In some special cases it may be necessary to force a rebuild to refresh the links to other modules associated with our program. This would only occur with much larger programming projects than we are going to do.

Most often we use the Cntr-F5 key to build and run the program together.



Step 13: The Debug tab gives us a variety of options for running our program. These can be very helpful for tracking down a difficult logical error. Most often we use the short cut Cntr-F5 to Start Without Debugging. This option compiles and executes the program. It also pauses the program just before the end of the program so that we can observe the program's output at our leisure.

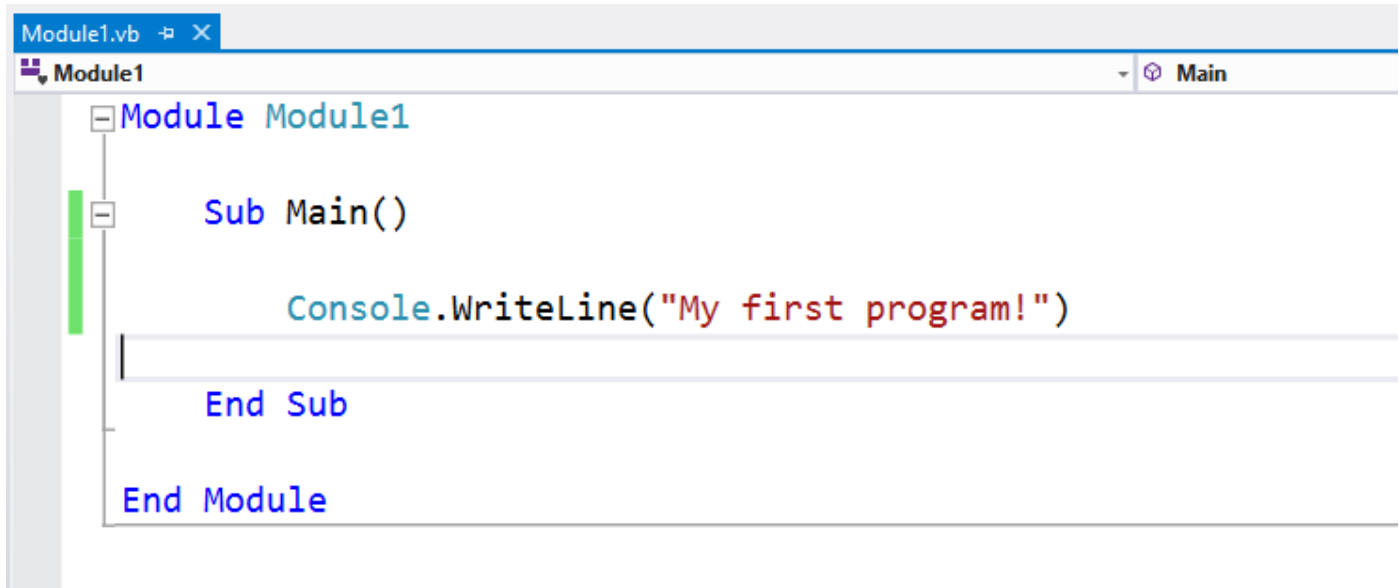
Step 14: The Start Debugging option allows us to step through the program one line at a time. This can be helpful if we have a problem that we cannot figure out. Seeing the variables change in a loop can often give us important insight to how the program is actually behaving.

Step 15: The Step Into option allows us to step into a method while the Step Over option allows us to skip over a method. The New Breakpoint option allows us to stop at a specific point in the program. This can be helpful on a long program where we don't need to do every single step.



For the most part we will not be using the remaining tabs. They are useful for more advanced types of programs.

Step 16: Now that we have some knowledge of the Integrated Development Environment let create a simple Hello World program. In the editing window enter the following program.

A screenshot of the Visual Studio IDE showing a code editor window for 'Module1.vb'. The editor displays a Visual Basic program template. The code is as follows:

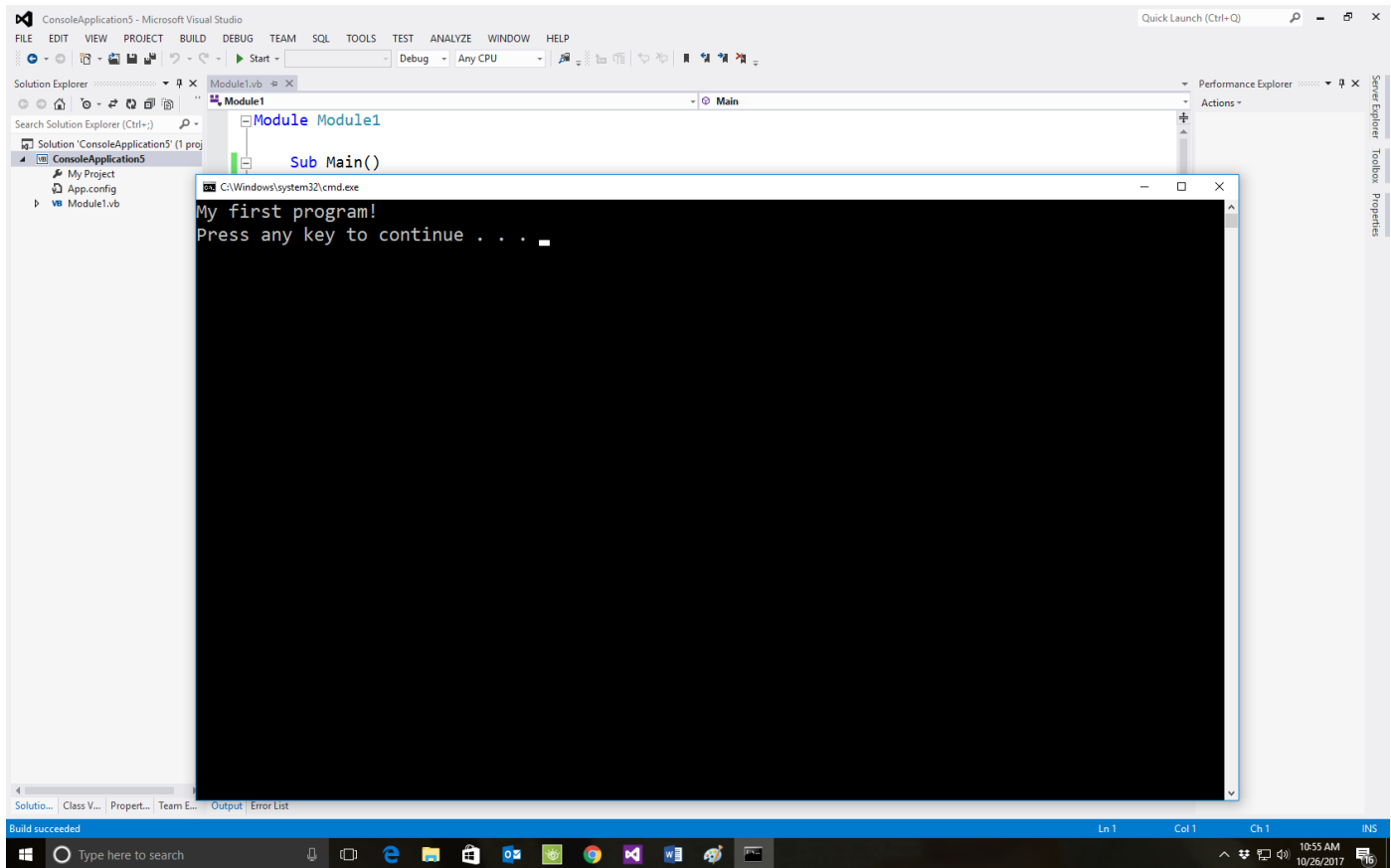
```
Module Module1
    Sub Main()
        Console.WriteLine("My first program!")
    End Sub
End Module
```

The interface includes a tab for 'Module1.vb', a 'Module1' container, and a 'Main' subprogram. A green vertical bar is visible on the left side of the code editor.

Step 17: Notice that Visual Studio created a starting program template for you. It consist of Module Module1, Sub Main(), End Sub, and End Module. Initially we will type our program between the key terms Sub Main() and End Sub.

Step 18: All Visual Basic programs must have a method named Main. This is the starting point of any Visual Basic program. Execution of the program starts at this point.

Step 19: After you enter the line: Console.WriteLine("My first program!") press and hold down the control key. Then press the F5 key. Release the Control Key and the F5 key at the same time. You should see the following:



Step 20: Congratulations, you have executed your first Visual Basic program. This concludes this presentation on the Visual Studio Integrated Development Environment.