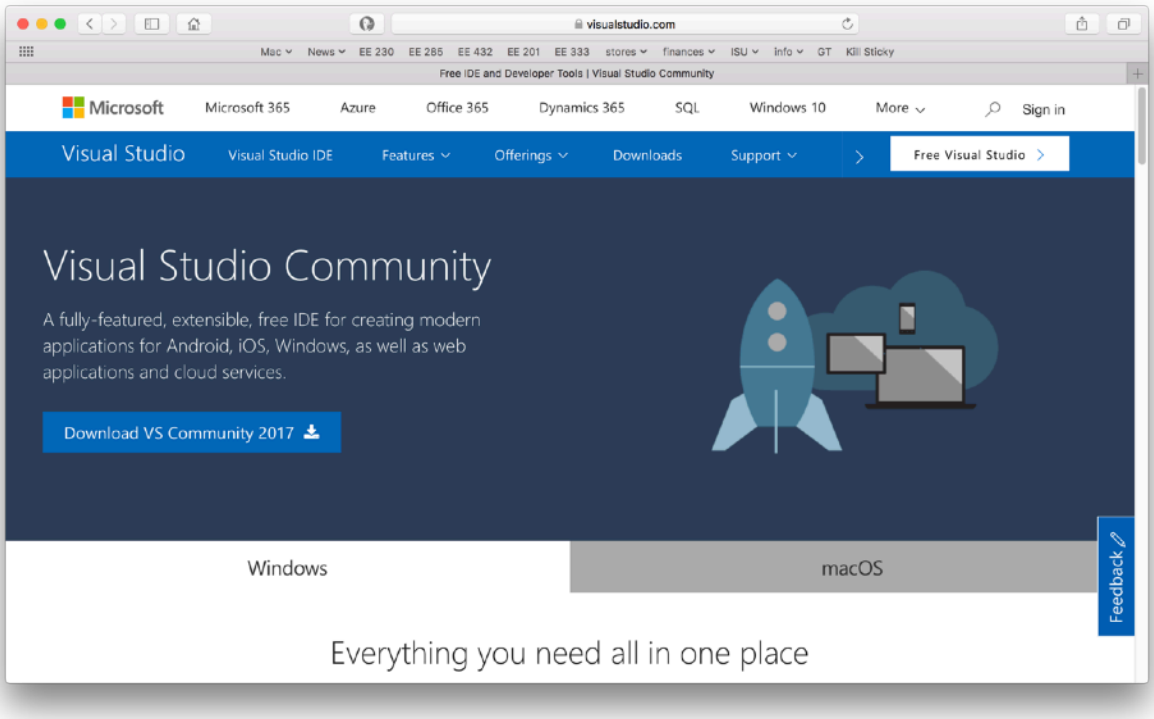
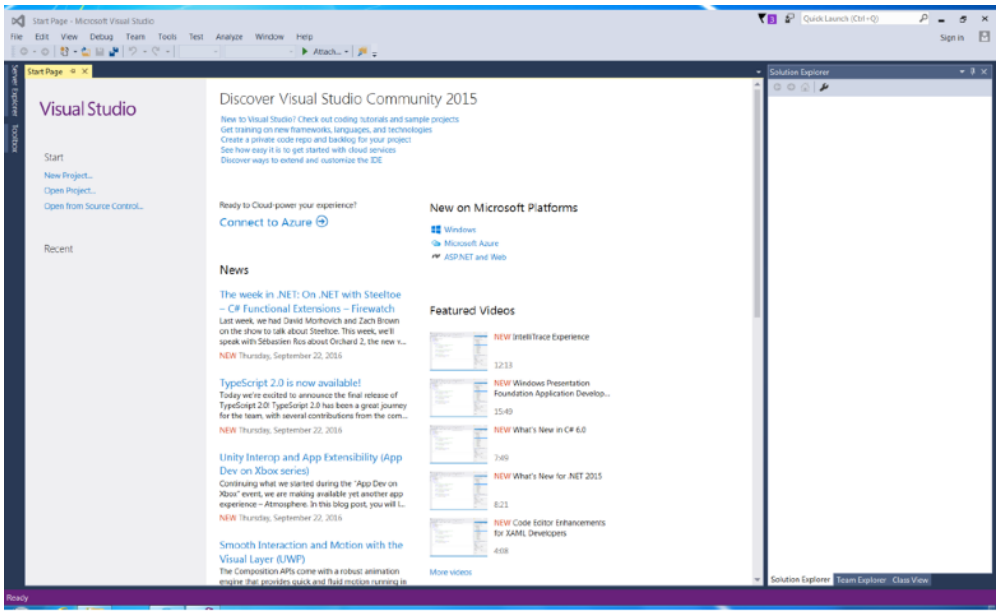
Installing and getting started with Visual Studio for C programming in Windows.

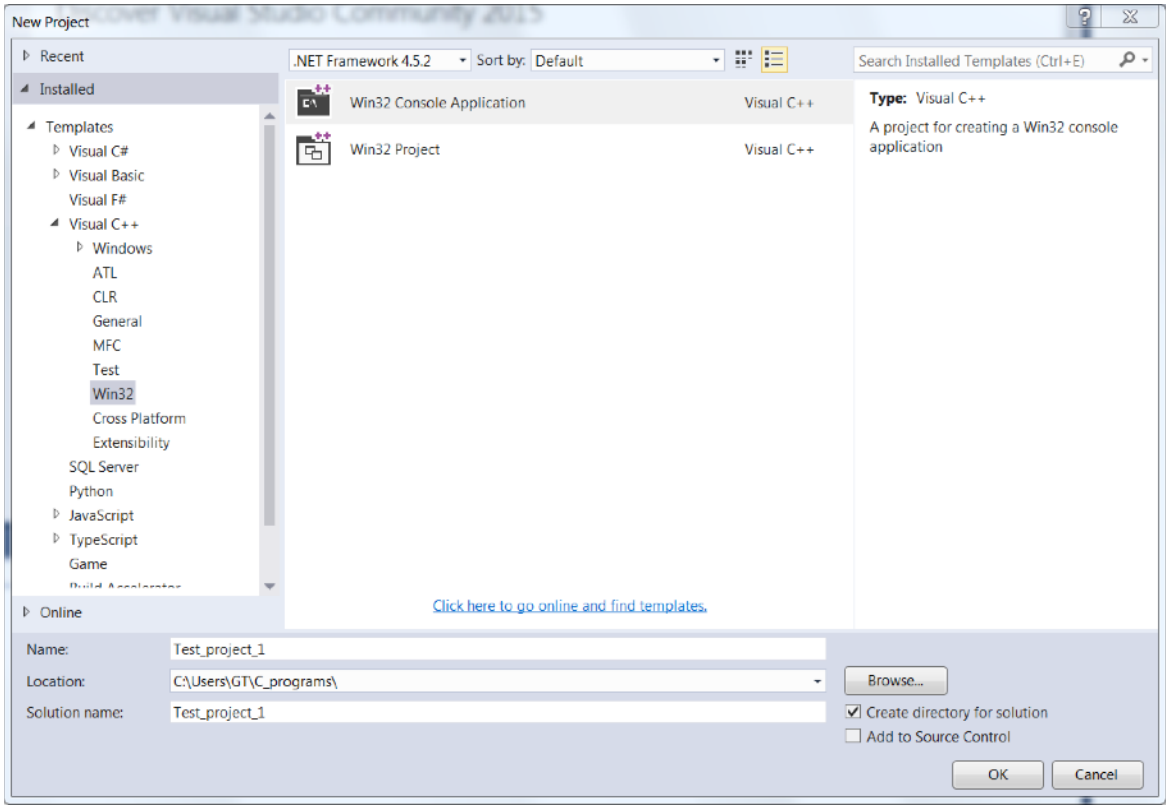
1. Download the free ("community") version VisualStudio tools. Go to https://www.visualstudio.com/vs/community/. Choose the Windows version (obviously). Download it and then install it. Be prepared to wait a bit. (It took me about 30 minutes to complete the process.)



2. Launch VisualStudio 2015. (Search for it in the start menu, if needed.) When it is launched, you will see the start screen.

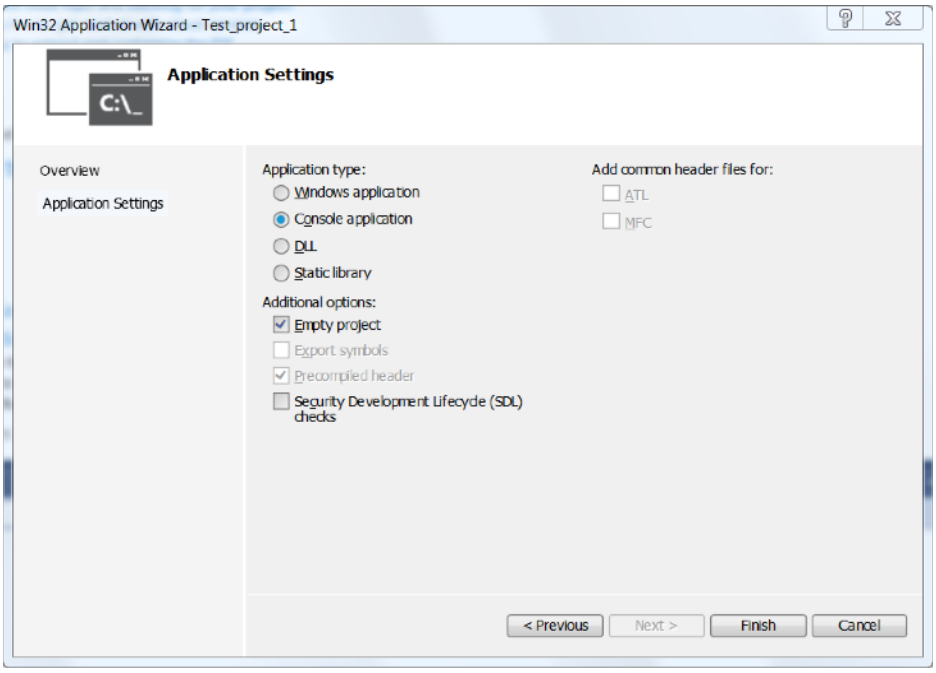


3. Click on "New Project..." in the upper left portion of the window. (Or choose that item from the "File" menu.) In the new project window that opens, choose Visual C++ --> Win32 from the options on the left and then choose "Win32 Console Application" from the two options in the center. Give the project a name (I used "Test_project_1".) and a location to save it. (I created a C_programs folder in the place where I keep most of my files on the C: drive. Click "OK" when everything is set the way you want it.

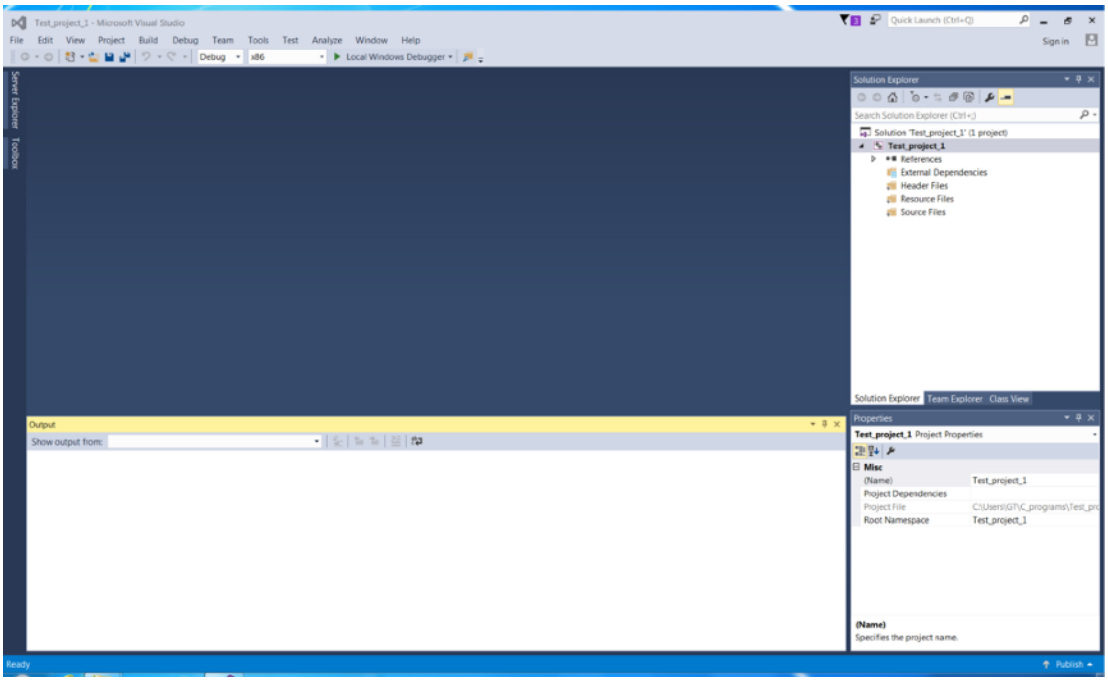


4. You are launched into a short setup wizard. You can click through the first screen.

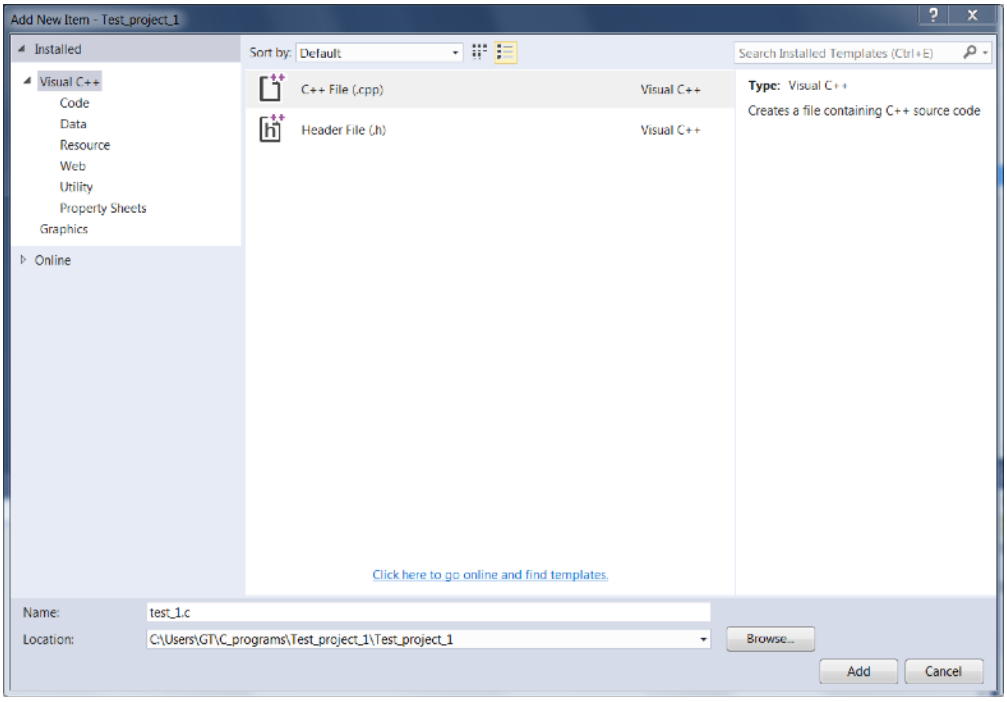
5. On the next screen, choose "Empty Project". You can uncheck "Security Development Lifecycle" if you want, although this is not important. Click “Finish".



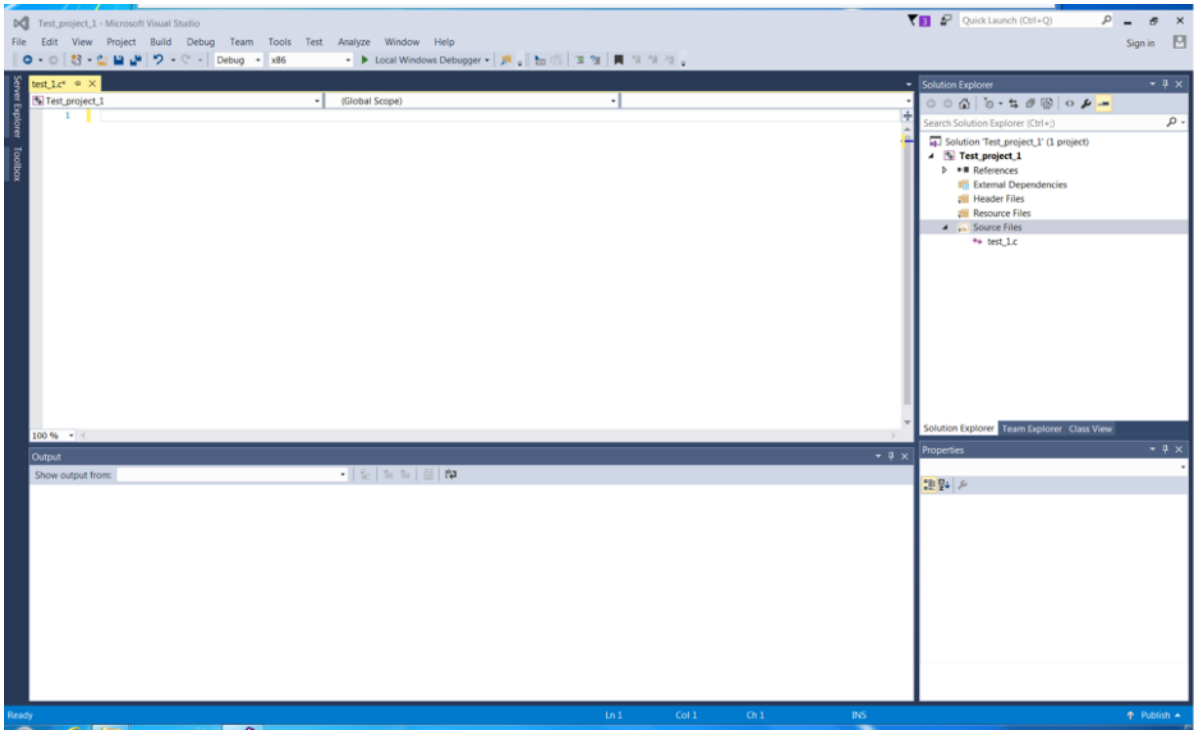
6. A blank project window opens.



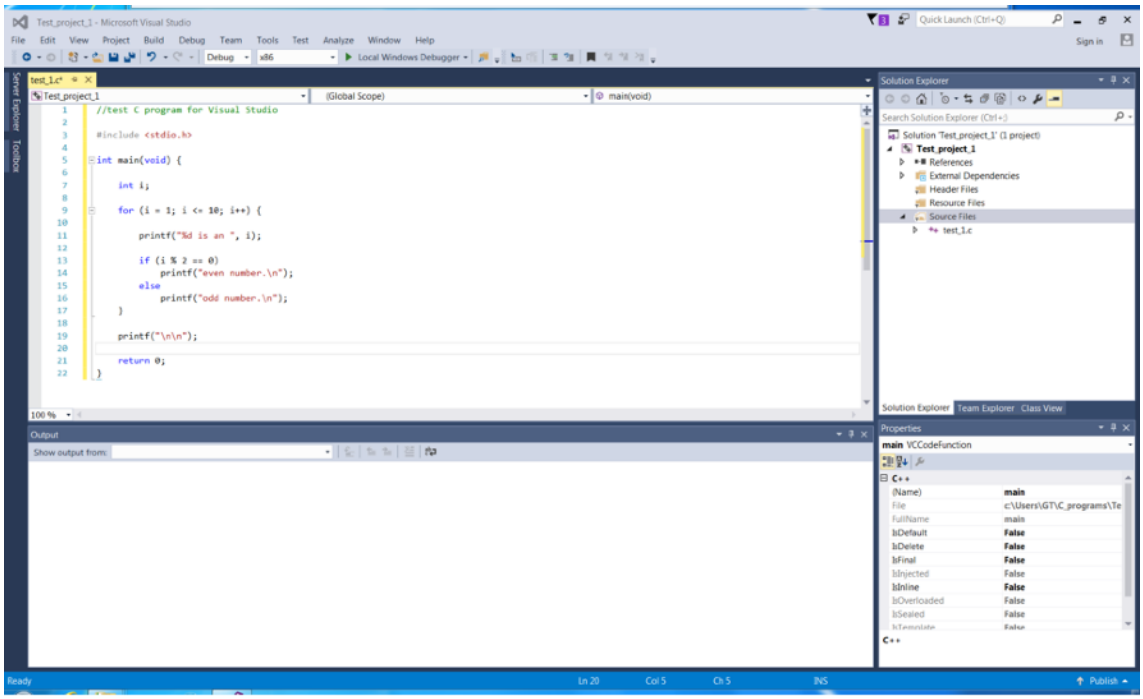
7. You need to add a text file for the code. To do that, right click on the "Source Files" list on the right, and choose the "Add-->New Item..." from the pop-up menu. In the dialog that opens, choose the "C++ File (.cpp)" item. Then enter a name for the text file that you are making. (I named it test_1.c, but the specific name is up to you.) Save the file where you would like. It makes sense to keep it in same location as the project, but that is not a requirement. Note that you must use the .c file designator. This tells the compiler that you are using "straight" C and not C ++.



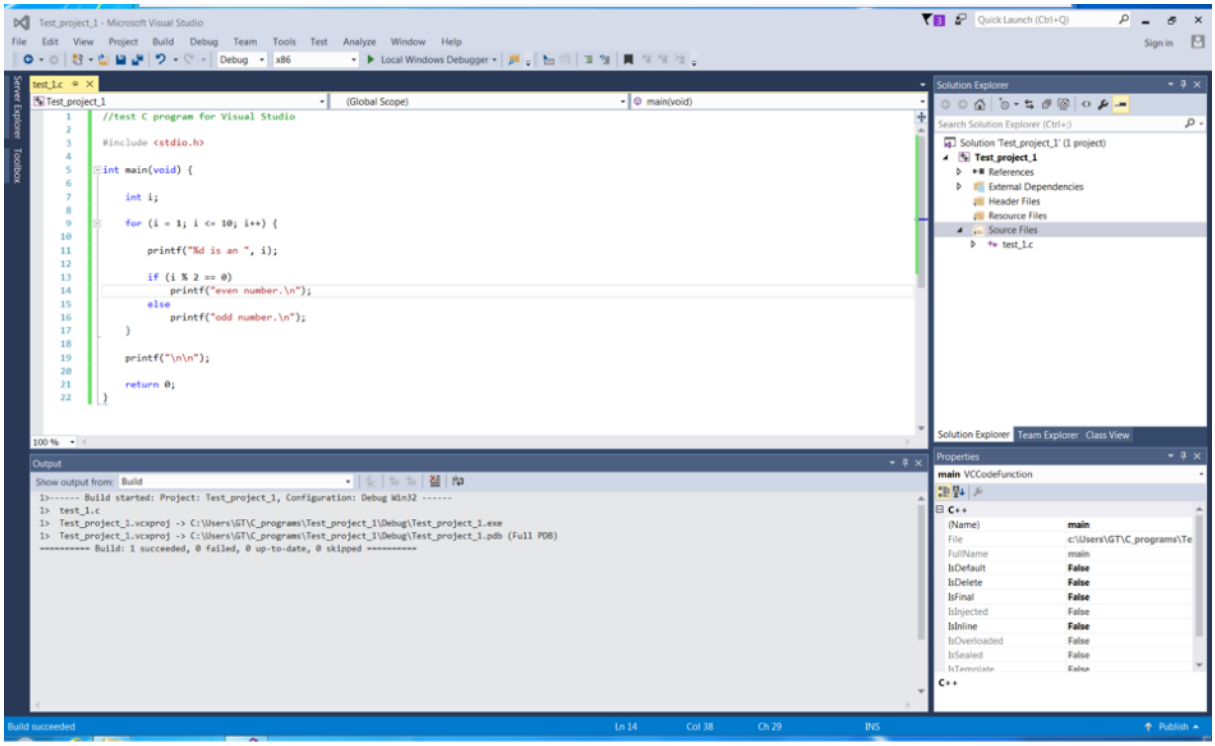
8. A blank text file opens.



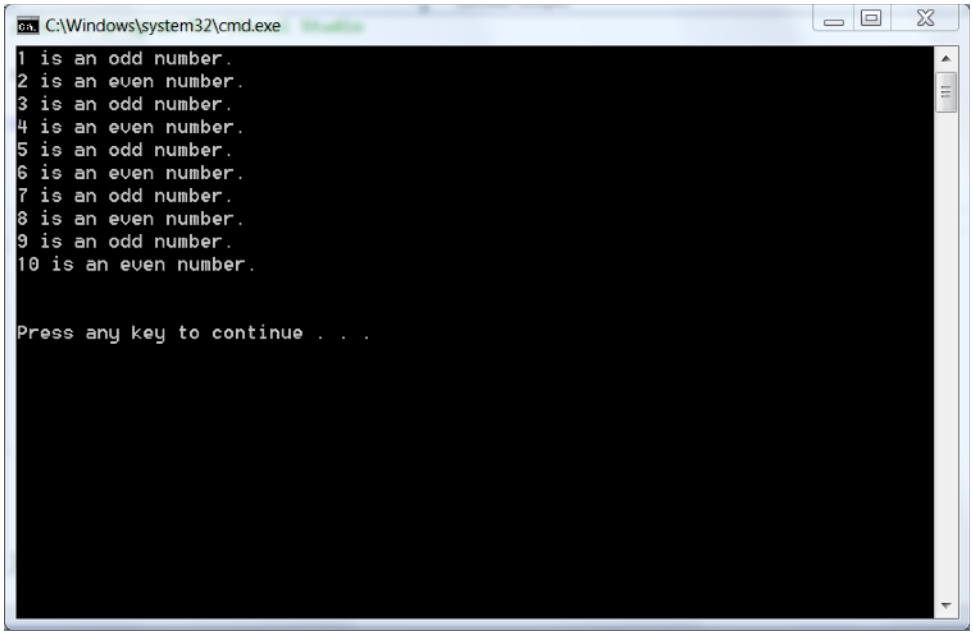
9. You can enter your code. Below is trivial bit of code to determine evens and odds.



10. When you have the code the way you want it, you can can compile it by choosing "Build Solution" from the "Build" menu. (Or use the Crtl-Shift-B key combination.) If there are any errors, they will be listed in an error window that appears below the code text file. If the codes compiles, there will be a confirmation message in the output window at the bottom.



11. Once the code builds correctly with no errors, then you can run it using the "Start Without Debugging" item in the "Debug" menu. (Or use Ctrl-F5.) The output will be printed to terminal-like window.



12. If you want save a hard copy of the output, you can use the Snipping tool that comes with Windows to grab an image of terminal. (Search for 'snip' from the start menu.) Suggestion: Use the rectangle or window options rather than grabbing a full screen shot.

To start a new project, begin at step 2 and enter the details of the new project. Carry through the remaining steps.

Using the VisualStudio "IDE" (integrated development environment) has some advantages beyond just not having to use a clunky Linux-like environment. One of the main things is the automatic formatting and colored keywords in the text editor. (All aspects of the formatting can be changed through the preferences. For now, you should probably use the default formatting. Once you have written enough code of your own to become fussy about how it looks, you can change the preferences.)

Also, if there are errors in compiling, Visual Studio gives much better feedback about the nature of the mistakes than you would get from a bare-bones Linux set up.

There is also an integrated debugger, which can help you find the errors in the operation of your code. As your programs get bigger and more complex, they may not run the way you think they should, even though all of the syntax is correct. The debugger helps you sort out problems by allowing you to stop and look at all of the variables at any point in time. This can be extremely useful, although we probably will not make much use of it in 285. But don't be afraid to try it out. With a bit of experimentation and some use of the help system, you can learn how to use the debugger and be able to make use of this very helpful tool.

There are other advantages to using an IDE, most of which are beyond our EE 285 capabilities. If you continue to program beyond this class (and you will), you may have opportunity to make use of other aspects of the development tool.