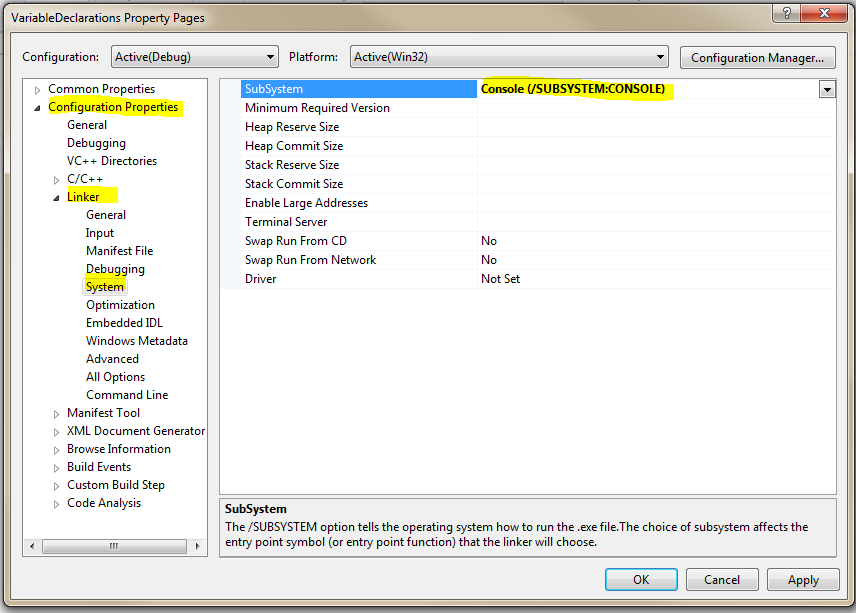
Vectors Demo

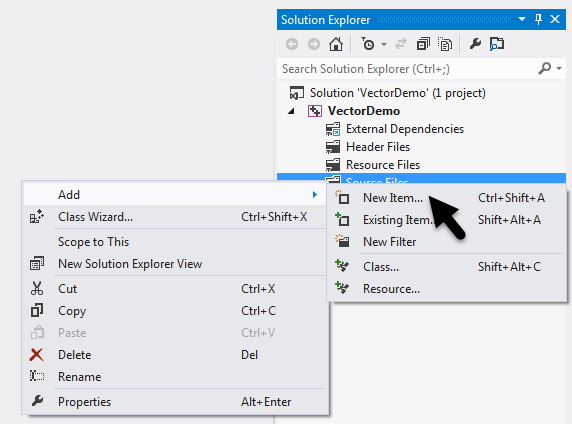
Video: <https://youtu.be/dxCjj-G1UuY?list=PLhoApZD2CmJZsWvHw3fmoj2Tx4tc8vD98v>

# Start our program

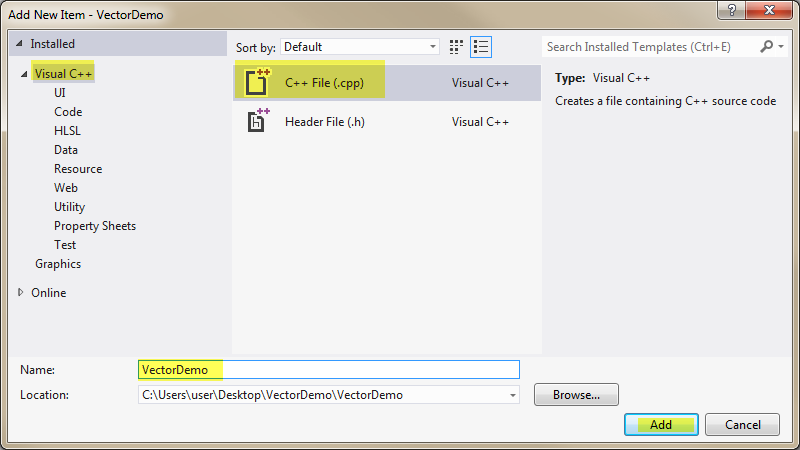
* 1. Start Visual Studio
  2. Create a new empty C++ project:
  3. Call it “VectorDemo”, pick the desktop as the location and click OK.
  4. Once your solution is created make sure your project is set up to run without closing the console window. Right click on the project (Called “VectorDemo”) and select “properties”.
  5. In the dialog window select Configuration Properties|Linker|System and set SubSystem to “Console(/SUBSYSTEM:CONSOLE)”:



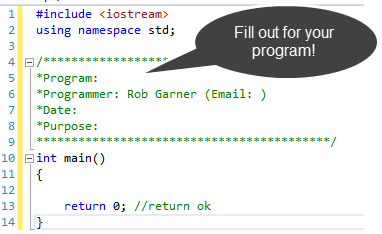
* 1. Click on Source Files and Add|New Item:



* 1. Select Visual C++, C++ File (.cpp), call it “VectorDemo”, then click “Add”:

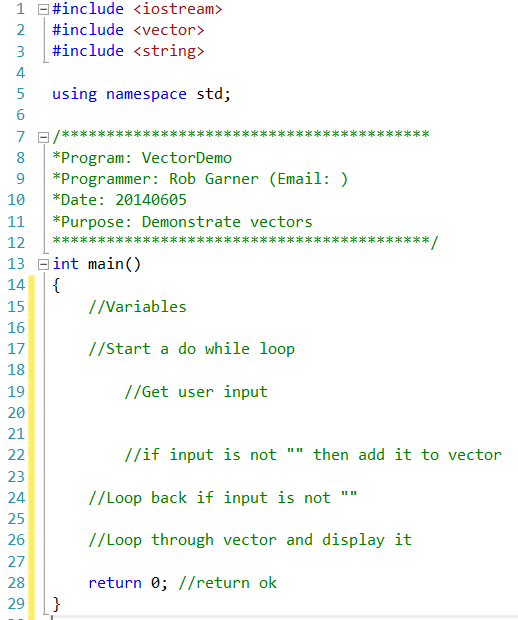


* 1. Enter the following code to start our program:

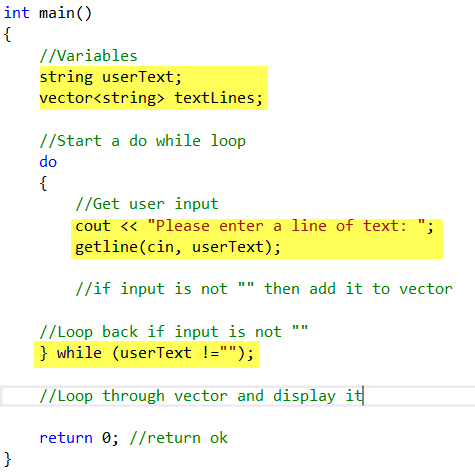


# Vectors

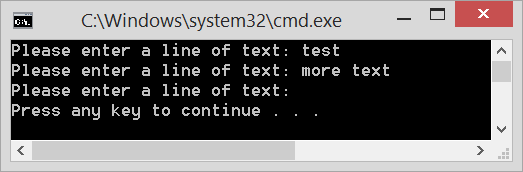
* 1. Add the following code:



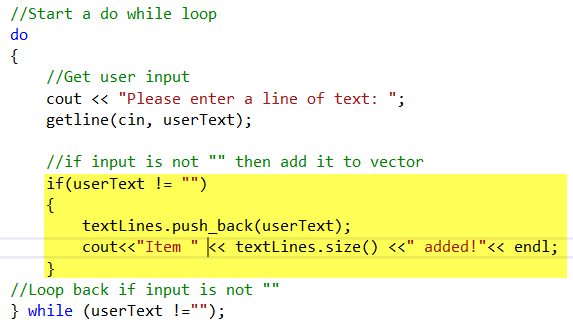
* 1. The comments is pseudo code. Notice that I have laid out my program before writing it. The advantage is that it allows me to think through what I am going to do and gives me an orderly way to attack the problem.
  2. Add the following code (in yellow):



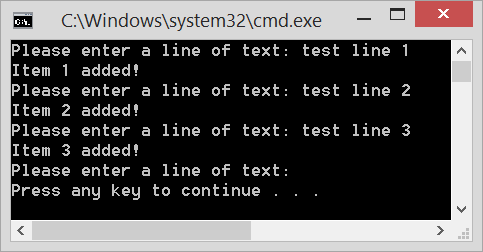
* 1. Notice I don’t try to solve the whole problem. Instead I try to get the loop to work first! By approaching the problem one piece at a time I make each section of code simple and easy to test.
  2. Try it with Ctrl-F5:



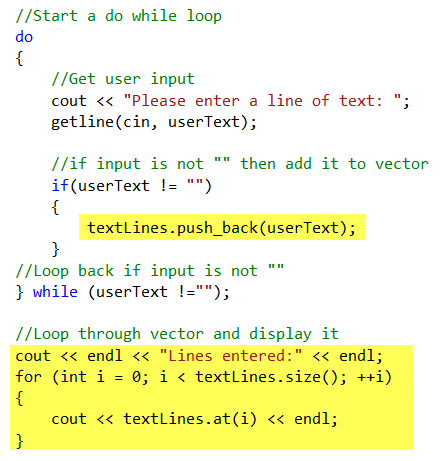
* 1. The loop works! Next lets work on adding items to the vector:
  2. Add the following code:



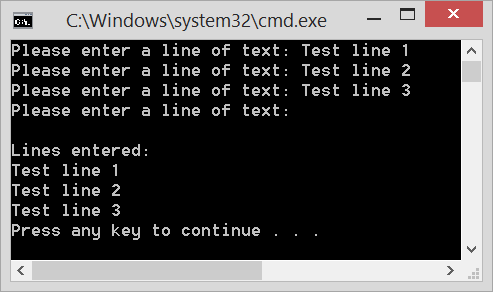
* 1. Notice that in addition to using the push\_back method to add the user input to textLines I also add a temporary cout statement displays the number of items in the vector using the .size() function so I can see that items are being added
  2. Try it with Ctrl-F5:



* 1. Notice that it works! We could keep the Item # added message, however, it doesn’t add a lot of utility to the program so we will delete it next step.
  2. Add code to display what was added. We will use a for loop:



* 1. Notice that our loop uses textLines.size() to get the size of the Vector. Also we use textLines.at(i) to get the element at the index i. Try it with Ctrl-F5:



* 1. Review the program and see how we used some of the following functions:



* 1. Notice how all of these functions are used by taking the name of the vector (in this case textLines) and add a dot then the name of the function. Example: textLines.pushBack(userText). This is object oriented programming. The functions in the table above are associated with variables declared as vectors. The functions above will work with vectors by using the dot operator but will not work with variables declared as other types.