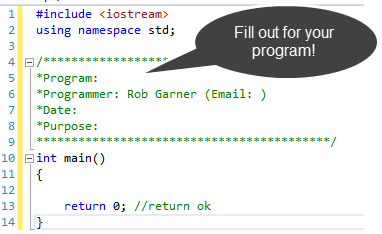
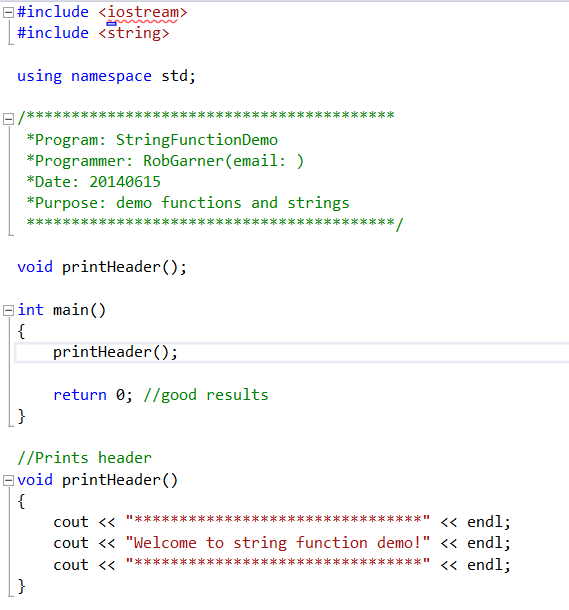
Functions Demo

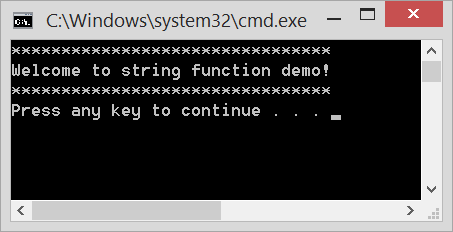
Video: <https://youtu.be/vb2K6GBAfho?list=PLhoApZD2CmJZsWvHw3fmoj2Tx4tc8vD98>

1. Start our program
   1. Start Visual Studio
   2. Create a new empty C++ project:
   3. Call it “DemoStringFunctions”, pick the desktop as the location and click OK.
   4. Click on Source Files and Add|New Item:
   5. Select Visual C++, C++ File (.cpp), call it “LoopDemo”, then click “Add”:
   6. Enter the following code to start our program:



1. Demo void parameterless function:
   1. Add the following code:

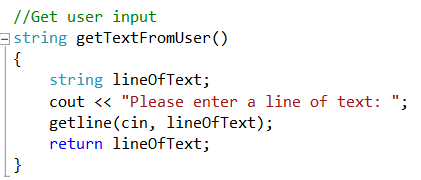


* 1. Notice that we have defined a function prototype above main, a simple function below main and call it within main.
  2. Try it with Ctrl-F5
  3. 

1. Demo function that returns a value but takes no parameters.
   1. Add another function:
   2. Add function prototype above main:



* 1. Add function definition below main:



* 1. Add function call in main:

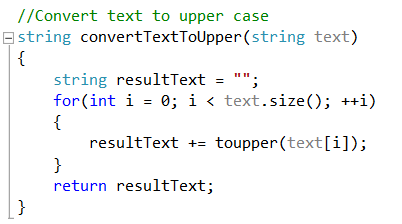


* 1. Try it with Ctrl-F5

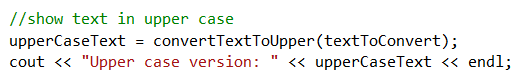
1. Demo function that takes a parameter and returns a value:
   1. Add another function:
   2. Add function prototype above main:



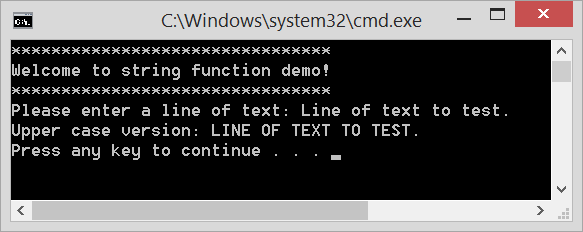
* 1. Add function definition below main:



* 1. Add function call in main:



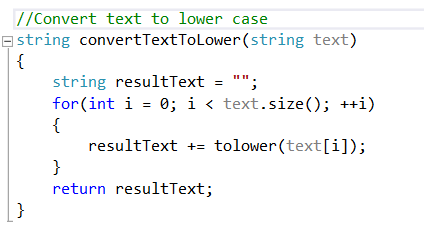
* 1. Try it with Ctrl-F5



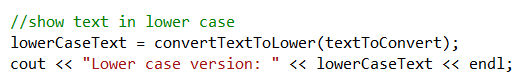
1. Demo another function that takes a parameter and returns a value:
   1. Add another function:
   2. Add function prototype above main:



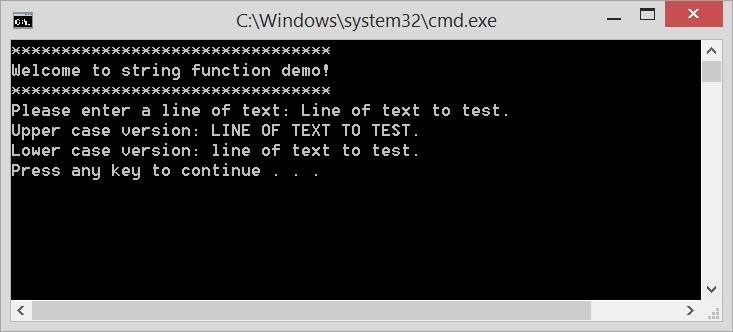
* 1. Add function definition below main:



* 1. Add function call in main:



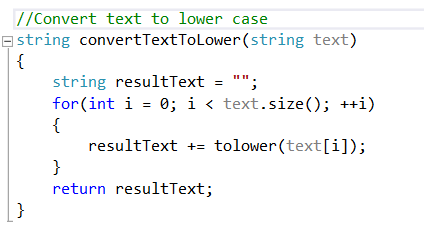
* 1. Try it with Ctrl-F5



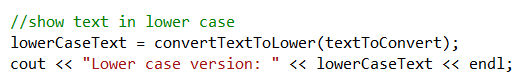
1. Demo another function that takes a parameter and returns a value:
   1. Add another function:
   2. Add function prototype above main:



* 1. Add function definition below main:

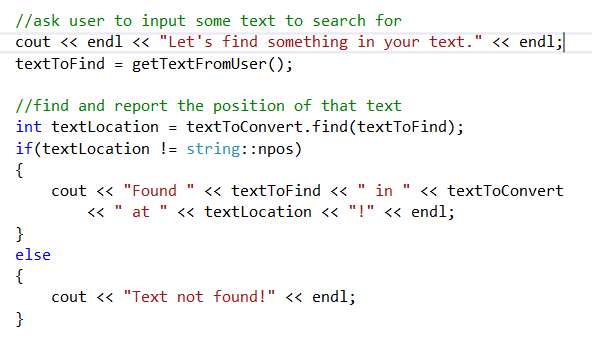


* 1. Add function call in main:



* 1. Try it with Ctrl-F5

1. Demo calling a string function
   1. Add the following code to main:



* 1. Notice that we reused the getTextFromUser function to get another line of text from the user.
  2. Notice that we use the string find function to get the location of that texttoFind n textToConvert.
  3. We check to see if textLocation equals string::npos. If it is not equal that means textLocation has an integer value that represents the location of the text we are searching for.
  4. If textLocation equals string::npos that means the text was not found, so we display a text not found message.