

# CS150 Assignment 7

## Grader with Functions!

**Date assigned:** Monday, November 15, 2010

**Date Due:** Monday, November 22, 2010, 1pm (30 points)

**Total points:** 30

### **Problem**

You need to take your quiz grader from Assignment 6 and rewrite the code to take advantage of functions. The input and output should be exactly the same as in Assignment 6. Test cases are posted to allow you to debug your solution to Assignment 6.

► You are required to use at least **three meaningful functions** (not counting main()). Look for instances of repeated code in your project to turn into functions. A meaningful function is called from at least two locations in the source code.

► Debug your code first before starting on this assignment!

### **To complete this assignment you must**

1. Create a new C++ project in Visual Studio. Name your project **07Graderxxxxxxx**, where xxxxxxxx must be replaced by your PUNetID. As an example, my project would be called "07Graderkhoz0332". It is vital that you name your project correctly!
2. Type the solution (**fully documented/commented**) to the problem into your project.
3. Remember to enter in your name as the author of the program.
4. Make sure that your program compiles and runs correctly. If you get any errors, double check that you typed everything correctly. Be aware that C++ is case-sensitive. Also, there must not be any warnings when compiling your program.
5. Once you are sure that the program works correctly, it is time to submit your program. You do this by logging on to Turing and placing your complete project folder in the **CS150-01 Drop** folder. Make sure that you copy your program folder and don't move the folder. If you move the folder, then you will not have your own copy!

### **Additional Notes**

1. You must follow the coding standards. Refer to the lecture notes on how to comment a function.
2. You must use constants when possible.
3. Your program will be graded on **efficiency**. In other words, you will be marked down for repeating code statements unnecessarily.
4. Your output must look **exactly** like the sample given.
5. Write and debug this project one function at a time!
6. You can use the **debugger** to set breakpoints in functions.
7. You can use the debugger to step into a function

and step out of a function.



**Remember, this is an individual assignment.** Refer to the syllabus for assignment policies