

## User-defined Functions:

## Simple Invoice

For this project, you will restructure your solution to the previous project by take advantage of the ability to write user-defined functions in C++. Since the functionality of this program is identical to that of the previous program, although you will submit your solution for testing on the Curator, no credit will be given unless your solution satisfies the specific requirements stated below.

It is now an explicit requirement that your program store the price information using `int` variables, rather than variables of type `double` or `float`.

### Specific requirements for functions:

Your solution must include at least five different user-defined functions, not counting `main()`. For reference, my version of the solution includes nine user-defined functions. You must implement functions that satisfy the following requirements:

- a function that reads all of the customer's address data from the input file, and communicates that data back to `main()` by using reference parameters.
- a function that writes the customer's address data to the output file.
- a function that computes a monetary amount and returns it via a `return` statement.
- a function that reads an integer value from the input file and returns it via a `return` statement.
- a function that writes a line of the table in the invoice file.

Your solution must absolutely not use any global variables! It is perfectly OK to use global constants and to make function declarations global. You must use the appropriate protocols when you pass parameters to your functions. Pass parameters by reference only when it is logically necessary; otherwise, pass parameters by value or by constant reference.

### Documentation requirements:

Since your solution must be checked to verify the requirements above, we will also check to see if you have included good comments. Read the Programming Standards page on the course website for guidelines. In particular, you must include a header comment for each of your user-defined functions that include the information shown in the sample function header comment.

### Submitting your program:

You will submit this assignment to the Curator System (read the *Student Guide*), and it will be graded automatically. Instructions for submitting, and a description of how the grading is done, are contained in the *Student Guide*.

You will be allowed up to ten submissions for this assignment. Use them wisely. Test your program thoroughly before submitting it. Make sure that your program produces correct results for every sample input file posted on the course website. If you do not get a perfect score, analyze the problem carefully and test your fix with the input file returned as part of the Curator e-mail message, before submitting again. The highest score you achieve will be counted.

The *Student Guide* and other pertinent information, such as the link to the proper submit page, can be found at:

<http://www.cs.vt.edu/curator/>

**Pledge:**

Each of your program submissions must be pledged to conform to the Honor Code requirements for this course. Specifically, you **must** include the following pledge statement in the header comment for your program:

```
// On my honor:  
//  
// - I have not discussed the C++ language code in my program with  
// anyone other than my instructor or the teaching assistants  
// assigned to this course.  
//  
// - I have not used C++ language code obtained from another student,  
// or any other unauthorized source, either modified or unmodified.  
//  
// - If any C++ language code or documentation used in my program  
// was obtained from another source, such as a text book or course  
// notes, that has been clearly noted with a proper citation in  
// the comments of my program.  
//  
// - I have not designed this program in such a way as to defeat or  
// interfere with the normal operation of the Curator System.  
//  
// <Student Name>
```

**Failure to include this pledge in a submission is a violation of the Honor Code.**