PROG 111 | Project #8 | Date Assigned: 14 November 2012
40 points | Chapters 7 and 8 Material | Date Due: 22 November 2012

General Instructions:

This is an open book project. You may receive assistance or help from anyone, including me, regarding the ***concepts only***. Therefore, you must do your own work on the program solution and not utilize anyone else with your code development. If you are approached by another student to supply him or her with your solution, you must decline their request. When you have finished, you must zip your solution into a Zip folder that includes the files described below and upload the folder via the Canvas upload feature.

Do not wait until the last day to start your work!

You are to design a class that will represent a restaurant order. The class will have two public methods, one to place an order and the second to calculate and display the order items and prices, the order sub-total, the tip, the tax, and the check total. The method to place the order will allow the User to enter *up to* 5 order items. The method is to present a Menu of items for selection. You are to create two *const static*, parallel arrays for the menu items, one for the item descriptions (*string*) and the second for the item costs (*double*), which, of course, will be class private data members. You are to provide 10 items for selection. You might find a restaurant menu online and to use for your 10 items. You will store the items selected in another one-dimensional array; the item numbers corresponding to the element in the two static arrays. You are to use a "sentinel" value to terminate the items entry OR terminate entry after 5 items have been entered. You are to clear the Console screen (see page 337 in the text) and present the User with a Menu of items and costs (from the two parallel arrays), two items per line. Each item shall have an integer selection number that the User will use to select the item, as an example:

1: Breakfast roll $1.75.

Display a prompt following the Menu presentation to enter the next item, indicating the order item number (1 – 5, maximum). You can use the Screen Control features described on page 464 of the text to place the cursor at the same position for each item selection prompt message. I will publish an example program with a function that will allow you to determine the current position of the Console cursor.

The second, public class method, which will display the customer's check will sub-total the cost of the items ordered, calculate a tip based on the sub-total amount then calculate the tax, again based on the sub-total amount, then calculate the check total as the sum of the order subtotal, the tax, and the tip. The method will display the entire order (each item description and cost), the tax, the tip, and the check total. If no items were ordered, do not display a check, but display a message indicating the order was cancelled.

Your project ***must*** meet these minimum requirements:

- Name the class: *RestaurantCheck*
- Include a Constructor that takes two arguments, the tax rate and the tip, both as a percentage:
    1. Validate that the tax rate is greater than or equal to 1% (0.01) and less than or equal to 12% (0.12). If the value passed to the Constructor is invalid, set the tax rate to 0.065
    2. Validate that the tip percentage is greater than or equal to 5% (0.05) and less than or equal to 20% (0.20). If the value passed to the Constructor is invalid, set the tip to 15% (0.15)
- Include a Default Constructor that sets the tax rate to 0.065 (6.5%) and the tip percentage to 15%
- The default values and the limit values should be declared as class *private*, *static const* values
- Name the public class method to place an order: *placeOrder*
- Name the public class method to display the check: *issueCheck*
- Create a private class method to display the Menu: *presentMenu*
- Create a private class method to calculate the tax: *calculateTax*

- Create a private class method to calculate the tip: *calculateTip*
- When displaying the check, make sure all amounts have their decimal points vertically aligned and display 2 digits of precision to the right of the decimal point (all amounts are *double*)

Using the Canvas Assignment upload feature, submit a Zip file named ***Lastname*Project8.zip** that includes the following files:

- Two **Source Code files (*.cpp),** one for method main() and the other for the definitions of the class methods and *const* values, a **Header file (*.h)** for the class definition, and the **Executable file (*.exe).** You do **not** need to include any additional project files.
- A **text file or word document containing short summary of your work:**
    1. What does or doesn't work and why
    2. Any additional functionality reasons or excuses for missing functionality.
    3. What you learned and what you found difficult or unclear about the problem.