COP2221 – Intermediate C++ Programming
Module #2 Assignment One

This assignment refers to **Learning Outcome #1:** Design, develop and debug modular computer programs. The program requires **repetition** structures (Chapter #5) and a good design (from the programmer – you!). You will need to use a **counter**(s) and an **accumulator**(s). Review as needed in the text and/or via YouTube videos. Your program requires **input validation**.

NOTE: this program is designed to practice the above programming concepts – it's not as elegant as a "real" user request but **repetition**, **counting** and **accumulating** are critical skills and require as much practice as possible.

Directions. Consider the following program requirements. Complete and test **ONE function at a time**. Do not write the complete program before testing – create a function main( ) that calls one function; once that function is perfected; create and test your program with a second function completed and so on…

Read the below steps before beginning your design so you know what the program will accomplish – **this is not an overly functional program but it practices some needed skills**.

1. Begin the program by calling a "welcome" function. This function generates an opening screen that welcomes the user and explains (briefly) the purpose of the program and any directions you think the user needs (**2 points**)
2. Next, call a function that asks the user to enter a series of positive numbers. Validate the users input (assure that the input is positive). The user will enter a zero to signal they have no more data to input. This function returns the number of valid entries (the number of positive numbers entered). (**5 points**)
3. Now, call a function that accepts the number of positive numbers entered by the user (from previous step). This function will display a message stating, "!0 or fewer positive numbers were entered" or "More than 10 positive numbers were entered. (**3 points**)
4. Call a "goodbye" function that displays your programming contact information (make up something). (**2 points**)

Additional requirements:

- Include opening comments (function main)
  - Programmer's name
  - Description of the program's purpose
  - Date code was last modified
- Use good programming style
  - Indentation
  - Comments within your code to explain major steps in the program
  - Blank lines to keep the code readable
- Test and debug your program one function at a time