Binghamton University

CS 211 – Introduction to Programming for Engineers

Extra Credit Lab
Functions, Functions with Pointer Parameters, If-Statements, Input, Output

Objectives

After completing the assignment, the student will:

- demonstrate the implementation of functions inside program to achieve modularity
- distinguish the differences between single return functions, and functions that return a value via the parameter list
- differentiate as to when to use a single return function and a function that return a value via the parameter list
- differentiate as to when to use a value parameter verses a reference parameter
- demonstrate the use of priming reads for a loop correctly
- demonstrate how to properly desk check results by producing output that is 100% accurate

Introduction:
Write a program that uses functions to solve the following problem. Input 4 integer quiz grades from the user and calculate the average of the quizzes, the highest and the lowest of the 4 grades, and the number of quiz grades above the average. Output the information as follows:

- The standard Course Header
- The input from the user including the prompt to the user and the 4 quiz grades on a single line
- A properly labelled student name who took the quizzes (use your name)
- A properly formatted and labelled table (with a heading) showing the 4 quiz grades that were input
- A properly labelled average of the 4 quiz grades
- A properly labelled highest quiz grade
- A properly labelled lowest quiz grade
- A properly labelled number of quiz grades above the average

Name this exercise: **`ExtraCreditLab_lastName`**
So if your name is John Smith, you would name this file, **`ExtraCreditLab_Smith`**
All input will be from the keyboard, all output will be to the screen.
You are going to create functions for tasks listed below and place them in your code in THE ORDER LISTED.

Exercise Notes

1. <u>Minimum requirements:</u>

A. The output of the program must be perfectly correct for any reasonable input entered by a user
B. The solution **<u>must not have any</u>** compiler warnings or errors
C. The solution must be able to be compiled and executed by the instructor / CA

2. Header File

Header file must have:

- The three lines to guard against multiple instantiations
- All your system includes
- Constants (macro and regular constants)
- Function prototypes.
- Macros that are need to be defined are listed in the directions below.
- Declare a regular constant for the number quizzes (4)

NOTE: Use all macros and constants in your code appropriately.

Function Order:

1. Create a divider function. You may use the divider function presented in an in-class example
   A. Do not call this function from the main function. Call it only from within other functions like the function for number 2 below.

2. Create a **void/void** function to print the standard course heading on the screen.
   A. Create macros (#define) for all strings (All macros need to be defined in your header file)
   B. Write the function to center the string on the screen. (You many copy the code from the in-class example)
   C. Your header should include school name, lab title, and your name each centered on the screen and on a separate line.

3. Write the function center the string on the screen. (You many copy the code from the in-class example)

4. Create a function to get four integer quiz grades from the user

5. Create a **Single Return** function to calculate the average of the four quiz grades.
   A. The return data type for average will be a double.
   B. Assign this function to a variable in the main function.
   C. HINT: You will need to use a cast

6. Create a **void/reference** function to determine the highest and lowest of the four quiz grades
   A. Be careful to only use pointer parameters when necessary
   B. This function should have a combination of value parameters and pointers parameters

7. Create a **Single Return** function to count the number of quizzes that are above the average
   A. You will have to cast the average (which is a double) to an integer temporarily in this function
      i. Do not change the value of the actual average
   B. The return type of this function will be integer
   C. Assign this function to a variable in the main function

8. Create a function to print a "**Quiz Results**" **heading** on the screen
   A. The first line should be your properly labelled students name (This does NOT have to be input from the user, you may use your name, however, you must use a macro)
   B. The second line should be "***Fall 2022 Quiz Results***" (another string constant (use a macro))
   C. Use the PrintDivider function as needed to make your output header look professional.
   D. You should call this function from the function you will create in step 9

9. Create a function to output ALL the user entered values and calculated results
   A. Make sure the output is clearly labeled. You may print the data in a table or one line at a time.
   B. There will be a LOT of parameter variables, output **3 decimal places** for the floating-point values
   C. See the assignment description for a list of required output

Program Notes:

1. The main function should ONLY include:
   A. Variable declarations for the quizzes and calculations (No one letter variable names)
   B. Calls to each function in the order listed above. There are only three functions that will **NOT** be called from the main:
      i. The divider should be called where appropriate
      ii. The quiz heading function should be called from the quiz output function
      iii. The function to center a string

2. Function Definitions (the actual function code) belongs below the main function in the order specified under the heading **Function Order:**
   A. Each function should have its own **function description** block as shown in the program examples
   B. For example

```
//--------------------------------------------------------------------------
//OutputDivider - prints on the screen the symbol,
//                the number of times specified by numberOfSymbols
//--------------------------------------------------------------------------
```

   C. **Do not forget to comment your code inside of your function definitions.**

3. Once the entire program is warning and error free:
   A. Select **Debug/Start Without Debugging**

4. Check your program's results for each of the three runs by performing the calculations manually using a calculator
   A. Programs submitted with incorrect results will not be graded and a grade of zero will be applied to your grade book

5. When you have all the output correct, copy and paste the screen output into a .txt file that you will submit with your lab.

6. Run the program **three** times and **copy and paste each set of output into a single output file** in THE ORDER LISTED below:
   A. 100, 85, 90, 95
   B. 45, 80, 65, 66
   C. 0, 80, 40, 81

7. Click on the Save All Icon to save all files in this project

8. Make sure you complete the assignment by the due date and time specified!

9. Once you submit your lab, you may resubmit it as many times as needed for 24-hours. After 24-hours the lab is subject to being graded.