CS 135

Design Assignment 1 (DA1-09/13)

Programming Assignment 2 (PA2-09/17)

As specified in your syllabus, you must turn your assignments in by 6:00 pm on the due date specified. If it is turned in late, but prior to 12:00 midnight the day it is due, credit will be reduced by 50% of the earned score. Any laboratories turned in more than 6 hours late will not earn any credit.

Objectives:

1) You will use a set of standardized functions to implement console I/O operations
2) You will create input and output operations with program abstraction through the use of utility functions
3) You will use global constants to assist with program clarity
4) You will initialize and use variables with a variety of data types
5) You will use functions with varying parameters, acquire data from the functions, and assign the results to appropriate variables
6) You will conduct simple mathematics in a program
7) You will become familiar with the use of a standard explanatory commenting process for developing functions
8) You will continue to use a common IDE to develop and test programs

Tasks:

Using a console-oriented I/O

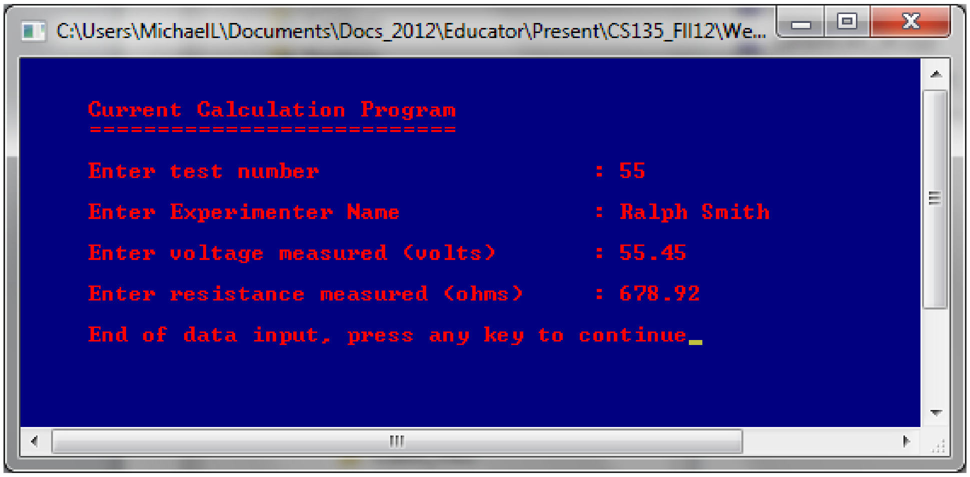
1) Your program name will be **`currentcalc.cpp`**. You will be using a slightly different I/O interface for this program. This interface is based on a library package called curses and allows you to place characters, numbers, text, etc. at specific x and y locations on the screen. It also supports a simple eight-color palette for the foreground (characters) and background.

2) This program is also quite simple in that the mathematical operation is small. The program will accept input from the user related to experiment numbers, experimenter names, voltages, and so on, and then, like the previous week's assignment, outputs them in a well organized way, along with a calculated result. Once again, most of your work will be using functions to acquire input and provide a nice looking display, although in a different kind of system.

3) Download the **`formatted_console_io_v17.h`** file from WebCampus. You may need to copy the text and paste it into your Dev C++ IDE, but if you do, make sure you save the file as a "Header file" as instructed in the first laboratory. This file has a number of utility functions that will help you implement many input and output operations.

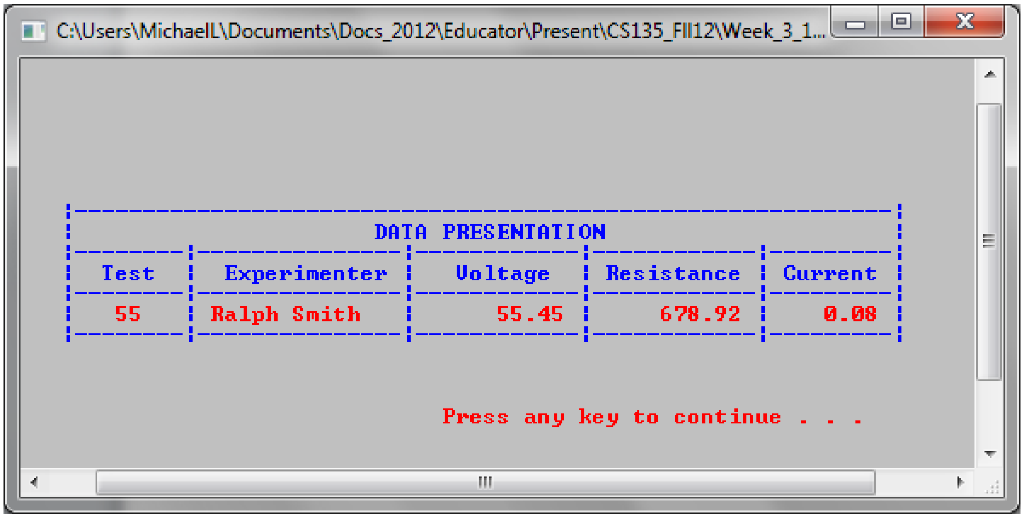
4) As was the case in the previous week’s assignment, there are instructions in the header file that support the use of the available functions and there are function descriptions for each one. You will be using function descriptions like these later on when you create your own functions, so it is worth your while to become familiar with them. Reading the function specifications also helps you understand what each function is used for and how you can use it.

5) Your assignment this week will be to create a program that first prompts for a set of data, as shown below.


C:\Users\MichaelL\Documents\Docs_2012\Educator\Present\CS135_Fll12\We...

```
     Current Calculation Program
     ===========================

     Enter test number                    : 55

     Enter Experimenter Name              : Ralph Smith

     Enter voltage measured (volts)       : 55.45

     Enter resistance measured (ohms)     : 678.92

     End of data input, press any key to continue
```

6) When the user presses any key, the input data and the calculated result are displayed, as shown below.


C:\Users\MichaelL\Documents\Docs_2012\Educator\Present\CS135_Fll12\Week_3_1...

```
|------------------------------------------------------------|
|                      DATA PRESENTATION                     |
|--------|---------------|------------|------------|---------|
|  Test  | Experimenter  |  Voltage   | Resistance | Current |
|--------|---------------|------------|------------|---------|
|   55   | Ralph Smith   |      55.45 |     678.92 |    0.08 |
|--------|---------------|------------|------------|---------|



                            Press any key to continue . . .
```

7) When the user presses any key at this screen, the program is ended.

8) The calculation for the current is voltage divided by resistance. Note that the test number is justified in the center of the display box, the experimenter name is justified to the left of the display box, and the numerical values are right justified in their respective boxes.

9) For your Design Assignment, you must create the first three steps of the Six Step Programming Process that will be discussed in class. This program is very simple, but you need to organize the program using modular structure, and then identify the appropriate modules (i.e., functions) you will be using to solve the problem. Your Programming Assignment for this laboratory will be your step 4.

10) A few programming notes:

a) to paint an entire screen a certain color, you must set the color first and then run the **`clearScreen`** function; the colors you may use are all specified in the formatted console header file

b) any program that uses the formatted console tools must place a **`startCurses`** function in the initialization area of the program (but after variable initialization); the program must also place an **`endCurses`** function as the last step before ending the program with the return statement

c) you must specify an x and a y position for everything you intend to place on the screen, including prompt and display actions; you must NOT use literal numbers but create variables such as **`xPos`** or **`yLoc`** and set those values as needed to provide information to the console functions; you may also use global constants in function calls as well

d) note also that you may not use any iostream functions such as cout or cin; you will only need the formatted console header file and you must use only those functions

11) Make sure that your input and output processes look as much like the example above as possible; in most cases, it should look exactly the same. Responses to prompts MUST be on the same line as the prompt (as shown), and the table output must be well organized.

12) Make sure that your program design is appropriately structured in blocks that represent each part of the task that you are resolving. Program readability will be graded in this and many of the laboratory assignments.

13) Create three example screenshots of this program using the data above for your first, and your own data for the next two. Remember to annotate each screenshot.

14) Important Assignment Note:

a) You must turn in your Design Assignment by Thursday (09/13) evening at 6:00 pm. We will not be grading them rigorously since this will be your first time doing it, but we need to see a reasonable attempt at designing the program.

b) The required design will be released after midnight on Thursday (after the Design Assignment deadline) for your review; it will be located in the Programming Assignment. You **MUST** use the course-provided design structure you download from WebCampus for your program (as opposed to your own).

c) Again, check the rubrics so you will know in advance how your work will be graded.

Turning in your Design Assignment:

Information:

Week: 3
Laboratory: Week 3
Design Assignment: 1
Due Date: 09/13, 6:00 pm

To turn in:

The first three steps of the Six Step Programming Process, including:

1. currentcalc_s1.cpp
2. currentcalc_s2.cpp
3. currentcalc_s3.cpp

Upload these as separate files.

For information on how to turn in Design Assignments, refer to the "How to Turn in Design Assignments" in the "General Course Information" folder

Turning in your Programming Assignment:

Information:

Week: 3
Laboratory: Week 3
Programming Assignment: 2
Due Date: 09/17, 6:00 pm

To turn in:

1. The Word file containing the following:
   a. There should be at least three (3) screenshots for the programs as specified in item #13 previously in this document
   b. Remember to clearly annotate every displayed result

2. The executable files:
   a. currentcalc.exe (currentcalc_s4.exe is acceptable)

3. The source code files:
   a. currentcalc_s4.cpp

These files must be compressed and uploaded as one zip file. To do this, select all of the required files, right click on them, and select “Send To”, then select “Compressed (zipped) Folder”.

Once the folder is created, it will be placed in the same folder in which you are working. Change the name of the zipped folder to “LastnameFirstname_PAX” (where ‘X’ is the number of the Programming Assignment) as shown in the following example: “LeveringtonMichael_PA3” (no quotes). After you have renamed the zipped folder, double click on it to verify that it has all the files it is supposed to have.

For information on how to turn in Programming Assignments, refer to the "How to Turn in Programming Assignments" in the "General Course Information" folder