**ECE 1111: Engineering Computation I**

**Homework No. 2: Formatted I/O**

**Goal:** Demonstrate that you can print formatted output to the terminal using the function fprintf. You will need to read about the specifics of how you print formatted data using any of the resources at your disposal. For example, you can learn about this using Google search or consulting any of the textbooks listed in the syllabus.

**Deliverables:** Include these files as separate attachments in your email (no zip files or archive files, just plain attachments):

1. A pdf file containing your homework assignment documenting your results, as explained in the course syllabus.
2. myprog\_01.cc containing the code requested below.
3. myprog\_02.cc containing the code requested below.
4. myprog\_03.cc containing the code requested below.

If your code fails to compile, you will receive a failing grade on this assignment.

**Note that you are not allowed to use cin and cout in this course. We will focus on using fprintf until you are very comfortable doing formatted I/O.**

**Description:** The tasks are:

1. Create a program that prints "hello world" to the terminal screen. Call this myprog\_01.cc. Compile it, run it, and demonstrate its output in your homework solution.
2. Create a second program, myprog\_02.cc. Declare these variables in your program:

**char c = 'q';**

**short int i = 27;**

**long j = 27272727;**

**float sum = 27.272727272727**

Demonstrate the use of fprintf to print these to the terminal using the following format:

**the value of <insert variable name> is <insert value>**

For example, for the short integer, the output should look like this:

**the value of i is 27**

1. Add to myprog\_02.cc code to do a formatted print statement. Construct a formatted print statement that prints the long integer and floating-point values in the same line:

**the value of I is 27 and the value of sum is 27.27**

Pay special attention to how to format the floating-point number so that only two decimal points of precision are printed. You can accomplish this by adding a format specification to the "%f" specification in your fprintf statement.

1. Create a file called "mytext.txt". It should contain the lines below:

**This is line number 1.**

**This is line number 2.**

**This is line number 3.**

**Hello world.**

**This is line number 5.**

Write a program called, called myprog\_03.cc, that reads this data from stdin:

**myprog\_03.exe < mytext.txt**

and prints the contents of this file line by line to the terminal.

You will need to Google search the solution to this problem since it involves writing a loop (which we haven't covered yet). Your program should work for any length file (the file can be 5, 6 or 1,000 lines long). Compare your output to what the "more" command produces.