**ECE 1111: Engineering Computation I**

**Homework No. 11: Header Files and Program Structure**

**Goal:** Streaming data systems require software to be able to handle I/O in variable-sized chunks of data. In this assignment, we will integrate several concepts we have discussed to develop a robust capability to process data read from a file. You will implement this assignment using a structured programming approach where you code is organized into a set of files that maximize reusability.

**Description:** The main goal of this assignment is simple: rewrite HW #7, in which you read data from a file and provided the ability to analyze windows of data centered about a frame, to use a circular buffer. You can use the code I provided previously as a starting point. The command line interface to your submission should be the same as before:

**myprog.exe myfile.dat 2 6**

The input and outputs should be the same as before. However, the major difference is that you will implement the I/O using a circular buffer so there is no shifting or large memory moves in your code.

This means you will need to implement a circular buffer. This should be done in a file cb\_00.cc, and a library libcb.a. Compile your source code and install it in a library. Link your main program with this library. It is up to you to decide what interface functions you provide for your circular buffer implementation, but the interface should be designed for reusability and ease of use.

In your main program, you will have a read loop that reads data from a file in chunks. The size of these chunks should vary randomly so we simulate an application that reads data from an unreliable source such as an Internet link. Your main loop should look something like this:

* Use a random number generator to generate an integer, N, that is your current buffer size.
* Read N samples from the file, simulating the fact that when you read data from a source on the Internet (like Netflix) you never know exactly how much data will be available.
* Insert this data onto a circular buffer and advance the write pointer.
* If there is enough data to fill a window, print the contents of the window and advance the read pointer. Print the contents on one line using the format I provided in the code I previously sent.
* Go to the top and repeat until there is no more data to read (use a while loop).

Start by first developing your circular buffer code. Write a driver program that adequately tests it. Then integrate this functionality into the main program I provided so the main program does not use memmove or any other such block memory move.

To submit this assignment, I want you to send me a compressed tar file (\*.tar.gz) that contains all the code necessary to compile, link and run your main program. This will presumably include a Makefile, a header file (cb.h) in which you declare the interface to your circular buffer implementation, a main program (myprog.cc) that supports the interface described above, and an implementation file (cb\_00.cc) that contains the implementation of your circular buffer class.

The filename your submission must have is the name lastname\_firstname\_hw11.tar.gz. It must be an attachment in your email. No other input format, such as zip or rar, will be accepted. To create a tar file, let’s assume your code is in the directory: $HOME/ece\_1111/hw\_11. Do the following:

* Delete all the binary and object code in your hw\_11 directory
* cd $HOME/ece\_1111
* tar cvfz lastname\_firstname\_hw11.tar.gz -C $PWD hw\_11

Submit this tar file as an attachment to your email. Your code should be well formatted and well commented.