# CST8234 – C Programming F17 (Lab 5)

## Programming Exercise

In this lab, we will gain experience with

- Using self-referencing structs (linked lists)
- Formatting delimited lists

## Statement of the problem

You will successively add integers into a singly-linked list, such that they are automatically sorted. Each node in your list will have a 'number' member which is a randomly generated from 1-1000.

After each insertion, you'll print out the entire list.

Once you have added all your numbers. You'll randomly select a number, and remove it from the list.

After each deletion, you'll print out the entire list.

When the list is empty again, exit.

Your output must be formatted as follows.

```
$ ./lab5.exe
Numbers to add to list? 5
Inserting 82
List: [82]
Inserting 662
List: [82, 662]
Inserting 900
List: [82, 662, 900]
Inserting 4
List: [4, 82, 662, 900]
Inserting 415
List: [4, 82, 415, 662, 900]
Deleting 82
List: [4, 415, 662, 900]
Deleting 900
List: [4, 415, 662]
Deleting 415
List: [4, 662]
Deleting 662
List: [4]
Deleting 4
List: []
```

## Requirements

1. Create a folder called algonquinUserID_L5 (e.g., "mynam00123_L5"). Do all of your work in this folder, and when complete, submit the zipped folder as per the "Lab Instructions" posted on Blackboard.
2. Write a program that
   a. Asks the use for the number of items to add to a list
   b. Creates a node containing a random number between 1-1000
   c. Reports what number it will be inserting
   d. Finds the appropriate place in the list to insert the node, such that the numbers are in increasing order
   e. Inserts the node into the list
   f. Prints out the list in the specified format
   g. Repeats steps b-f for until the list has grown to the size captured in step a
   h. Randomly selects one of the nodes for deletion
   i. Reports what number it will be deleting
   j. Removes the node from the list
   k. Prints out the list in the specified format
   l. Free's the removed node
   m. Repeats steps h-l until the list is empty
3. You must have a node.c/node.h in addition to the source file that contains your main function. You will use a makefile will invoke gcc with the standard compiler flags (`-g -Wall -w -pendantic -ansi`)

## Marking

This assignment is out of 30

- 10 for coding correctness (i.e., correct results)
- 10 for appropriate and efficient logic (i.e., no spaghetti code!)
- 10 for clear comments and coding convention (not necessarily K&R, but it should be clean and consistent)

You can also lose marks for incorrect submission (e.g., including unnecessary files in the zipped folder), compiler warnings, typographic errors (including in comments).

## Submission

When you are done, submit your program to Blackboard. Make sure that you have the appropriate header in your source file(s), and have zipped up the appropriately name directory. Only include the source code (.c, .h) and your makefile (mandatory!).

You ***must*** include a makefile, as part of your submission! When grading your reports I will unpack your zip file and type '`make`'… and if I don't end up with a 'lab4.exe' to run, ***I will not grade your assignment***.