## Assignment 3 ****Option B**** (Chapter 5) - Revised sequence class with the Linked List Toolkit

**The Assignment:**

You will implement and test a revised sequence class that uses a linked list to store data items.

**Purposes:**

Ensure that you can write a small class that uses the linked list toolkit to create and manipulate a linked list.

**Before Starting:**

Read all of Chapter 5, with particular attention to Sections 5.3 and 5.4.

**Due Date:**

N/A

**How to Turn In:**

Pack your files (source code only!)  in a single WinZip file (Windows) or tar file (Unix/Linux) and submit it through Blackboard. Please do not submit your executable files.

**Files that you must write and turn in:**

1. sequence3.h: the header file for the new sequence class. You can use the textbook version at <http://www.cs.colorado.edu/~main/chapter5/sequence3.h>. By the way, you might want to compare this header file with the header file of the sequence class using fixed size array (<http://www.cs.colorado.edu/~main/chapter3/sequence1.h>), and the header file of the sequence class using dynamic array [(http://www.cs.colorado.edu/~main/chapter4/sequence2.h)](file:///C:\teaching\teaching\teaching\teaching\CSc21200_S12\Logistics\(http:\www.cs.colorado.edu\~main\chapter4\sequence2.h)). The linked list version no longer has a CAPACITY constant nor a DEFAULT\_CAPACITY constant because the items are stored on a linked list instead of an array.
2. sequence3.cxx: The implementation file for the new sequence class.
3. node1.h (from http://www.cs.colorado.edu/~main/chapter5/node1.h) and node1.cxx (from http://www.cs.colorado.edu/~main/chapter5/node1.cxx). Copy these files to your subdirectory. They contain the linked list toolkit from Section 5.2. You may use these files without changing them. For function list\_piece, you can use the implementation discussed at the slide #43 of lecture 10 if you are going to use list\_piece in writing your copy constructor and overloading your assignment operator. If you do make changes on node1.h and node1.cxx, please turn them in.

**Other files that you may find helpful:**

1. sequence\_test.cxx: This is the same interactive test program that you used with the earlier sequences <http://www.cs.colorado.edu/~main/chapter3/sequence_test.cxx>. This is exactly what we expect for the information hiding of the class - the users do not need to know how the class is implemented (both the member variables and the functions) as far as the interfaces (the prototype of the member functions) remain the same. So if you want to use the test program with the new sequence, the only thing you need to do is to copy it to your directory and open it with your editor. Then change the statements

#include "sequence1.h"   
using namespace main\_savitch\_3;   
to   
#include "sequence3.h"   
using namespace main\_savitch\_5;

1. seq\_ex3.cxx (<http://www.cs.colorado.edu/~main/chapter5/sequence_exam3.cxx>): A non-interactive test program that will be used to grade the correctness of your new sequence class. The points given by the program will only be served as a reference; we are going to look into your code and your analysis (in the form of comment lines) for actual grading.