

Programming Assignment 4: The Sequence Class with a Linked List

The Assignment:

You will implement and test a revised sequence class that uses a linked list to store the 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:

Tuesday, March 26, 2019. No work will be accepted after Tuesday.

How to Submit:

1. Pack your source code files in a single zip file. Name the zip file in the following way: **LASTNAME_FIRSTNAME_ASSGN#**, where LASTNAME, FIRSTNAME, and # are filled in.
2. Attach this file to an email with the following subject line: **CSc212 Assignment 3 Submission**.
3. Send this to **tcrupi@qc.cuny.edu**.

Files that you must write:

1. `sequence3.h`: The header file for the new sequence class. Actually, you don't have to write much of this file. Just start with our version and add your name and other information at the top. If some of your member functions are implemented as inline functions, then you may put those implementations in this file too. By the way, you might want to compare this header file with your first sequence header file (`sequence1.h`), and second sequence header file (`sequence1.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. You will write all of this file, which will have the implementations of all the sequence's member functions.

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. If you want to use it with the new sequence, then copy it to your directory and open it with your editor. Then change the statement:
`#include "sequence1.h"` to `#include "sequence3.h"`
and change the namespace to `main_savitch_5`.
2. `node1.h` and `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.

Discussion of the Assignment

Your sequence class for this assignment will differ from the your previous sequence in the following ways:

- The sequence's items are now stored on a linked list. The head pointer of the linked list is a private member variable of the sequence class. I suggest that you also have a tail pointer as an additional private member variable of the sequence class. The reason for the tail pointer is explained in Section 5.4 of the class text.
- Because you are dynamically allocation memory within your sequence class, you will need to define a copy constructor, an assignment operator, and a destructor.

Start by declaring the new sequence's private member variables in sequence3.h. You might try declaring these variables yourself, and then compare your solution with the suggestion in Section 5.4.

Once again, do your work in small pieces. For example, my first version of the sequence had only a constructor, start, insert, advance, and current. My other member functions started out as stubs.

Use the interactive test program and the debugger to track down errors in your implementation. If you have an error, *do not start making changes until you have identified the cause of the error.*

For those working in the Unix operating system: Spend some time writing your make file. A correct Unix makefile will save you time. Make sure that each compilation command in the make file has a TAB as the first character. In order to insert a TAB using emacs, type Ctrl-Q and then press the TAB key. Or, if you are working from a modem where the TAB key is disabled, you can type ESCAPE x quoted-insert RETURN TAB.

More info: <http://www.cs.colby.edu/maxwell/courses/tutorials/maketutor/>