Programming Assignment #2

List ADT implemented with a Linked List

CS 3358.253, Spring 2015
Instructor: Jill Seaman

Due: Wednesday, 2/25/2015 (upload electronic copy by 1:30pm)

Problem:

For this assignment you will implement the List ADT from the class demo (List_3358 on the class website) using a doubly linked list. This will make the insert and delete O(1). The class demo is implemented using an array, and the insert and delete are O(n).

Use the following header file: **list_3358_LL.h** (on the class website). The header file contains the datatype and member variables necessary for the doubly linked list. Make a separate implementation file (list_3358_LL.cpp) for the member function definitions.

I recommend writing a good test program (similar to list_test.cpp) to make sure all the functions work properly. Since the array-based implementation of list_3358.h and the linked list implementation of list_3358_LL.h share the same exact interface, you should be able to use the same test program for both of these implementations—and you should get the exact same results! You may want to add some tests to the list_test.cpp program (or write an entirely new one) to sufficiently test your linked list implementation.

NOTES:

- Read the comments in the *.h files carefully (they are the same as in the class demo). They explain what each function needs to do.
- Do not implement the copy constructor until last (you may want to implement it by calling other function(s) in the class).
- Make sure you maintain both the next AND previous pointers (and head AND tail) as you implement the operations. Use NULL for EOL.
- Run your test program on the class demo to get the correct output, then run it on your linked list implementation to validate the output from your implementation.
- The purpose of this assignment is to get experience implementing an ADT in two different ways, to practice with linked lists and pointers, and to get more familiar with the separation of interface and implementation.

Style:

See the Style Guidelines document on the course website. You do not need function description comments this time (they are already in the *.h file).

Logistics:

You need to submit only one file for this assignment:

`list_3358_LL.cpp`

We will test it using the original list_3358_LL.h file and our own test driver.

Submit: an electronic copy only, using the Assignments tool on the TRACS website for this class.