COP3330 Programming Assignment 9

A doubly Linked list and an application

Objectives

* Practice using dynamically allocated memory and pointer operations
* Practice implementing a linked list class
* Practice using linked lists to solve problems

Project details

This assignment has two tasks. The first is to implement a custom doubly linked list class called mylist that is designed to be used in the word count program (assignment 8). The second is to rewrite your assignment 8 program to use mylist to store the identifiers and numbers found.

A complete header file, a skeleton implementation file, a complete test driver program, and a makefile to compile the programs have been provided to you:

[mylist.h](file:///C%3A%5CUsers%5Csurfing.root-PC%5CApplication%20Data%5CSSH%5Ctemp%5Cgiven%5CMyString.h)
[mylist.cpp](file:///C%3A%5CUsers%5Csurfing.root-PC%5CApplication%20Data%5CSSH%5Ctemp%5Cgiven%5CMyString.cpp)
[test\_driver.cpp](file:///C%3A%5CUsers%5Csurfing.root-PC%5CApplication%20Data%5CSSH%5Ctemp%5Cgiven%5Cdriver.cpp)

makefile

You should not change mylist.h in any way. Submissions will be tested using the mylist.h.

The skeleton implementation file, mylist.cpp, contains detailed descriptions of each of the functions as well as some sample implementations for some routines. The complete routines are marked by “this routine is complete” in the comments. The places where you need to write code are marked by “change me, your code here”. You should first complete the mylist implementation so that the test driver program produces the same output as the sample executable given. Once you complete the mylist class, you must then rewrite proj8.cpp program by using mylist to store the identifiers and numbers found. The rewrite should be very minor if your proj8.cpp works.

This project is the largest and the hardest in the semester. You will need to use the similar steps as those used in assignment 7 to complete the project:

1. Look over mylist.h to get an overview of what the mylist class looks like (also the class is also discussed in Lecture 17).
2. Review the function descriptions in mylist.cpp. It is not necessary to know how to implement each function yet, just get an idea of the input/output parameters and the general purpose of the function.
3. It is strongly suggested that you read and understand the complete routines given in mylist.cpp before starting your code development. The code patterns that you need to write can be found in the completed routines.
4. Start filling in the function definitions. You should implement the routine following the testing order in the test\_driver.cpp.
5. Go to recitation, office hours, lecture and email your TAs and/or me with pointed questions.

Submission

The due time for this assignment is July 26 (Friday), 2013. 11:59pm.

Name your new word count program proj9.cpp. Tar proj9.cpp, mylist.cpp, mylist.h, test\_driver.cpp, and your makefile and name the tar file yourlastname\_firstinitial\_proj9.tar and submit the tar file in blackboard.

Grading policy

The program must work on linprog. O point for programs with any g++ compiler error on linprog. The insertpos and removepos routines should NOT call any other routine.

* ‘g++ proj9.cpp mylist.cpp’ and ‘g++ test\_driver.cpp mylist.cpp’ have no compiler error (20 points)
* Tar file name, comments in program, etc (5 points)
* Correct makefile: one make command should produce all executables for this project (5 points)
* Insertfront and removeback(10 points)
* Insertafter and insertpos from scratch(10 points)
* Copy constructor and assignment operator (10 points)
* Remove and removepos (10 points)
* Back (5 points)
* Findmaxcount and searchandinc (10 points)
* New word count program with mylist as the storage for identifiers and numbers (15 points)

Hints

1. Start the project as soon as possible. This program is largest and hardest among all assignments.
2. Understand the routines given first before you start your own coding.