CS400/600: Data Structures and Algorithms

Programming Assignment #1

Due Sep. 21st, 2010, 11:55pm

1 Overview

The objective of this programming assignment is to refresh your C++ programming skills, while getting used to using classes and designing modular solutions to problems.

For this assignment, you will turn in two files: LinkedSortedList.h and LinkedSortedList.cpp. These files will define your sorted, linked list. I will provide you with the following files:

- SortedList.h – The abstract data type to be implemented. Your LinkedSortedList must be declared as a child class of the SortedList class.
- LinkedNode.h – A simple, single-linked list node. You may modify this file if you wish to use a doubly-linked implementation.
- main.cpp – Code that I will use to test your implementation.

2 Requirements

1. Your code should follow the Code Standards posted on webCT. Your code will be graded according to its correctness, efficiency, organization, and readability.
2. Make sure that each file includes your name in the header comments.
3. Your class should be implemented as a template, and instantiated for two element classes: LinkedSortedList<int> and LinkedSortedList<double>.

3 Code Submission and Testing

1. You must submit an ELECTRONIC COPY of your source program and report through webCT before the due date. If for some reason WebCT is unavailable, submit your source code by email to wlodarski.4 AT wright.edu. If you want, you can also cc to the instructor Meilin Liu, whose email address is meilin.liu AT wright.edu.
2. Submit all your source codes, makefile, executables, README, and any other required files. You must explain your programs clearly in the README file. You must write down your name and email address in the README file.
3. The grader will test your programs under the schools UNIX environment, e.g., unixapps1.wright.edu. It is YOUR responsibility to make your programs workable and runnable by others under school's UNIX environment.

4. The programming assignment is individual. You must do the project by yourself. If you allow others to copy your programs or answers, you will get the same punishment as those who copy yours.

4 Grading

This is a 50 point lab. Points will be assigned as follows:

1. Correctness: 30 points – your program should compile correctly, and your LinkedSortedList should inherit from the SortedList class provided in SortedList.h. The test code that I provide to you should execute correctly.

2. Readability and Style: 20 points – your code should be well written, well documented, and readable. See the Coding Standards handout for details.