CS400/600: Data Structures and Algorithms
Programming Assignment #2
Due Oct. 7th, 2010, 11:55pm

Overview

For this assignment, you will write a simple database for storing and retrieving employee records. This database will serve as a testing ground for future data structures. You should begin by creating class Employee, which will contain information about a single employee (see below). Your database will then consist of a LinkedSortedList of Employee objects sorted by last name. Note – that if you used the standard comparison operators (==, <, >, <=, >=, etc.) in your LinkedSortedList implementation, you will need to overload these operators for your Employee class.

Your database should support the following fields: **Last Name (string), First Name (string), Employee ID (integer), Salary (integer), Department (string), Phone Number (string), Office Building and Number (string), Hire Date (string), and Email address (string)**. Your database **should allow duplicate values**, as two employees might have the same last name (and even the same first and last names).

Your database should have a user interface that supports the following operations:

- Insert new records (prompt user for all fields)
- Search on Last Name (print ALL matching records)
- Save database to disk
- Load database from disk

Running your program should produce a menu similar to the one shown in the example below.

Saving the database should store records in last name order. When loading a database from the disk, all current records should be deleted, and the database should be loaded from a file, and the list should be rebuilt. An example of a saved database file will be provided in the assignment files. Your program should be able to read this file and write files in this format.

The search operation should report the number of Employee objects searched, in addition to the results. For example, a Last Name query for the name “Powers” would return results similar to the following:

```
MENU
(I)nsert new record
(L)ast name search
(S)ave database to a file
(R)ead database from a file

 Enter choice: L

 Enter Last Name:  Powers
 Searching….

 4 Employees searched.  Found 1 record:

 Last:    Powers
 First:   Susan
 EID:     662312
 Salary:  $105,000
 Dept:    Legal
```

REQUIREMENTS

1. Your code should follow the Code Standards handed out during the first day of class. Your code will be graded according to its correctness (70 pts), and style/readability (30 pts).
2. You should implement the Employee class in the files `Employee.cpp` and `Employee.h`. You do not need to use a template for this class, nor will I be providing an abstract class for you to implement.
3. You should use a `LinkedSortedList<Employee>` to hold your employee objects.
4. Your main program should create a user interface similar to the example above. The file name for the main program should be `lab2.cpp`.
5. Turn in all files needed to compile and execute your code (including any needed files from the previous lab) via webCT. 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.
6. 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.
7. If there are any special instructions that I need to know about, be sure to include a file named README.TXT detailing them.
8. 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.