Computing and Algorithms II: Programming Assignment 2

CS-102 **Fall 2016**

In this programming assignment, you will demonstrate your knowledge of dynamic data structures (in particular, linked lists) by revising the tropical storm data program from Programming Assignment 1 to incorporate a dynamic data structure.

New Internal Requirements

For this assignment, you should replace your `Database` class class with a new implementation which uses a *linked list of linked lists* to organize the information within the database.

The nodes in the upper-level linked list will represent years, with one node per year. Each year node will contain a reference to a linked list of storms for that year.

You should maintain the items in each linked list in sorted order, sorting the upper list by year (in numerical order, low-to-high) and the lower list by name (in standard dictionary order).

You may use any variation on linked lists which you desire; dummy head nodes, head/end pointers, doubly-linked lists, *etc.*. You may implement additional classes as desired in order to manage the list. (In particular, consider implementing "Node" classes which contain links to other objects, such as `Movie` or `Showing` objects.)

New Functional Requirements

Your program will be an extension of Programming Assignment 1, and thus should operate in the same manner as that assignment, unless otherwise specified herein. In particular, this means that any errors present in your submissions for Programming Assignment 1 should be fixed for this assignment.

The following new requirements should be implemented as well:

- All print commands should print the contents of the database in the same sorted order(s) used to store the items. (This should be the natural way to code the print commands in any case.)
- A new command should be implemented which allows the user to insert a new storm from the main menu line. If selected, the program should prompt the user for all the necessary information and insert the transaction into the data structure appropriately. If the user enters a storm whose specified year and name already appear in the database, the entry should be rejected and the user notified. (Note that the same name may appear in different years.)
- A new command should be implemented which allows the user to delete a storm. If selected, the program should prompt the user for the year and name of the storm to be deleted. If a matching storm exists in the database, the user should be asked to

confirm the deletion, and appropriate action taken. If no matching storm exists in the database, the user should be informed of that fact.

Submitting Your Program

Before 11:59:59 p.m., Wednesday, 26 October 2016 (4th Wednesday), you must upload a zip archive to the course Blackboard assignment for Programming Assignment 2. This zip archive must contain all source code files for your program, including a class named `Prog2` with a `main` method.

In addition, you must deliver to the instructor a printout of your program files at the start of class on Friday, 28 October 2016 (4th Friday).

Notes

1. *Plan for the future!* If your submission for Programming Assignment 1 was well designed, the number of changes to classes outside of the Database class should be minimal. In Programming Assignment 3, you will be asked to replace the Database class with an implementation of a different dynamic data structure; again, changes outside of the Database class should be minimal. Design your program with this in mind.

2. Keep in mind that the midterm will be held on 4 November (5th Friday). Finishing this program (and understanding it) will be excellent preparation for the exam ...