Program Description

• This program takes in a movie text file, creates individual doubly linked lists, organizes movies, and allows the user to search for a movie.

Purpose of the Assignment

• The purpose of this program is to obtain knowledge of linked lists and implementing and remastering class templets. We use a new operator (operator <) and we needed to learn how to use this new operator in order to get our program to work.

Data Structures Description

• Write which data structures have you learned about by doing in this programming assignment.

I learned how linked lists save running time because it does not have to reallocate a whole new array for every node that is added. The values that are stored are also very easy to reorganize and saves even more computing time compared to an array.

• Write about implementation of data structures using the C++ programming language.

Big O complexity is reduced when using a linked list (that adds nodes when required) compared to an array that must reallocate memory for a whole array when adding a node.

Algorithm Description

• Write about algorithms that you used to solve the problem and describe their implementations.

InsertBefore: I had to redirect the pointers of the previous and current nodes after creating a new node. I dealt with creating the node first, then rearranging the pointers.

InsertAfter: I had to redirect the pointers of the next and current nodes after creating a new node. I dealt with creating the node first, then rearranging the pointers.

DeleteBefore: I had to redirect the pointers of the previous and current nodes then I had to delete the previous node. I dealt with rearranging the pointers first, and then I deleted the node.

DeleteAfter: I had to redirect the pointers of the next and current nodes then I had to delete the next node. I dealt with rearranging the pointers first, and then I deleted the node.

Copy Constructor: I had to make sure that I was copying everything in the linked list, not only the linked list, so I focused on creating the list, then I went 1 by 1 through the list and copied the elements.

Assignment Operator: I had to delete the list one at a time starting with the first node, until I got to the end of the list. Once I deleted everything in the list, the header pointed to the trailer and vice versa. Once the entire list was deleted, I copied my copy constructer's functions and created another deep copy of the list.

Output Operator: I had to make a temp node in order to print out the nodes one at a time, so I set the temp node equal to the first node in the list, printed out the elements, and had the node go through the list until it reached the trailer.

Part II:

Overloading the input, output, less-than operators, and their complexity analysis

Overloading: overloading the input replaces each of the values, so the runtime complexity is O(n) as it goes through n elements.

Overloading Output: overloading the output prints out each different value that is changed, so the runtime complexity is O(n) as it goes through n elements.

Overloading < operator: the overloading < operator will go through each of the lists comparing the values of each of the stored values with the input value, the runtime complexity varies from (1/26)O(n) to O(n)) because it may obtain the correct value on the first list, or worst case scenario it obtains the same value or no value by the 26^{th} list.

• Analyze algorithms according to assignment requirements.

Program Organization and Description of Classes

• List all the classes or interfaces you used and show relation between them.

I used a record class to store the values of each linked list, and I used a template class to create the linked lists. The linked lists that were created via the template class were using the record class to store and overload the information that is required (title, isbn, edition, author, year).

– Main Class

The record class was the main class, as it stored and accessed the values of each linked list. The template class was only used to create each list.

– Classes you used to implement the main data structures

Record class made use of the data structures while the template class was used to create the data structures.

– Exceptions classes

Template class for creating linked lists.

– Other classes

Instructions to Compile and Run your Program

• How to compile: (specify the directory and names of files)

g++ -std=c++11 *.cpp –o Main

• How to run: (Specify the name of a file to run)

./Main

Input and Output Specifications

introduce one item per line or all items on the same line. Make an example.

Input: Title: Pride, Author: Random Guy, ISBN: 978-0439064872, year: 2001, Edition: 1st edition

Output (searching for the same ISBN): Title: Pride, Author: Random Guy, ISBN: 978-0439064872, year: 2001, Edition: 1st edition

• Cases in which your program crashes because of wrong input (eg, wrong file name, letter instead of number, or the program expects 10 items and it only finds 9.)

none

• Cases of wrong input that you catch with Exceptions.

Range error, could not open file, error input fail (check isbn), integer exception, error unknown.

Logical Exceptions

List cases in which your program crashes or when you catch on exception due to logical execution problems. For example

division by 0, or deletion of an item from an empty list.

• Cases in which the program crashes

none

• Cases in which you catch exceptions

Range error, could not open file, error input fail (check isbn), integer exception, error unknown.

C++ object oriented or generic programming features

List and describe the C++ object oriented or generic programming features you used in this assignment:

• Templates – used to create several linked lists

Tests

Test the logical correctness of your program. Do not test the menu. Describe the cases, and copy and paste the execution of your program in those cases. For example, insertion of an item at the beginning of a list or at the end, or different type of sorting.

• Valid Cases

The Harrowing

J.J

973-12312312

2016

1st Edition

Outputs the same except in a sorted list

• Invalid Cases

1

2

3

4

5

Outputs out of range error and terminates

• Random Cases

Random

Case

999-99999999

2015

Random Edition

Outputs:

Random

Case

999-99999999

2015

Random Edition

In the 10th slot of the list.