Programming Project Report

Name: Cameron Eddy
Date: 6/12/2024

Academic Integrity Statement: I pledge that I have neither given nor received unauthorized help on this programming assignment.

Problem Statement:

The goal of this programming assignment was to use the abstract data type of a linked list. The program only had one input which was the text file that was to be read. The only error checking for this was that the file existed and was formatted correctly. The program outputs library information such as what books are in the library, what authors have books in the library, when a book is checked out, and when a book is returned.

Design:

This project required a linked list abstract data type. Each node was made through a class called BookNode and the linked list itself was made through a class called Library. I'll highlight the unique things I put in each class.

In the BookNode class instead of having a setNumAvailable() function I set the number available using returnBook() and checkoutBook(). These functions would either decrease or increment the number of books available making it easier to keep track of how many copies of that book were in the library at all times. For the library class, I used two search functions one for searching by title and one by author. The title search is used for returning books as one author could have multiple books so seeing if the book was already in the library using the searchAuthor() function would have been meaningless. However, for the input that required searching books in the library written by a certain author, the author search was definitely needed. My insert function is a sorted function that first will sort by author and then by title as seen with the nested while loops in the function.

Implementation:

No starter code besides the project2-books.txt file. First I made the BookNode class. Then I made sure my main.cpp could correctly read in the file. Next, I moved onto creating the Library linked list. I started by creating my sorted insert function. Because I got the insert function to sort by author and title this made the implementation of the search and delete function really simple and the project was a lot more straight forward from there.

Testing:

The only testing needed was to make sure it would not let me put a file in that did not exist and that the file was formatted correctly. If either of these were not correct I printed an error message telling the user what they did wrong. I also tested my code incrementally with lots of print statements to make sure everything worked the way it should have. A few other error like things were break statements in while loops and lazy evaluations incase my pointers were ever null.

Conclusions:

I definitely spent way too much time on the insert function. All though I got it to work correctly this project would have taken a lot less time if I hadn't do it that way. However, I'm proud I got it to work correctly. Overall this project took me around 10-15 hours to complete. Next time I will use the box method more when dealing with abstract data types instead of moving things around and praying it would work.