Kyle Kelly

C++ Program

Library Management System

**\*\*\* Please do not run the borrow or reserve methods first as these are not working 100% and will cause problems testing the other parts of the program that function properly.**

**Architecture of Program**

Throughout the project we mainly followed the given suggestions as to how to design our project. Additionally, we included multiple new classes to create a cleaner program that would also allow us to edit our code much easier.

Our classes were the following:

**Final\_Main**

Void main:

Runs file reader

Introduction message

Runs authentication method located in Final\_Main

Bool authentication:

Determines if username is correct

Masks the password input and determines if it is correct

Sends to the class DisplayOption to show various methods to be used

**Book**

Constructors allow for creating new books

Setters allow for altering current books

Accessors allow for finding books

getNextReserver method finds who first reserved the book and gives them access to it

getnReserved returns the number of reservers

printBook outputs the book information in format to books.dat

**Copy**

Setters allows librarian to alter copies and create new ones from current books

Setter for borrower name, borrowdate, reserverdate, and experiation date

Accessors for all of these as well

PrintCopy prints the proper format to the file copy.dat

**DisplayOption**

Constructor determines what version of displayOption to use, 0 = reader, 1 = librarian

**Following methods alter the data files**

getUserType and setUserType can also alter this

searchBooks displays the 5options when searching books

we use a switch statement to search depending on which method was chosen

sortPrint sorts and prints vector of books found in the search

addBook allows the librarian to add books to the library

deleteBook is used when there are no copies remaining or when librarian deletes

searchUsers searches for a username by username and returns the user info

addUser allows the librarian to enter either a student, teacher, or librarian

deleteUser allows librarian to delete a user

displayInfo displays username password and other info regarding the user

borrowBook allows users to borrow a book when they are eligible

returnBook allows users to return borrowed books, effects penalties and future borrowing

reserveBook allows users to reserve a book, puts them in line to take out a book

cancelReservation lets a user cancel a reservation, takes them off list

changePassword allows a user to change their password

switchOptions adjusts clock and brings back to the users options list

displayReaderOptions shows the possible options a member of the reader class may use

displayLibrarianOptions displays the options for librarians

**FileStorage**

Declares vectors for readers, librarians, books, and copies

Read is a method to opens the four files and checks for errors

Write is used to alter the data files

readReaders is used to read in the reader.dat file in the proper format, uses the values read to declare variables such as total number of readers, stores in vector

readLibrarians reads in the librarian.dat file and reads in the data of the librarian users, stores in vector

readBooks takes in the books.dat file and stores the values in the proper vector

readCopies reads in the book info and the unique variables. Like the book it tells who is reserving the book and borrowed the book

writeReaders writes out the data of the readers to readers.dat, uses the printReaders method

writeLibrarian writes the librarian.dat info to the file, uses printLibrarian method

writeBooks writes the output to the books.dat using the printBook method

writeCopy writes to the copy.dat file displaying all copies and their info

removeCopy allows a user to remove a copy from the file that will also remove from books if it is the last remaining copy

**Librarian**

This method is simply various constructors to create the user librarian and also contains a print method

**Reader**

Constructors, getters, and setters that allow for configuring a reader user with the variables penalties, number of borrowed copies, number of reserved copies and keep times

returnCopy allows the reader to return a copy of a book

Constructors for vectors such as copy that contains all borrowedCopies

Can create a reservation and cancel a reservation

Print method to display all important data

**User**

Constructors to create a user, username and password

setPasswordNew and setPasswordChangeallows user to change their password

setUserType determines what kind of user it is, librarian, student, or teacher

getters and setters for all variables

In the project, a specific example of using templates for other classes to obtain the User class methods. This was vital because it almost all of our classes there was a need to use the User class methods such as the original constructor that allowed for a user to be created or when a method was required to use a setter method from user to alter the account. We were able to complete masking for the password but unfortunately it only covers Windows OS because it utilizes conio.h. We did not use overloading for the print statements although we certainly could have. Instead we simply created different print methods for each of the classes.

The highlights of our project are probably the methods of the users and the file reading/writing. Altering the files was very challenging and unfortunately, we still were not able to completely debug every piece of our code. While testing if one is to run a borrow or reserve method it will incorrectly alter the output of the files which will cause corruption throughout the entire code.

