**Library**

This program is a simulation of a basic library system using files to store user credentials as well as books. Users would be given different roles that allow them to have access to different functions such as searching up a book, or adding a book.

**Classes**

**User:**

**functions:**

**Default constructor**: constructor without arguments.

**Overloaded constructor**

**Accessor and mutator functions:** for userNum, username, password, and role.

**validateUser():** takes in 3 string arguments. Opens the file: “credential.txt”. Using strtok, splits the string on the format recnum|role|username|password. Then if username and password match the inputted username and password then stores the role of the user. Has file exception to catch whether “credential.txt” exists.

**doesBookExist():** takes a string argument and point int argument. Opens the “books.txt” file. Reads in the line and using strtok finds position 3 which is the isbn. If the isbn matches the string argument which is an isbn then changes the bool variable bookExist to true. The bookExist variable determines whether the book is in the books.txt file.

**searchBook():** takes a string argument. Opens the books.txt file. Checks whether a book exists or not by reading in books.txt and using strtok to find the position three which is the isbn of the book. It then matches whether the isbn in books.txt is the same as the inputted string argument isbn. If it matches then returns a message telling the user that the “Book exists”. It then checks the 6th position which is the checkout status and returns a message saying whether the book is available for checkout or not. Has file exception to catch whether “books.txt” exists.

**Librarian:**

Subclass of User  
**functions:**

**Default constructor**: constructor without arguments.

**addBook()**: prompts user to enter book info such as the title, author, isbn, year of publication, and publisher. Uses the function doesBookExist() from the User class to check whether the book exists. If it doesn’t yet exist it adds the inputs to the string newBookRecord then writes newBookRecord into the book text file. Else if book already exists it’ll give user a messaged that the book already exists to avoid a redundant entry.

**Customer:**

Subclass of User

Since the customers only function is searchBook() which it inherits from the User class it doesn’t have any functions.

**Main:**

The login system creates a user variable type User() and checks the validation of user using the validatedUser() method and stores the result in the int variable validated. If validated passes it logins the user based on the role. Else it’ll give the user two more tries before locking the user out. If the users rold is a customer the user will be given a menu with two options: check book status and exit. The librarian is also given a menu with the same options and with an additional option: add a book. Both of these menus are done with switch case.