# Lab 8

## Library Books

Skills Required

* Create and use classes
* Exception Handling, Read and write Files, Work with arrays and vectors, Create Functions, Include Headers and other files, Loops (while, for), Conditional (if, switch), Datatypes, assignment, etc.

**Assignment**

You work at the Computer Science 201L Library, and your boss just bought a bunch of new books for the library! All of the new books need to be cataloged and sorted back on the shelf.

You don’t need to keep track of what customer has the book or not, just whether they are checked in or out.

**Objects**

You need to create a class called LibraryBook. It needs to have the following member variables, stored privately: string title, string author, and string ISBN. You also need to have a Boolean to store its status whether it’s checked out or not. Call this variable checkedOut.

Create a header file and a .cpp file for the LibraryBook class. Call it LibraryBook.cpp and LibraryBook.h

For functions, this library book class needs to have:

* A default constructor setting all strings blank and the checked out status to false
* A constructor setting all strings through the parameters (title, author, ISBN), and setting the checked out status to false.
* “Getters” to get the values from the private variables. (So getTitle(), getAuthor(), getISBN())
* A checkOut() function to change the status so it is checked out
* A checkIn() function to change the status so it is checked in (or NOT checked out)
* A isCheckedOut() function to return the status of the book

**More Information and File Structure**

The books.txt file has a list of all of the books at the library. The books are listed with the title on one line, the author on the next, then the ISBN on the next. For simplicity, all three are stored as strings in the class, including the number. Using the file streams, you’ve learned how to read in one word or number at a time. To read in an entire line, use the getline() function. For example:

string str; // string variable

ifstream fin(“file.txt”); // input file stream

getline(fin, str); // read one line, store in variable

This will read in the while line (not the first word) from the file and store it in the variable str, which is a string. Use this example to read in the title, author, and ISBN from the file.

From there, you can pass in your data in as parameters into the LibraryBook object. For example:

string title = “title”;

string author = “author”;

string ISBN = “123”;

LibraryBook myBook(title, author, ISBN);

STOP HERE! Make sure your program works correctly at this point before continuing.

Next, in the isbns.txt file, you have a list of ISBN numbers, one per line. If the ISBN number appears in this file, it has passed through the barcode scanner on the computer to be checked in or out. You can read these in with the normal style you are used to (for example: string theISBN; fin >> theISBN; ). You can then either use the LibraryBook’s checkIn() or checkOut() function to tag it as either there or not there. To determine which one to use, call the isCheckedOut() function to find out if it’s at the library or not, then use that Boolean to determine which function you call.

STOP HERE! Make sure your program works correctly (again) at this point before continuing.

Finally, you need to print the report your boss needs. The report should be named checkedout.csv (NOT .txt) and will be in a comma-separated values format. It should begin with the line:   
“Title” , “Author” , “ISBN”

The quotation marks should be included in the file. (Printing a quotation mark within a string is demonstrated below.) It should then have one book’s information on each line, with each field in quotation marks and a comma between them, outside the quotation marks. The file will list only the books that are checked out. You will need to iterate through the loop and print out only the books that are checked out, sending them to the file in the specified format. (You may want to write a function that takes an ostream passed by reference and a LibraryBook passed by const reference as parameters.) So your output file should look something like this:

“Title”, “Author”, “ISBN”

“Code: The Hidden Language of Computer Hardware and Software”, “Charles Petzold”, “1735611319”

“Interconnection Networks”, “Jose Duato”, “1558608524”

“Understanding Computers: Today & Tomorrow, Comprehensive”, “Deborah Morley”, “1423925211”

“Logic and Computer Design Fundamentals”, “M. Morris Mano”, “1131678493”

Why do we output it in that style? Because we can open that file with just about any spreadsheet, such as Excel or LibreOffice Calc, and it will place each item into its own column, with the items in the first line as column headings. Try it!

**Programming notes:**

Whether to sort the books by ISBN before dealing with the checkins and checkouts is something of a judgement call. Our library is small, and performance isn’t critical here. A linear search will work well enough. For a larger library, we’d definitely want to sort the books by number so we can search them quickly. At some point, we’d probably want to consider a data structure more flexible than an array or vector for handling our data.

In your source code, a quotation mark usually indicates the beginning or ending of a string variable. What if you want the quotation mark as part of the string? Then you need to use the backslash as an escape character, to tell the compiler that something is part of the string and not a delimiter:

cout << "This is in \"quotes.\"" << endl;

So you’ll write code something like this:

fout << "\"Title\",\"Author\",\"ISBN\"\n";

to write the first line in the file.

You could also declare a variable:

char quotemark = '"';

And use it instead, adding it to the front or back of various strings as needed:

out\_str = quotemark + SomeBook.GetTitle() + quotemark + ',' + ... (other stuff)

**Submit your assignment**

1. Update your files on GitHub, being sure to sync your final commit so everything’s updated properly. OR,
2. delete the contents of the Debug folder(s), zip up your project, and upload your zipfile to Canvas.