Assignment #2 – Classes and File I/O in C++
Due: Sunday, 04/30/17, 11:59pm

Grading: For each programming assignment, you are graded by explaining and demoing your code to a TA. **You must demo your program BEFORE the next assignment is due**, and if you fail to do so, you will automatically lose 50 points! **Your job is to convince the TA that your program works correctly, i.e. show your TA how to use/break your program**☺

(90 pts) **Problem Statement:** We will simulate a very simple online library system. You will write a library class that has hours of operation, a cart for holding books, a librarian, and a patron**,** who use the cart for storing books. In this assignment, you need to have at least three files to support the **database of books**, the **librarian/patron login information**, and the **books checked out by specific patrons**. Your program will have the following functionality.

- Any user can find out the library hours of operation for the week or a specific day.
- Patrons with a valid id can view all the books in the library or search the database of books by title, author, or year. In addition, they can check in and out books using the cart, as well as view all the books currently checked out.
- Librarians with a valid id can change the library hours of operation, add new books to the library using the cart, remove old books from the library, and view books that are checked out.

Below is a guideline for your class breakdown. You can make decisions about the file setup, additional functions and parameters you may need, and the user interface. However, **you must continually ask users if they want to view hours or login as a librarian or patron, until they want to quit the program**. Have fun and be creative! ☺

librarian.h

```
class librarian {
   private:
     string name;
     int id;
   public:
     //You need constructors, accessors, mutators, and
     //any other functions (use const where necessary)
     …
     void change_library_hours();
     void add_new_books(const cart &);
     void remove_old_books();
     void view_all_books_checked_out();
     void view_specific_book_checked_out(string);
```

```
};
```

patron.h

```
class patron {
   private:
     string name;
     int id;
     int books_checked_out;
   public:
     //You need constructors, accessors, mutators, and
     //any other functions (use const where necessary)
     …
     void check_out_books(const cart &);
     void check_in_books(const cart &);
     void view_my_books_out();
};
```

structs.h

```
struct hours {
   ??? begin_time;
   ??? end_time;
};
struct book {
   string title;
   int num_authors;
   string *authors;
   int year;
   int copies;
};
```

cart.h

```
class cart {
   private:
     book *books;
     int num_books;
     void resize_books(int);
   public:
     //You need constructors, accessors, mutators, and
     //any other functions (use const where necessary)
     …
     void add_to_cart(const book &);
     void display_books();
     void empty_cart();
};
```

library.h

```
class library {
   private:
     hours week[7];
     int num_books;
     book *books;
```

```
        cart c;
        librarian l;
        patron p;
    public:
        //You need constructors, accessors, mutators, and
        //any other functions (use const where necessary)
        …
        void display_daily_hours(string);
        void display_weekly_hours();
};
```

Requirements:

- You are required to have constructors and the appropriate use of const and the Big Three for every class.
- All member variables need to be private, and you must have accessor and mutator functions for each member in the class.
- Appropriate behavior for each object to correctly implement the library functionality.
- You must have a Makefile, .h and .cpp for each class, and a driver file with the main function.
- Your program must not have a memory leak!!!
- Keep your functions under 20 lines!!!

Extra Credit: 10 pts

Make an error handling library that contains functions for making sure input is a valid positive integer, valid integer, valid positive floating point number, valid floating point number, etc. Use your library to handle all possible errors in this program and future programs.

(10 pts) Program Style/Comments

In your implementation, make sure that you include a program header in your program, in addition to proper indentation/spacing and other comments! Below is an example header to include. Make sure you review the style guidelines for this class, and begin trying to follow them, i.e. don't align everything on the left or put everything on one line! http://classes.engr.oregonstate.edu/eecs/spring2017/cs162-001/162_style_guideline.pdf

```
/*************************************************
** Program: library.cpp
** Author: Your Name
** Date: 04/26/2017
** Description:
** Input:
** Output:
*************************************************/
```

Electronically submit your C++ program (**.h, .cpp, Makefile, and text files**, not your executable!!!) and **as a tarred archive** by the assignment due date, using TEACH.