CSCI 135 Section 2
Programming Project 2

Due: Friday, November 21st, 2014, 11:59 pm

1 Introduction

In this project you have to combine your skills to create an Library Management System. Some tools that you should use are file I/O, arrays and classes. Note that this project should be completed in an Object Oriented way, and I suggest you first think about what sort of classes you will have and then run them by me before starting to code them up.

1.1 Programming Rules

Under "Course Information" on Blackboard (for CSCI 135), you will see a document called "Programming Rules". Please read it and make sure you follow the rules described therein.

1.2 Commenting

Note that you will lose points if you do not comment your code. Apart from explaining what a block of code is doing, you should also write a brief description of your program at the top of your `.cpp` files using multiline comments (`/* ... */`). Meaningful comments will also assist the instructor in understanding your code easily.

1.3 Design

For this project, first think of the classes you will need, and how you will store objects of those classes in memory during program execution.

You will also need to write some functions to handle conversion from space-separated strings to underscored-separated strings, plus other functions to handle the dates.

2 Library Management System

ABC University has asked you to create a Library Management System for them. Currently, the library keeps all its information in two text files and they view that information in Linux's *gedit*. Your task is to make a program that lets them search for specific records using some search parameter, and displays them in a user-friendly format. Your program should also provide an easy way to update the information.

File formats

ABC University's library uses two text files: one to maintain their collection of books, and the other to maintain circulation (i.e., books borrowed by members).

Each line of the the books' collection file, called "books.txt", has one record. The format of the record is as follows: ID, Title, Author, Category, Number of Copies. For simplicity, assume each book falls under exactly one category. An extract from books.txt looks like this:

```
11 Absolute_C++ Walter_Savitch Computer_Science 3
84 Programming_Languages Louden Computer_Science 1
```

and so on.

The second file, "circulation.txt", contains records of borrowed books. Each line contains the Book ID, Member ID, Issue Date, Due Date. The dates are stored as an 8 digit number, starting with the year, then the month, and ending with the day. For example, the date 11/9/2014 would be written as `20141109`. An extract from circulation.txt looks like this:

```
11 914 20141101 20141115
11 32 20140912 20140926
84 667 20141011 20141018
99 113 20141011 20141018
```

and so on.

Note that while the strings in the text files contain underscores in place of spaces, you should always display spaces when printing information in your program.

3 Your Task

Your task is *not* to create a new storage mechanism. Instead, you have to write a software that reads and edits these files, so that the university staff does not have to do it manually. Specifically, you have to implement the features listed below. The program should have a main menu, from where the user can perform each task.

1. **Add Book to Collection**

   The program should ask the user for data (title, author, category, copies) regarding this new book, and do one of the following:

   - If the book does not already exist in books.txt, add it as a new record in the file.
   - If the book already exists in books.txt, update the number of copies. If the category specified is different from the one in the file, do not update the number of copies; instead, print an error message and return to the main menu. However, if the ID is different, your program should ignore the new ID.

Note that the user can use spaces in the book title, author name and category fields, but your program should convert them to underscores before storing in the file.

2. **Edit Books' Details**

   The user should be able to edit the details of records in books.txt. Your program should ask the user for the ID of the book whose details need to be edited, then show the current details, and then ask the user which field needs to be edited. When the user specifies the field (0 for ID, 1 for Title, 2 for Author, 3 for Category, and 4 for Number of copies), your program should ask for the new value and then save that new value in the file.

   After this, the program should ask the user if they want to edit another field for the currently selected book. If they say yes, repeat the above process, else return to the main menu.

   If the user updates the number of copies to 0, that record should be deleted from the file.

3. **Get List of Books Borrowed by a Member**

   Your program should ask the user for a Member ID, and then print a list of all the books borrowed by that member. Note that even though the file circulation.txt contains only the Book ID, your program should look up that book's details in book.txt and then print them.

4. **Get List of Overdue Books**

   Your program should determine the current date, and then print a list of all overdue books and the members who borrowed them. In this task, you may only print the Book ID.

   The following code can be used to get the current date in C++:

```
time_t t = time(0);   // get current time in time_t struct format
struct tm * now = localtime( & t ); //convert it to a more useful format
int currYear = now->tm_year + 1900; //Extract year
int currMonth = now->tm_mon + 1; //Extract month
int currYDay = now->tm_mday; //Extract day
```

   Note that you have to include the file `<ctime>`. Since the dates in circulation.txt are stored as just a single number, you should think of a way to compare both formats correctly.

4 Submission

The format of the filename should be `<YOUR_LAST_NAME>_hw2.cpp`. Replace `<YOUR_LAST_NAME>` with your last name.

Now goto CSCI 135 on Blackboard, goto the "Programming Projects" section, and click on "Project 2". You will see an option to upload your files. Upload both files and click "Submit". If you have something to tell the instructor, enter it in "Comments".