CS 201, Spring 2017

Homework Assignment 1

Due: 23:59, April 10 (Monday), 2017

In this homework assignment, you are supposed to implement a library that contains multiple books. For each book, you are going to store a record in which you keep

1. The book title,
2. The name of its publishing house,
3. Its publishing year, and
4. Its author list, in which you keep the names of its coauthor(s).

In your implementation, you **MUST** keep the books in a dynamically allocated array. For each of these books, you **MUST** use another dynamically allocated array to keep its coauthor(s). Neither the number of books nor the number of coauthors in a book is known in advance.

Your system will have the following functionalities. The details of these functionalities are given below

1. Add a book
2. Remove a book
3. Add a coauthor to the book
4. Remove a coauthor from the book
5. Display all books
6. Show detailed information about a particular book
7. Find all books written by a particular author

Add a book: This function adds a book to the library system. The title of the book, its publishing house, and its publishing year are specified as parameters. In this function, the author list is not specified; the coauthor(s) of the book will be added later. In this system, book titles are unique (case insensitive). Thus, if a user attempts to add a book with an existing title, display a warning message and do not perform the requested action.

Remove a book: This function removes a book from the library system. The title of this book is specified as a parameter. If there is no book with the given title, display a warning message and do not perform the requested action. Note that this function also clears the author list of the specified book.

Add a coauthor to the book: This function adds a coauthor to the author list of a book. The title of the book to which the author is added and the author name are specified as parameters. In this function, you should take care of the following issues:

- If the book with the specified title does not exist in the library system, display a warning message and do not perform the requested action.
- All author names are unique (case insensitive) within the same book. Thus, if a user attempts to add an author with an existing name in the same book, display a warning message and do not perform the requested action. However, different books can be written by the same author (that is, different books can have the same author in their author lists).

Remove a coauthor from the book: This function removes a coauthor from the author list of a book. The book title and the author name are given as parameters. If there is no book with the specified name or if the author name is not in the author list of the specified book, display a warning message and do not perform the requested action.

Display all books: This function lists all books already found in the library system. The output should be in the following format. If there are no books in the library, display `--EMPTY--`

```
Book title, publishing year (for the 1st book)
Book title, publishing year (for the 2nd book)
. . .
```

Show detailed information about a particular book: This function displays all of the information about a book whose title is specified as a parameter. The output should be in the following format. If the book with the specified name does not exist in the library, display `--EMPTY--`

```
Book, publishing house, publishing year
Author name (for the 1st coauthor)
Author name (for the 2nd coauthor)
. . .
```

Find all books written by a particular author: This function lists all of the books written by the author whose name is given as a parameter. The output should be in the following format. If the specified author does not write any book, display `--EMPTY--`

```
Author name
Book title, publishing year (for the 1st book)
Book title, publishing year (for the 2nd book)
. . .
```

Below is the required public part of the `Library` class that you must write in this assignment. The name of the class **must** be **`Library`**. The interface for the class must be written in a file called `Library.h` and its implementation must be written in a file called `Library.cpp`. Your class interface file ("`Library.h`") should contain the following member functions. If necessary, you may also define additional public and private member functions and data members in this class. You can also define additional classes in your solution. For example, defining an additional `Book` class will make your implementation easier and clearer. **On the other hand, you are not allowed to delete any of the given functions or modify the prototype of any of these given functions**.

```
#include <string>
using namespace std;

class Library {

public:
    Library ();
    ~Library ();

    void addBook( string bookTitle, string publishingHouse, int publishingYear );
    void removeBook( string bookTitle );

    void addCoauthor( string bookTitle, string coauthorName );
    void removeCoauthor ( string bookTitle, string coauthorName );

    void displayAllBooks();
    void displayBook( string bookTitle );
    void displayAuthor( string authorName );

    // ...
    // you may define additional member functions and data members,
    // if necessary

};
```

NOTES ABOUT IMPLEMENTATION:

1. You ARE NOT ALLOWED to modify the given parts of the header file. You MUST use dynamically allocated arrays in your implementation. You will get no points if you use fixed-sized arrays, linked-lists or any other data structures such as `vector/array` from the standard library. However, if necessary, you may define additional data members and member functions.
2. Moreover, you ARE NOT ALLOWED to use any global variables or any global functions.
3. Your code must not have any memory leaks. You will lose points if you have memory leaks in your program even though the outputs of the operations are correct.
4. All book titles and author names are case insensitive. For example, "The Waves" and "THe wAves" should be considered as the same book.
5. The use of an additional `Book` class will make your implementation easier and clearer. Thus, we strongly recommend you to define this class and use it in your implementation. And if you decide to define the `Book` class, do not forget to implement its constructor, destructor, and copy constructor as well as do not forget to overload its assignment operator. Otherwise, you may encounter some unexpected run time errors.

NOTES ABOUT SUBMISSION:

This assignment is due by 23:59 on April 10, 2017. This homework will be graded by your TA Necmi Acarsoy (necmi.acarsoy at bilkent edu tr).

1. In this assignment, you must have separate interface and implementation files (i.e., separate .h and .cpp files) for your class. The file names should be "`Library.h`" and "`Library.cpp`". You should also submit other .h and .cpp files if you implement additional classes. We will test your implementation by writing our own main function. Thus, you should not submit any function that contains the main function.

   Although you are not going to submit it, we recommend you to write your own driver file to test each of your functions. However, you SHOULD NOT submit this test code (we will use our own test code). If you submit your own main function, you will lose some points.

2. You should put your "`Library.h`" and "`Library.cpp`" (and additional .h and .cpp files if you implement additional classes) into a folder and zip the folder (in this zip file, there should not be any file containing the main function). The name of this zip file should conform the following name convention: secX-Firstname-Lastname-StudentID.zip where X is your section number.

   The submissions that do not obey these rules will not be graded.

3. **Then, before 23:59 on April 10, you need to send an email with a subject line CS 201 HW1 to Necmi Acarsoy, by attaching this zipped file.**

   No hardcopy submission is needed. The standard rules about late homework submissions apply. Please see the course syllabus for further discussion of the late homework policy as well as academic integrity.

4. You are free to write your programs in any environment (you may use either Linux or Windows). On the other hand, we will test your programs on "`dijkstra.ug.bcc.bilkent.edu.tr`" and we will expect your programs to compile and run on the `dijkstra` machine. If we could not get your program properly work on the `dijkstra` machine, you would lose a considerable amount of points. Therefore, we recommend you to make sure that your program compiles and properly works on "`dijkstra.ug.bcc.bilkent.edu.tr`" before submitting your assignment.
5. This homework will be graded by your TA Necmi Acarsoy (necmi.acarsoy at bilkent edu tr). Thus, you may ask your homework related questions directly to them.