

### **Homework #5: Due May 8**

In this homework assignment, you will have to implement a searchable textbook database. This will allow the user to search for the author (or ISBN or publisher, etc.) for a book with a particular title, as well as finding all books by a particular author, among other things.

You will implement 2 classes: *Book* and *BookDatabase*.

The Book class must contain:

A string for the title and publisher, an integer for the year of publication, and a vector (or list, or set) of strings for the author. (You can choose whichever one you want).

The Book class will have the following member functions defined on it:

**Book(const string& title, const string& publisher, int yearOfPublication, string\* authors, int numAuthors);**

This is the book constructor. It will initialize all the member variables and also store all of the authors names in the container that will hold these names.

Accessor functions for all of the data fields, e.g. **string** getTitle(), etc.

**bool operator ==(Book& b1, Book& b2);**

This will return true iff the two Books have the same title.

(You can assume that the database will not contain 2 books with exactly the same title.

**bool operator < (Book& b1, Book& b2);**

This will return true iff the title of book1 is lexicographically before the title of book2.

**ostream& operator << (ostream& os, Book& b);**

This will print a representation for a Book. For example:

Title: Data Structures with C++ Using STL

Authors: William Topp and William Ford

Publisher: Prentice Hall  
Year: 2002

You may, of course, write as many private functions as you would like to help your implementation.

The BookDatabase class must have the following member variables:

A map whose keys are strings and values are Book objects, such that the key is a string representing the book's title, and the value is a Book object corresponding to that title.

For example for the Book object described above, the key would be "Data Structures with C++ Using STL" and the value would be the Book object containing that same string as a title, "Prentice Hall" as the publisher, 2002 as the year, and {"William Topp", "William Ford"} (However you want to represent this.) as the authors.

The declaration for that map should be:

```
map <string, Book> bookData;
```

The BookDatabase class should have the following operations:

**Book&** searchByTitle(**const string&** title);

returns a reference to the Book object that contains the given title.

**set<Book>** searchByAuthor(**const string&** author);

returns a set of Books by the given author. (This one is a bit tricky).

**set<Book>** publishedBefore(**int** year);

returns a set of Books that were published before a given year.

**set<Book>** searchByPublisher(**const string&** publisher);

returns a set of Books that were published by the given publisher.

`void addBook(Book& book);`

adds a new book to the database if it is not there already. (You will implement this by adding a new pair into the map where the key is *book.getTitle()* and the value is *book*). If it is there already, do nothing.

`void removeBook(const string& title);`

Remove the book with the given title from the database if it is there.

`ostream& operator <<(ostream& os, BookDatabase& bookDatabase);`

Prints a representation for a book database.

You may use the “books.txt” file that I have uploaded as your data, but you should add some more items to the file for testing purposes.

You must submit:

Books.h, Books.cpp, BookDatabase.h, bookDatabase.cpp, and useDatabase.cpp, which is a test program for the database.

You may use **ONLY** those functions that are part of the API on the course webpage. Unless you ask me first, you may not use any fancy map or set member functions unless you write them yourself.