RMIT International University Vietnam

Department of Information Systems

COSC2131 OOP in C++ - 2012, Semester B

Assignment 2 – Library System

(This assignment contributes 25% to the total mark)

Learning Objectives

The objectives of this assignment are to

- get more skills with OO features of C++ and operator overloading,
- use streams and file I/O,
- get skills with generic programming and STL, algorithms and function objects,
- and use the exception mechanism of C++.

Assignment Description

The software to be written for this assignment is a Library support system. The system will contain information about books and borrowers. The system will track the books that are overdue, book availability etc.

For this assignment you will **work in groups of two**.

Books

The system must store at least the following data about books:

- Title
- Edition
- A maximum of 5 authors
- ISBN code (must be unique)
- Publisher
- Year of print
- Number of pages
- Category (Business, marketing, finance, programming, database, etc.)

The information above is the general information about a certain title. Of course, the library can have more than one copy of each book. Therefore, the database must store additional information for each “instance”, each copy of a book:

- Barcode (must be unique)
- Borrowing status (in storeroom, in library, borrowed, lost, etc.)
- Number of times borrowed

- The date that is was borrowed
- The number of days it is borrowed this time

Note: In a later phase of the implementation, the library will connect a barcode scanner to the system, and the system will allow the librarian to use the scanner to input barcodes. Currently this needs to be done by hand.

Borrowers

The system must store at least the following information about borrowers:

- ID (unique)
- Name
- Type (student, teacher)
- Department (IS, BCOMM, UPP, etc.)
- Phone number
- Number of books borrowed in the past
- Number of books returned late in the past

Entering data

Your program must have features for the following data entry operations:

1. **Enter a new book.** The librarian can type all information about a book. Only basic error checking needs to be done (no maximum length, no restrictions in terms of allowable characters, etc.)
2. **Create new copies.** After entering the basic information, the librarian can add a number of actual copies. This involves entering the barcode and the status. The default status should be “In library” or “In storeroom” so the librarian just needs to confirms this. Typically, the librarian will enter 20 or 30 copies of books at once, so please make it easy to do.
3. **Enter a new borrower.** The librarian can type all information about a new borrower. Again, only basic error checking.

Everyday operations

The following features will be used very frequently:

1. **Borrow a book**. The librarian types the ID of the person who wants to borrow. After retrieving the information, the system shows the name of the borrower and the librarian must confirm the selection. Afterwards, the librarian types the barcode of the book to be borrowed. The system then shows the name of the book. It can then be chosen to borrow.
2. **Return a book**. The librarian types the barcode of the book that is returned. The status of the book is updated. If the book is returned late, the system shows a warning message saying how many days the book was returned late.
3. **Change the status of a book**. The librarian first has to specify which book it is about. This can be done in two ways:
    1. If the book is present (the librarian has the physical copy in his/her hands) the librarian can enter the barcode
    2. If the librarian doesn’t have the book (for example because it is lost) the author/title can be entered and the copy selected from a list.

Next, the librarian selects the new status and the database is updated.

Reports

The following reports are available to the librarian:

1. A **list with all books**, one book per line. One line shows information about a title, not about a particular copy! In other words: the list must not contain all copies of all books, but just one line for each book. Each line should show at least basic information, but also the number of copies in the library and the number of copies currently borrowed.
2. A **list with copies of books**, but filtered. Suggested filters are: currently borrowed copies, copies overdue, copies of a specified ISBN number.
3. A **list with borrowers**. One line shows at least basic information about a borrower, but also the number of book borrowed in the past and currently.
4. A **list with borrowers who have books overdue**. The librarian should be able to see the contact information of a borrower, so overdue borrowers can be contacted.

The list can be sorted as the librarian wishes. Lists with books can be sorted on at least name, author and number of copies. Lists with borrowers can be sorted on at least name and the number of books they have borrowed in the past.

Software aspects and Implementation requirements

There are two comments on the implementation of the application.

1. The data structure must be chosen carefully. The system will potentially store a huge amount of books and borrowers, so fast data structures and algorithms must be chosen.
2. It is required to use all C++ features in this assignment. Sometimes it will be necessary to think of a possible use of a feature. It is important not to try too hard and then implement your application in an unnatural way. In any case, the code should be understandable. (Don’t overload the + operator to do subtraction!)

Documentation

You are required to write an extensive report that shows how your program uses C++ language features (file I/O, classes, operator overloading, STL & algorithms, exceptions, etc.) For example, explain where you use exceptions in your program, and what it does in your program. Another example, explain what data file(s) you choose to use and how file I/O was implemented.

Your explanation must include how you used C++ features in your assignment, where you use them, why you use them, what is the importance of using them, what is the impact of using them, what would happen if you did not use them, etc.; all of which are explained in the context of your assignment.

A suggestion is writing your explanation in separate paragraphs per C++ feature.

In the documentation, do not write the definition of C++ concepts or the explanation of them. For example, you do not need to include the definition of inheritance and you do not need to explain what inheritance is.

A second thing that must be included in the documentation is your choice of data structures and algorithms. For example, why did you choose a particular data structure; do you think it is flexible, performs well, saves memory, etc.

Any other explanation or defense may be included in the documentation.

The documentation is to be included as a readme.txt file, included with the source.

Assessment

The weight of this assignment is 25% of the total mark for this course.

Discussion on the best use of language features or class design is encouraged, however only authentic submissions are acceptable. Plagiarism detection software will be used and if detected, plagiarism will be penalized as outlined in the Course guide.

As well as a correctly functioning executable, assignment markers will be looking for good use of object-oriented class design, correct use of C++ language features and the C++ standard library, and good use of comments to show your level of understanding.

Submission

To be submitted is a single zip file with the source code and project file, workspace file or Makefile. A good filename for the zip file is like “Nguyen_Kim_assign2.zip”.

Submissions comprise a report in HTML format, C++ code, and a `Makefile`. A softcopy of your work must be submitted to Blackboard by the deadline. You are required to adhere to the submission standards published on Blackboard.

A readme.txt file with the documentation is required to be submitted. It is to be included with the other source files in the zip file.

Plagiarism notice

This is an individual assignment. While discussion about program design, algorithms & data structures, C++ features and quality issues is encouraged, exchange of files and/or exchange of code are regarded plagiarism. Refer to the document “Referencing and Plagiarism in Programming Assignments” for details. Consult with your lecturer for advice on this issue if necessary.

Any copying of any code from any source is strictly forbidden. The submitted work must be 100% your own.