RMIT University Vietnam

COSC2131 - Programming using C++

Assignment 2 – Multimedia Library collections and services System

Assignment objectives

The aim of this assignment is to help students:

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

Overview

The software to be written for this assignment is a Multimedia library collections and services system. The system will contain information about multimedia collections such as DVDs, Blu-Ray, laser discs, and CD ROMs and borrowers. The system will track the multimedia collections that are overdue, their availability etc.

Submission

Zip your Assignment 2 C++ files + report in HTML format + `Makefile` and name the zipped archive with your id and name. For example, if your name is Nguyen Van Anh and your id is s1234567, then the zipped archives should be called Assignment2_s1234567_NguyenVanAnh.zip. Upload the zip file of the assignment 1 with report in the Google Drive and share that in Edit mode on or before **9.00 AM Wednesday** of **week 11. I.e. 25/12/2013**. Not submitting in this format will mean your submission will not be graded.

Late submission: Applications for extension should be via email to the lecturer at least 24 hours before the submission deadline. The duration of extension is decided by the lecturer and is no more than 2 days. Submissions 1 to 5 days late receive 10% penalty per day. Submissions more than 5 days late receive 100% penalty.

Assessment requirements: This is an individual assignment. The minimum penalty for plagiarism is failure for this assignment. If this means that a hurdle requirement is not met, the student fails the course. Please read http://blackboard.rmit.edu.vn/ → assignments section for further information on School plagiarism policy.

Additional Notes

- You are free to refer to textbooks, notes, work in study groups etc. to discover approaches to problems; however the assignment should be your own individual work.
- You will only be assessed on your own code, so the use of third party code or libraries is not allowed.

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 passages are regarded plagiarism and will be penalized to the maximum extent. Consult with your lecturer for advice on this issue if necessary.

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

Multimedia collections

The system must store the following data about each item:

- Type (CD/DVD/……..)
- Data type (images / audio/video….)
- Title
- Version
- Identification code (must be unique)
- Publisher
- Year of recorded/burned
- Duration (running time in minutes)
- Category (Engineering and Technology, History, Film and media studies, Psychology, Images, photographs& maps. General, Business and Economics….)

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

- Barcode (must be unique)
- Borrowing status (in storeroom, in library, borrowed, lost, etc.)
- Number of times borrowed
- The date it 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 (IT, BCOMM, etc.)
- Mobile number
- Number of items borrowed in the past
- Number of items returned late in the past

Entering data

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

1. **Enter a new collection.** The librarian can type all information about a collection. 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 item 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 collection**. 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 particular item to be borrowed. The system then shows the name of that collection. It can then be chosen to borrow.
2. **Return a collection**. The librarian types the barcode of the item that is returned. The status of the item is updated. If the item is returned late, the system shows a warning message saying how many days the item was returned late.
3. **Change the status of an item**. The librarian first has to specify which item it is about. This can be done in two ways:
    a. If the item is present (the librarian has the physical copy in his/her hands) the librarian can enter the barcode
    b. If the librarian doesn't have the item (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 collection**, one item 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 collections, but just one line for each item. Each line should show at least basic information, but also the number of copies in the library and the number of copies currently borrowed.

    Note: The user must have a choice to list the collections based on type, category, and data type.
2. A **list with copies of items**, but filtered. Suggested filters are: currently borrowed copies, copies overdue, copies of a specified identification number.
3. A **list with borrowers**. One line shows at least basic information about a borrower, but also the number of items borrowed in the past and currently.
4. A **list with borrowers who have 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 collections 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 collections 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 collections 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.

COSC2131 – Programming Using C++
ASSIGNMENT 2: Multimedia Library collections and services System
MARKING GUIDE

Items	Check if completed	Possible marks	Actual marks
1. Report [10 marks] • Introduction & Reference • How to build/install and run the program • Overview of the implementation • Data structures • Additional information for marker		 2 2 2 2 2	
2. Data definition [10 marks] • Class design (header files) • Definition of member function		 5 5	
3. Data Entry [15 marks] • Enter a new collection • Create new copies • Enter a new borrower (no duplicates)		 4 5 6	
4. Everyday operations [15 marks] • Borrow a collection (bar code reading) • Return a collection + late return • Change the status of an item		 6 6 3	
5. Reports (output) [18 marks] • A list with collection • A list with copies of items • A list with borrowers • A list with borrowers who have overdue		 4 5 4 5	
6. Reading and writing from/to files [8 marks]		8	
7. Additional features [22 marks] Interactive menu options Interactive output (Print, download/save reports) Reading and writing to → Databases		 6 10 8	
Total possible		100	

* If you have completed the listed requirements (1, 2, 3, 4, 5, and 6) then the maximum mark for this assignment **will be 78 or DI**. So in order to get HD, you must **complete all requirements. Make sure that your** code run at least two different platforms namely, UNIX/LINUX and Windows.

Comments/ Feedback:
