

Lab 1

For this lab we are going to design a system to Manage loans from the local public library. For this we will need the following entities, plus collections for each of the entities: Patrons, Books and Loans

The data for a Book will contain at least the following:

1. Author
2. Title
3. ISBN Number
4. Library ID number
5. Cost
6. Current Status (In, Out, Repair, Lost)

You may add other data needed for your implementation as well as you will need accessor and mutator functions for the data.

The data for a Patron will contain at least:

1. Name (e.g. Fred Smith)
2. ID number (6 digits e.g. 123456)
3. Fine Balance
4. Current # of books out

You may add other data needed for your implementation as well as you will need accessor and mutator functions for the data.

The data for a Loan (The transaction entity) will contain at least the following:

1. Loan ID
2. Book ID
3. Patron ID
4. Due Date and Time
5. Current Status (overdue, normal)

You may add other data needed for your implementation as well as you will need accessor and mutator functions for the data.

For the collections of each of the Patrons and Books Classes identified above you will need to include the ability to:

1. Add
2. Edit
3. delete
4. Search/Find based on appropriate criteria
5. Print a list of all entries in the collection
6. Print the details for a single entity (do a find first)
7. Other methods you may identify

For Loans you will need:

1. Check Out a book (update book and patron info as well as add a loan)
2. Check in a book (check for fines and update patron and book info and delete loan)
3. List all overdue
4. List all books for a particular patron
5. Update loan status based on system clock
6. Re-Check a book
7. Edit a loan
8. Report lost (update book and charge patron book cost as well)
9. Other methods you may identify

You will need to verify the following

1. Before borrowing a book, make sure Patron has no overdue books and that total book out will be ≤ 6 including new borrow
2. When checking a book in, determine if fines are owed
3. Reporting a book as lost records the cost of the book to the patron's fine balance
4. For Loans Add = Borrow a book
1. Delete = Return a book
2. Edit = Re-check
5. Also, will need a PayFines (in Patrons)
3. Report Lost (in Loans but will have to update books and patrons)
4. Print a list of overdue books with patron info (in loans but will have to update
5. books and patrons)

You will need to provide an appropriate menu system that can be multi-level if you like.

Do not attempt to provide card catalog services for allowing patrons to search for books. You may assume each book has a unique acquisition number, and you may use these numbers to refer to books borrowed and returned.

You will need to load and store the data. This can be done automatically when the program starts and ends. You should also want to store after an add, delete, or edit to make sure changes to the data are preserved.

You can assume the following Loan period is 10 days with an additional recheck of 10 days (1 recheck only)

A max of 6 books can be out to a single patron at a time

Fine rate is \$0.25 per day (24 hour period)

For this design you will need to turn in the following:

A diagram set consisting of:

1. A title page with your name, assignment, course, and title
2. a single class diagram showing only the relationships between the entities
3. a set of six individual class diagrams showing the attributes and methods for each of the classes in #2
4. Step by Step pseudo code algorithms for every method defined in every class in the diagram from #2. You do not need to provide pseudo code for simple accessor and mutator functions (i.e., sets and gets)
5. A 1-2 paragraph report about your design experience.

All these items should be gathered, in order, in a single PDF File that you will turn in on Canvas

NOTE: This assignment is for the design only

Nothing you turn in should look like C/C++ code