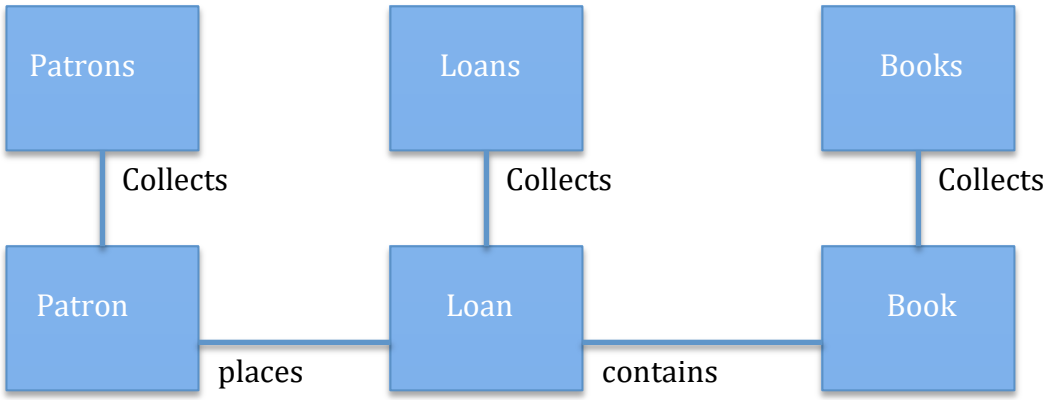
CSCE 1040.001/002

Homework 2

BY: Zachary Warren

Class Relationships



Class Contents

Patron
Name (string)
ID (int)
Fine Balance (int)
Num books out (int)
Constructors
Sets/Gets variables
checkIfTheycanRecheck
Print

Loan
Loan ID (int)
Book ID (int)
Patron ID (int)
Times Rechecked (int)
Due date and time (time_t)
Current Status (int)
Constructor
Sets/Gets variables
Print

Book
Author (string)
Title (string)
ID (int)
ISBN (string)
Cost (float)
Current status (int)
Constructor
Sets/Gets for variables
Print

Patrons
count (int)
patronList (Vector)
iterator (Patron)
nextAvaiableID (id)
Constructor
Get, inc, dec count
Add Patron
Delete Patron
Find Patron by ID
Find Patron by name
cleanup
Print Patrons
Load Patrons
Store Patrons

Loans
Count (int) iterator (Loan)
loanList (Vector)
NextAvaiableID
Constructor
Get, inc, dec count
Add Loan
Delete Loan
Check Overdue
Recheck
Find by ID
Find by book and patron ID
Find by patron ID
Find by BookID
cleanup
Print Loans
Load Loans
Store Loans

Books
Count (int)
bookList (Vector)
iterator (Book)
NextAvaiableID
Constructor
Get, inc, dec count
Add Book
Delete Book
Find Book
Find BookID
Find Book by title
Find Book by
cleanup
Print Books
Store Books
Load Books

Function Pseudo Code (not all defined – examples only)

For this design there would be pseudo code for at least 17 methods

Patron's methods (No Pseudo code changed)

Add Patron ()

Prompt user for ID
Prompt user for name
Create Patron Object with information form the user
Populate Object
Add object to collection

Edit Patron

Prompt user for ID
Prompt user for what they want to edit
Call Find Patron
Call Sets for Patron

DeletePatron

Prompt user for ID
Call Find Patron
Checks if Find Patron returned a -1
    Call the destructor for Patron
    Move non-deleted Patrons over by 1 (leave no blanks in Patron's vector)
    Print success

Find Patron (ID)

For lop starting at 0 and going till count-1
    If statement to see if ID's match
        When match return index
    Else
        Print error
        Return -1

PrintPatron(ID)

Call Find Patron
Checks if Find Patron returned a -1
    Print to terminal Name, ID, Fine Balance, and number of books out

PrintPatrons()

For loop starting at 0 and going till count -1

Print to terminal Name, ID, Fine Balance, and number of books out

Pay Fines

Prompt user for ID
Call Find Patron
Checks if Find Patron returned a -1
Output how much they owe in fines
Output / input how much they want to pay today
Put the subtracted value into a temp int
Call set Fine Balance to the temp int

Read Patrons

Create Ifstream object
Open file
Create Patrons object
While loop checking if something was read in each line
Creating Patron object
Input data into Patron object
Increase count
Close file

Write Patrons

Create Ofstream object
Open file
For Loop from 0 to count-1
Writing data to the file
Close file

Loan's Methods

AddLoan

Prompt user for book ID
Prompt user for Patron ID
Create a new loan in loans with information from the user
if statement that checks if the patron has more or equals to 6 books checked out (including the new book) and have no overdue books

if Parton has fines greater than 0
call Payfines
Set due date and time
Set times Rechecked to 0
Update data for book and patron

DeleteLoan

Prompt user for loan ID
If Patrons fines are greater than 0
Output that then need to pay their fines
Update data for book and patron
Call delete method in loans for loan

EditLoan

Prompt user for loan ID
If statement checking if the Patron has already rechecked the book before
if statement that checks if the patron has more or equals to 6 books checked out (including the new book) and have no overdue books
if Parton has fines greater than 0
call Payfines
Set due date and time
Set rechecked to 1
Update data for book and patron
Else
Output if Patron has to many books out or has an overdue book
Else
Output error statement

UpdatesStatus

Prompt user for loan ID
Call get current status to Update current status of loan based on system clock

ReportLost Book

Prompt user for loan ID
Update current status of book to lost
Update Patron's Fine Balance to the price of the book

Print LoanedBooks

Call print books

Print Loan

Output data from loan in a suitable format to the user

Print Overdue

For loop the goes from 0 to count -1

    Checks if the loan at the index is overdue

        Prints data for the overdue book

Read Loans

Create Ifstream object
Open file
Create Loans object
While loop checking if something was read in each line
    Creating loan object
    Input data into loan object
    Increase count
Close file

Write Loans

Create Ofstream object
Open file
For Loop from 0 to count-1
    Writing data to the file
Close file

Book's Methods

AddBook

Prompt user for book ID
Prompt user for Title
Prompt user for Author
Prompt user for ISBN
Prompt user for cost
Create a new book in books with information from the user
Set Status to 0(in)

DeleteBook

Prompt user for ID
Call Find Book
Call destructor for book of the index found

EditBook

Prompts user for book ID
Prompts user for what they would like to change
Calls the set of the function required

UpdatesStatus

Prompt user for book ID
Prompt user for desired status
Calls set current status and passes the desired status

Print Books

For loop that goes from 0 to count-1

    Outputs book of index in correct format

Print Book

Prompts user for book Id
Calls Find book with ID
Checks if Find Book returned a -1
    Prints book of that index in desired format

FindBook(Book Id)

For loop the goes from 0 to count -1

    Checks if the provided Book Id is equal to the Id of the book form the index

        Returns index

    Else

        Print error

        Return -1

Read Books

Create Ifstream object
Open file
Create Books object
While loop checking if something was read in each line
    Creating book object
    Input data into book object
    Increase count
Close file

Write Books

Create Ofstream object
Open file
For Loop from 0 to count-1
    Writing data to the file
Close file

Report:

This assignment took me about 4 hours to complete. I started this assignment the 21th of September because of tests in the week I started later than I would have liked. I found some aspects of this assignment to be easy but also it has more challenging parts. The easy parts for me where the relationship chart and class contents of the single entity classes. I found the vector classes and pseudo code to be the hardest, along with the large number of functions I find it hard to write the English summed up version of code.

I learned a lot about how to write pseudo code, because in my high school classes when we wrote outlines of code it was usually the methods and classes. I also learned that even though I try to manage my time well, I need to focus on splitting up assignments throughout weeks. One of my main troubling parts of the assignment, was thinking ahead to write down every function and what they would do before any coding. The only comments I have is I have a hard time thinking of missing or useful functions when I'm not actively coding the assignment