48024 Applications Programming
Assignment 1

Topics: OO Design, Standard Patterns, Lists
Learning Outcomes: This assessment task addresses the following subject learning objectives (SLOs): 1, 2 and 3
Due date: 11:59PM Monday the 18th of September
Weight: 30%

1. Individual work

All work is individual. You may discuss ideas, approaches and problems, but you should write every line of code yourself except for code copied from the lecture notes, lecture code or lab code. You MUST NOT let another student see your solution code, and you MUST NOT look at another student's solution code. More information about Academic Misconduct can be found at:
http://www.gsu.uts.edu.au/rules/student/section-16.html

2. Specification

Library Management System '85

Here at the NSW Institute of Technology we stay abreast of the latest advances in technology. As part of our ongoing process of

modernisation, the Library has purchased, at great expense, a DEC VT220 minicomputer and a DEC VAX 11/780-5 mainframe, running the brand new SunOS 1.2. To tie this together, we need a state of the art computerised

library management system to replace our aging index card system. The new system will be written in the very early prototype programming language "Oak" (rumours suggest this name is only temporary – please check the bulletin board for regular updates). This requirement is urgent, as the index card system has recently become the home of a large silverfish colony that is threatening to consume building 4.

This document details the requirements of the new system.

Now in a more readable font…

The Library System will consist of two main components, an administrative component, and a catalogue. The administrative section will allow, through text-based menus, the addition and removal of patrons, the addition and removal of books in the catalogue, the display of patron records and the display of a patron's 3 favourite books. The catalogue will store a list of all the books in the library, a list of the genres of those books, a list of all the authors with books in the library and a list of the books on loan. The catalogue will also allow display of the books in a number of ways, including by availability, genre and author. Finally, the catalogue will also handle the borrowing and return of books, and the placing of holds on books.

Each patron record will include the patron's name, their chosen ID number, which books they are currently borrowing and their full borrowing history.

Each book record will include the book's title, its author and its genre.

An aside...

While reading the the first part of the specification, you will notice there is a lot going on.

- *How many functions did you identify?*
- *How many classes did you identify?*
- *What are the fields in each class?*
- *How many goals did you identify?*
- *How many patterns did you think of that might be applicable?*

This assignment will be challenging and you will probably want to manage your time well.

- *How long do you think it will take you to code the functions?*
- *How long do you think it will take you to code each goal?*

A good rule of thumb is to think of an estimate, and then multiply that number by 3 or 4!

To manage your time well, you may need to figure out which parts of the assignment you can start early.

- *Which parts can you start now?*
- *Which parts can you start in week 6?*

If you complete parts in the same week that you learn the topics (while they are fresh in your mind), they will take less time to complete.

The User Interface

Below is a sample I/O trace. The red text indicates user input (you are not expected to print colour text). Not every conceivable scenario is shown below and you should submit your code to PLATE to see what specific scenarios are tested. You should also implement your solution in the same order as PLATE's test cases so that you can receive incremental feedback and marks as you progress.

```
Welcome to the Library! Please make a selection from the menu:
1. Explore the catalogue.
2. View your patron record.
3. Show you favourite books.
4. Enter Admin Mode.
X. Exit the system.
Enter a choice: 4
Welcome to the administration menu:
1. Add a patron.
2. Remove a patron.
3. Add a book to the catalogue.
4. Remove a book from the catalogue.
R. Return to the previous menu.
Enter a choice: 1
```

```
Adding a new patron.
Enter a new ID: 42
Enter the patron's name: Luke Mathieson
Patron added.

Welcome to the administration menu:
1. Add a patron.
2. Remove a patron.
3. Add a book to the catalogue.
4. Remove a book from the catalogue.
R. Return to the previous menu.
Enter a choice: 3

Adding a new book.
Enter the title of the book: Effective Java
Enter the author's name: Joshua Bloch
Enter the genre: reference
Added Effective Java to catalogue.

Welcome to the administration menu:
1. Add a patron.
2. Remove a patron.
3. Add a book to the catalogue.
4. Remove a book from the catalogue.
R. Return to the previous menu.
Enter a choice: R
Welcome to the Library! Please make a selection from the menu:
1. Explore the catalogue.
2. View your patron record.
3. Show you favourite books.
4. Enter Admin Mode.
X. Exit the system.
Enter a choice: 1
Welcome to the Catalogue! Please make a selection from the menu:
1. Display all books.
2. Display all available books.
3. Display all genres.
4. Display books in a genre.
5. Display all authors.
6. Display all books by an author.
7. Borrow a book.
8. Return a book.
9. Place a hold.
R. Return to previous menu.
Enter a choice: 1

The Library has the following books:
Joshua Bloch, Effective Java

Welcome to the Catalogue! Please make a selection from the menu:
1. Display all books.
2. Display all available books.
3. Display all genres.
4. Display books in a genre.
5. Display all authors.
6. Display all books by an author.
7. Borrow a book.
8. Return a book.
9. Place a hold.
```

```
R. Return to previous menu.
Enter a choice: 3

The Library has books in the following genres:
reference

Welcome to the Catalogue! Please make a selection from the menu:
1. Display all books.
2. Display all available books.
3. Display all genres.
4. Display books in a genre.
5. Display all authors.
6. Display all books by an author.
7. Borrow a book.
8. Return a book.
9. Place a hold.
R. Return to previous menu.
Enter a choice: 5

The following authors have books in this Library:
Joshua Bloch

Welcome to the Catalogue! Please make a selection from the menu:
1. Display all books.
2. Display all available books.
3. Display all genres.
4. Display books in a genre.
5. Display all authors.
6. Display all books by an author.
7. Borrow a book.
8. Return a book.
9. Place a hold.
R. Return to previous menu.
Enter a choice: 7

Enter a valid patron ID: 42
Enter the title of the book you wish to borrow: Effective Java
Book loaned.

Welcome to the Catalogue! Please make a selection from the menu:
1. Display all books.
2. Display all available books.
3. Display all genres.
4. Display books in a genre.
5. Display all authors.
6. Display all books by an author.
7. Borrow a book.
8. Return a book.
9. Place a hold.
R. Return to previous menu.
Enter a choice: 2

The following books are on the shelf:

Welcome to the Catalogue! Please make a selection from the menu:
1. Display all books.
2. Display all available books.
3. Display all genres.
4. Display books in a genre.
```

```
5. Display all authors.
6. Display all books by an author.
7. Borrow a book.
8. Return a book.
9. Place a hold.
R. Return to previous menu.
Enter a choice: 8

Enter a valid patron ID: 42
Luke Mathieson has the following books:
Books currently borrowed by Luke Mathieson:
Joshua Bloch, Effective Java
Enter the title of the book you wish to return: Effective Java
Effective Java has been returned.

Welcome to the Catalogue! Please make a selection from the menu:
1. Display all books.
2. Display all available books.
3. Display all genres.
4. Display books in a genre.
5. Display all authors.
6. Display all books by an author.
7. Borrow a book.
8. Return a book.
9. Place a hold.
R. Return to previous menu.
Enter a choice: R
Welcome to the Library! Please make a selection from the menu:
1. Explore the catalogue.
2. View your patron record.
3. Show you favourite books.
4. Enter Admin Mode.
X. Exit the system.
Enter a choice: X
```

Requirements

- Your design will consist of exactly the following classes with the listed fields, declared as indicated. You may not add or remove classes or fields, however you may add constructors, functions and procedures to complete your design (in fact, you will have to!). You should pay careful attention to the tests on PLATE, as these will help guide you with some (but not all) of these methods.
- To help visualise the design, a partial class diagram has been provide – THIS DOES NOT HAVE EVERYTHING YOU NEED!
- Classes – your design will consist of these 7 classes:
    1. `Library`
    2. `Catalogue`
    3. `Patron`
    4. `Book`
    5. `Genre`
    6. `Author`
    7. `In` (this is just the class you've been using throughout the labs to facilitate simpler I/O – just copy it over)
- Fields – the classes will have the following fields:

```java
public class Library {
      private Catalogue catalogue;
      private List<Patron> patrons;
}

public class Catalogue {
      private Library library;
      private List<Book> booksOnShelf;
      private List<Book> booksOnLoan;
      private List<Genre> genres;
      private List<Author> authors;
}

public class Patron {
      private int ID;
      private String name;
      private List<Book> currentlyBorrowed;
      private List<Book> borrowingHistory;
}

public class Book {
      private String title;
      private Author author;
      private Genre genre;
      private List<Patron> holds;
}

public class Genre {
      private String name;
}

public class Author {
      private String name;
}
```

- The fields also have some additional requirements and strictures:
    1. Lists all have the abstract type of List<>, but must be instantiated with a concrete type that implements the List<> behaviour (you will see two of these in week 6, you can choose either – you may also want to think about why you might do things this way).
    2. The name String in Genre is stored in lowercase.

- Constructors – the constructors of the class have the following requirements:
    1. All constructors initialise the fields of their class.
    2. The Library constructor takes no parameters.
    3. The Catalogue constructor takes a single parameter, the Library which it belongs to.
    4. The Patron constructor takes two parameters, the ID and name, corresponding to the two fields identically named.
    5. The Book constructor takes three parameters, corresponding to the title, author and genre, with the same types as the respective fields.

6. The Genre constructor takes a single parameter, the name of the genre.
7. The Author constructor takes a single parameter, the name of the author.

- toString() – several of the classes will have a toString() function, with the following formatting:
    1. `Patron.toString()` will produce a string of the form:

       `<ID> <name>`

       e.g.

       `42 Luke Mathieson`

    2. `Book.toString()` will produce a string of the form:

       `<Author's name>, <Title>`

       e.g.

       `Joshua Bloch, Effective Java`

    3. `Genre.toString()` will produce a string of the form:

       `<genre name>`

       e.g.

       `new weird`

    4. `Author.toString()` will simply produce the author's name.

- The main method of the program will be in the Library class.
- Holds – the library should also implement a hold system. A patron may request a book to be held for them. If a book has any holds, it may not be loaned out to any other patron except the first patron on the hold list. Holds are processed sequentially, so multiple patrons may place a hold, and whoever gets in first must be the next to borrow it. A single patron may only place a single hold on a particular book – if they're already in the list, then that's all they get. A hold may only be cleared when that patron borrows the book (`TODO: this policy may require revision at the next meeting of the NSWIT board meeting`)

Advanced Requirements

To achieve a mark of >84, you must implement a favourites reporting function (or set of functions). This corresponds to an item in the main menu (see the trace above). Choosing this option should prompt the user for a valid patron ID, and then display the user's 3 most borrowed books, in order of highest borrowing frequency to lowest. If the user has only borrowed less than three books, it will show those books alone. A sample partial I/O trace follows, again red text is used to indicate user input:

```
Welcome to the Library! Please make a selection from the menu:
1. Explore the catalogue.
2. View your patron record.
3. Show you favourite books.
4. Enter Admin Mode.
X. Exit the system.
Enter a choice: 3
Enter a patron ID: 42
Luke Mathieson's favourite books are:
David Foster Wallace, Infinite Jest
Tim Powers, The Anubis Gates
Christopher Priest, The Islanders
```

3. Expected workload

The time to do the assignment to a credit/distinction level has been estimated at 25 hours for a student of average ability who has completed all the tutorial and lab exercises.

4. Online support

The Assignment 1 discussion board has been set up on UTSOnline so that students can ask questions, and other students can reply. The course coordinator will only post a reply only if the student response was wrong, or in the case of correcting a mistake in the assignment specification.

You must not post or share Java code to the discussion board. The board is there to help you, not to provide the solution. **Posting your code is academic misconduct and will reported**. Each time this rule is violated, the code will be removed and replaced with a comment of the form: "Strike 1: Posting code". After 3 strikes, the discussion board will be deleted because it did not work.

FAQs (Frequently Asked Questions) and their answers will be posted on PLATE alongside the assignment specification. If you have a question, check the FAQ first; it may already be answered there. You should read the FAQ at least once before you hand in your solution, but to be safe check it every couple of days. Anything posted on the FAQ is considered to be part of the assignment specification. The FAQ will be frozen (no new entries) two days before the due date; no questions will be answered after it is frozen.

If anything about the specification is unclear or inconsistent, contact the subject coordinator who will try to make it clearer by replying to you directly and posting the common questions and answers to the FAQ. This is similar to working on the job, where you ask your client if you are unsure what has to be done, but then you write all the code to do the task. Email luke.mathieson@uts.edu.au to ask for any clarifications or corrections to the assignment.

5. PLATE marking

Your solution is marked for correctness by PLATE (https://plate.it.uts.edu.au/) by comparing the output of your system to the output of the benchmark system. You can submit a solution to PLATE many times; I urge you to do this, so you receive credit for your work. Submission takes the same form as for the labs, you must package your assignment in a JAR file, including the source code.

PLATE will test the features of your program in a certain order: Classes and fields, then constructors, then goals from basic to advanced. PLATE cannot test the more advanced goals until the basic goals are working. To receive marks, you must pass PLATE's test cases in the order in which PLATE tests them.

Your code is marked by software, so you can get a good mark by fooling or spoofing the software. If you spoof a task worth N marks, you receive a penalty of 2*N marks.

6. Submission and return

Your solution is to be submitted to PLATE at https://plate.it.uts.edu.au/ under Applications Programming / Assessments / Assignment 1, in the same manner as your labs – package in a JAR file including the source code. Your provisional mark and feedback is generated immediately each time you submit to PLATE. However, analysis of spoofing, plagiarism, collusion and general cheating is done in the two weeks following the due date. If you are suspected of Academic Misconduct, I will forward your case to the Misconduct Committee and will notify you by email.

There is no scheduled late submission period. An extension of up to one week may be given by the subject coordinator before the due date; you have to supply documentary evidence of your claim. An extension CANNOT be given after the due date.

You may also apply for special consideration for reasons including unexpected health, family or work problems. More information about how to apply for special consideration can be found at http://www.sau.uts.edu.au/assessment/consideration.html.

7. Marking scheme

The marks for the assignment are divided into the following functionality components (note that individual tests may test several functionality components, and a functionality component may be tested by several tests):

Functionality Component	**Mark Allocation**
Fields	6
Constructors	8
Main Menu	5
Admin Menu	5
Catalogue Menu	5
Add Patron	5
Add Book	5
Remove Patron	5
Remove Book	5
Show Books	5
Show Books By Category	5
Borrow	5
Return	5
Patron Record	5
Favourites	16
Hold	10

This adds to a mark out of 100, at makes up 30% of your final assessment mark.