CST8132 Object-Oriented Programming

Lab 6: Library Management System

Due Date: Check Brightspace for the due dates

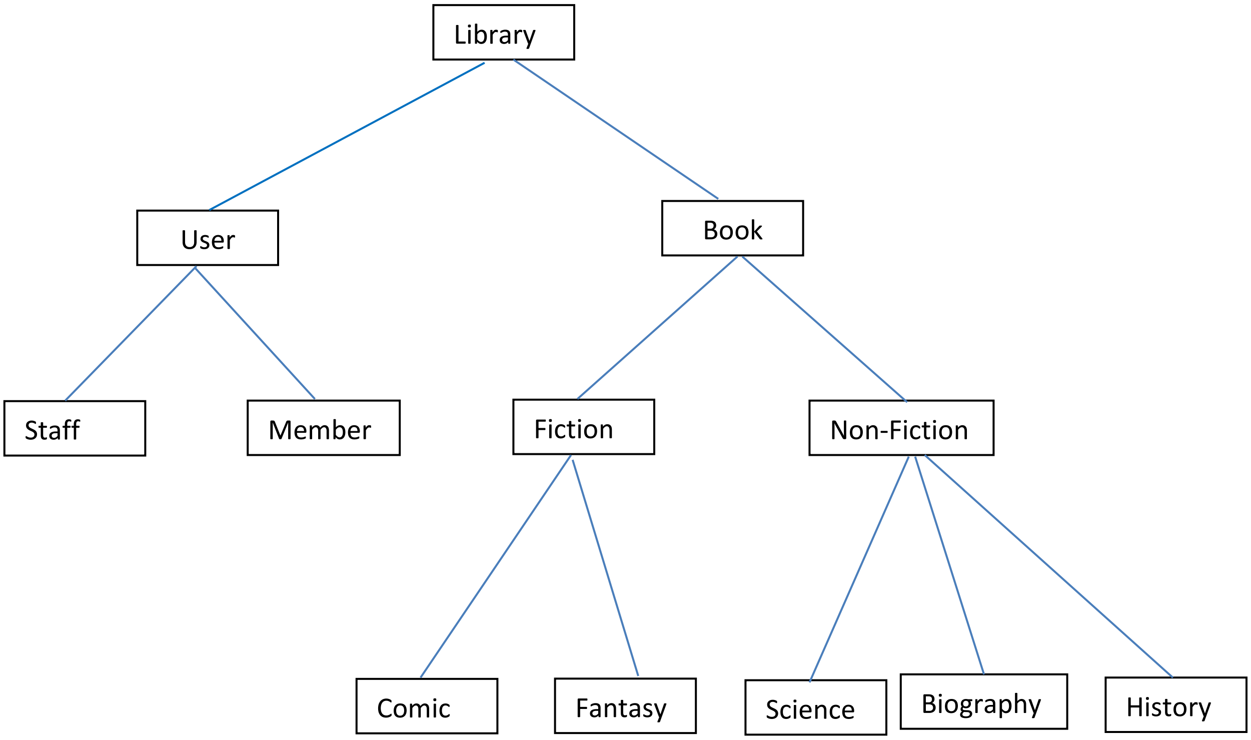
Marks: 40 marks (worth 5% of term mark)

Demo: Demo your code and output to your lab professor during your own lab hours.

Recommended Reading: Chapter 9 -11,15 of Deitel and Deitel, Java How to Program book

Exercise

In this lab, we create a few classes to create a library system. A library (name must be entered by the user) has users registered with them. A user can be a staff or a member. Each user has common attributes of a person (first name, last name, email, and phone number). Also, a staff will have a staff ID, floor and section on which they are working. Members will have an ID, age, list of books that they take from the library. Library has fiction and non-fiction books. Under fiction category, comic and fantasy books are there. Under non-fiction category, they have books from Science, History and Biography categories. There are specific **policies** that must be applied for the full system. After the due date, a fine of $1 will be applied each day until the book is returned. Maximum number of books a user can take is 5 (this is finalized, an no chance to change). Every time, when a user attempts to take a book, this check must be done so that they will not be able to have more than 5 books in their list. Every book should be returned within 14 days. If somebody missed the due date, fine should be added. Here is how the book classification looks like (NOT a UML diagram, but just a pictorial view of different types of books and users):



Requirements

1. Make sure that object-oriented principles are applied
2. Information should be read from the keyboard and from file
3. Exception handling should be included
4. Bulletproof your code so that it will not crash at any point
5. Only one Scanner object for the entire project. It should be passed accordingly
6. Main method which is in the LibraryManagement class will call appropriate methods from the Library class, and from there, rest of the methods for various functionalities should be called. Make sure that encapsulation is applied everywhere.
7. Do some research and find relevant properties for each class (in addition to the ones listed earlier).
8. In this lab, there is no strict restrictions about names of attributes, methods etc. But always make sure that you follow all OO principles.
9. You need to use ArrayList and arrays as required. If we know the size is fixed, you can use array. But, if there is a chance to grow, use ArrayList.

Submission

Submit your work (all java files, UML, Javadoc, test plan) to Brightspace by the due date given in Brightspace. Also, demonstrate your work to your lab professor. Both submission and the demo of submitted code are required to get grades.

Grading Scheme

Item	**Marks**
Correct access specifiers	3
Encapsulation	3
Inheritance	3
Polymorphism	3
Reading from keyboard	3
Reading from file	3
Exception Handling	3
Bulletproofing code	3
LibraryManagement class (driver)	3
Comments & Javadoc	5
UML	3
Test plan	5