CSC220 Lab03 - Inheritance

The goal of this week's lab is:

1. Practice the concept of inheritance
2. Start using JavaDoc.

Things you must do:

1. There are many details in this lab. Make sure you read and follow this instruction carefully and do things in order.
2. Always remember Java is case sensitive.
3. You must use file names, class names, package names as instructed.
4. You must include all the functions you have written during the lab when you are submitting your assignment.

Things you must not do:

1. You must not change the signature of any of these methods (name, parameters, ...). Just fill in the missing code inside them.
2. You must not create any different class.

In this lab we are going to construct a program for libraries that allows books to be checked in and out electronically. A book is represented by an **ISBN**, an **author**, and a **title**, all of which **cannot change** once the book has been created. (Please note that ISBNs are unique.)

A **library book** is a **book** together with a **holder** (representation of the person who has the book checked out) and a **due date**, both of which **can change** as needed. (Please note that for our purposes, all holders are unique.) For this lab, the holders are represented by their name as a String.

Part 0 - Create a project for this lab in

Eclipse

1. You are going to create a new project for this (and each of the remaining labs). You learned how to create a project in Eclipse before. Create a new Java project and call it **Lab03** (with no space and no other name – notice the capital 'L')

2. Create a package inside your project and call it **lab03** (no space and no other name – all lowercase).

3. Log into Blackboard and grab the four Java files from the **Lab3** folder ((on Google Drive link from blackboard assignment) and place them in the "lab03" package of the project you just created.

4. Refresh your Eclipse to make sure you are looking at the template code.

Part 1 – Complete Book class

The base class **Book** has been started for you. The **equals** method is left for you to fill in. Do not make any other changes to the Book class. This is similar to what you have done for your last assignment. The difference is that you first need to make sure the Object we are comparing to is a Book and then 'cast' it to a 'Book'. Use 'instanceof', if you don't know how to use it consult the slides. Do not worry about the errors in **Library.java**. They would be taken care of as soon as you get to Part 2.

Part 2 – LibraryBook class

1. Construct a class **LibraryBook** derived from Book and containing the library book's holder (a String) and due date represented by a GregorianCalendar. Consult JavaDoc page for how to use Objects from this class: http://docs.oracle.com/javase/6/docs/api/java/util/GregorianCalendar.html

2. Make sure to add the following line at the top of your class so that you can use GregorianCalendar .

```
import java.util.GregorianCalendar;
```

3. The **LibraryBook** class must include the following methods (you may add other methods as needed but not required).

- public LibraryBook(long isbn, String author, String title)
- public String getHolder()
- public GregorianCalendar getDueDate()
- public void checkin()
    - o When a library book is checked in, its holder and due date should be set to null.
- public void checkout(String holder, GregorianCalendar dueDate)
    - o When a library book is checked out, its holder and due date should be set accordingly.

- **Do not override the equals method in Book.**
- write tests in **LibraryTest.java** to make sure the implementation of your methods in LibraryBook is correct. There is sample book information for you under LibraryTester.java (part 2 samples). Don't forget to check the equal method in Book.java!

Part 3 – Library **class**

The **Library** class has been started for you. Fill in the method implementations as indicated. **Do not change the method signatures. Do not change the ones that have been implemented for you already.** There are many functions in this class to fill in. For the lab, you are only required to finish two of them.

- public String lookup (long isbn)
- public boolean checkout(long isbn, String holder, int month, int day, int year)

Test your Library implementation (so far) in LibraryTest.java. Remove anything

you have added to the main function and uncomment the code under “lab03 – part 2 test”. If you have implemented everything properly you should see “Testing done” after running your program. If you see any “TEST FAILED ...” you need to go back and debug your code.

You should have created a new project during the lab and you will be using the same project for this assignment. You are going to continue working on that project and complete the partial implementation of your Library class.

As you saw during the lab, we are going to construct a program for libraries that allows books to be checked in and out electronically. A book is represented by an ISBN, an author, and a title, all of which cannot change once the book has been created. (Please note that ISBNs are unique.) A library book is a book together with a holder (representation of the person who has the book checked out) and a due date, both of which can change as needed. (Please note that for our purposes, all holders are unique.) For this lab, a holder is represented by his/her name as a String.

Your job is to complete several methods defined that are incomplete.

Important Note: There are only two parts in this assignment BUT there are many details involved. Start early! You are encouraged to write more tests in your main function to make sure you have implemented all methods properly. Be sure that the project you submit contains LibraryTest.java so that we can confirm you have tested your code. Note that we will not run your tester code. Instead, we will first run your code with the examples we provided and then, we will test your code with new and different examples.

Part 0

- You first must make sure that you have already finished the lab successfully and have all methods in the lab instruction working properly. Then, grab the new version of **LibraryTest.java** from the assignment ZIP file on Blackboard.

- Note that there are special cases that you should be careful about, some of which are included in the LibraryTest for you.

Part 1 - lookup

```
public ArrayList<LibraryBook> lookup(String holder)
```

- Returns the list of library books checked out to the specified holder.
- If the specified holder has no books checked out, returns an empty list.

Part 2 - checkin

```
public boolean checkin(long isbn)
```

- Unsets the holder and due date of the library book.
- If no book with the specified ISBN is in the library, returns false.
- If the book with the specified ISBN is already checked in, returns false.
- Otherwise, returns true.

```
public boolean checkin(String holder)
```

- Unsets the holder and due date for all library books checked out by the specified holder.
- If no books with the specified holder are in the library, returns false;
- Otherwise, returns true.