**CSC220 Assignment04**

**Generic Programming**

The goal of this week’s assignment is:

1. To practice generic programming

2. Learn how to use Comparator interface

3. Continue experimenting with object types

**Things you must do:**

1. There are many details in this assignment. Make sure you read the whole thing carefully before writing any code and closely follow this instruction.

2. You must complete your assignment individually.

3. Always remember Java is case sensitive.

4. Your file names, class names, and package name must match exactly as they are specified here.

5. Your project must include the methods you implemented in the lab.

**Things you must not do:**

1. You must not change the file names, class names, package names.

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

3. You must not create any different class.

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.)

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

**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.
* **DO NOT** start your assignment unless you have all the features in the lab working. If you are having trouble with the material from the lab, please seek help from TAs during office hours and email.

**Part 1 - Retrieving a list of library books sorted by author**

* This is similar to the feature you implemented in the lab getInventoryList. You can accomplish this task by finishing the implementation of the AuthorComparator (the class declaration is provided, you must fill in the code), and then you must fill in the getOrderedByAuthor.
* You do not need to sort by last name, just sort by the full author String as is. If two books have the same author, this Comparator should break the tie with the book title.
* For example, Mushroom\_Publishing.txt contains the following two books:

Moyra Caldecott The Eye of Callanish

Moyra Caldecott Crystal Legends

* AuthorComparator should treat "Crystal Legends" as less than "The Eye of Callanish", even though they have the same author, but since 'C' is alphabetically less than 'T'. To perform these comparisons, simply use String's built-in compareTo method.
* Back in getOrderedByAuthor, invoke the sort method with an instance of AuthorComparator.

**Part 2 - Retrieving a list of overdue library books sorted by due date (oldest first)**

This feature is left for you to implement. Hints:

* In getOverdueList, first make a copy of the list of library books, but include only those that are overdue (note that GregorianCalendar has a compareTo method for comparing dates).
  + **Be careful** when iterating over a library book whose due date is **NULL**! (Why?)
* Then invoke the sort method on the overdue list with an instance of the DueDateComparator class (a Comparator), for which the class declaration is provided, but you must fill in the rest of the implementation. If you don’t know how to use sort look at the implementation of getInventoryList.

**Part 3 – Testing**

Test the features you implemented by adding your own examples inthe LibraryGenericTest! We have provided a file that includes a list of sample books called Mushroom\_Publishing.txt. In order to use this file:

* Right click on the Lab04 (i.e., the project name – not the src folder) and then “new->file” and type “Mushroom\_Publishing.txt”. A file with the same name will show up under “JRE System Library”.
* Now grab a copy of Mushroom\_Publishing.txt from the assignment 4 ZIP folder from Google Drive and paste the content of it into the file you just created.
* Now you can uncomment lines after “test a medium library” and start testing your implementation.

**Remarks**

* If you see any “TEST FAILED …” you need to go back and debug your code. You are encouraged to write more tests to make sure you have implemented all methods properly.
* Make sure to submit your assignment by uploading your **Lab04** folder into your **csc220-cXXXX** folder by the deadline **(Tuesday @ 11:59pm)**
* Be sure to include **LibraryGenericTest.java** in your submission to BOX. Remember that we will **not** run your testing code; we will only confirm that you have tested your code.
* **For all your assignments, please start early and seek help early (either from the instructor or the TAs).**