CSC220 Lab 4
Generic Programming

The goal of this week's lab 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 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. Your assignment submission must include the methods you have written during the lab.

Things you must not do:

1. You must not change the signature of any of these methods (name, parameters, …) and just fill in the missing code inside them, unless you have been asked to do so.
2. You must not create any different class.

In this lab we are going to construct another version of a program for libraries that allows books to be checked in and out electronically. Similar to last lab, 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.)

To make our task more challenging this time, some of the libraries that will use our program represent holders with names. Others represent holders with phone numbers. Furthermore, we

hope to sell our program to even more libraries whose representation of holders we cannot anticipate. Thus, our library program must be **generic**.

Finally, we hope to squeeze even more money out of our customers by offering two additional features: an operation that retrieves the list of all books in the library sorted either by ISBN or by author (for inventory purposes), and an operation that retrieves the list of all overdue library books.

Part 0

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 **Lab04** (with no space and no other name – notice the capital 'L')
2. Create a package inside your project and call it **lab04** (no space and no other name – all lowercase).
3. Create a classes (or Java files) in this package similar to previous lab. That is:
    a. Book.java (nothing else)
    b. Library.java (nothing else)
    c. LibraryBook.java (nothing else)
    d. LibraryTest.java (nothing else)
4. Open your university box account, grab your assignment 3 files and replace the files you just created with your assignment.
5. Log into blackboard, go to **Lab03** (yes, lab03!) and grab a copy of LibraryTest.java.
6. Uncomment the code after line 27.
7. Run LibraryTest.java. Assuming you have finished your homework properly, all you should see is "Testing done." If you see anything else (or any red text) start debugging your code and ask for help from TA.

You need a correct implementation of the code to continue.

Part 1 – LibraryBookGeneric class

1. Create a class or Java file called LibraryBookGeneric.java in your project.
2. Copy the content of your LibraryBook class then modify it to make the type of the library book's holder generic. The header for your new class should be the following.

    a. `public class LibraryBookGeneric<Type> extends Book`

3. For the most part, modification will involve replacing the `String` type for the library book's holder in your original class to `Type` in the new class. **Be careful. It is not correct to replace every occurrence of String with `Type`.**

Part 2 – LibraryGeneric class

1. Create a class (or Java file) in this package and call it LibraryGeneric.java. Copy your Library class in this file and modify it to make the list of library books a list of generic library books. The header for your new class should be the following.
    a. `public class LibraryGeneric<Type>`
2. For the **most part**, modification will involve replacing the `String` type for the library book's holder in your original class to `Type` in the new class and replacing the `ArrayList<LibraryBook>` type for the library in your original class to `ArrayList<LibraryBookGeneric<Type>>` in the new class. **Be careful. Go through the class to make sure you have applied required changes to make your class generic.**

Part 3 – LibraryGenericTest

1. Create a class (or Java file) in this package and call it LibraryGenericTest.java.
2. The sample test class LibraryGenericTest has been started for you. Grab it from Blackboard in Lab04 folder and replace the file you just created.
3. Create a class (or Java file) in this package and call it PhoneNumber class. This class has been implemented for you. Grab it from Blackboard in Lab04 folder and replace the file you just created.

4. Now run **LibraryGenericTest** (not LibraryTest!), you should ONLY see "Testing done". If you see anything else (especially red text), you need to debug your implement (the generic classes).
5. The LibraryGenericTest class creates two libraries, one that identifies holders with (String) names and another that identifies holders with PhoneNumber objects.
6. **You MUST extend this class to include more exhaustive testing.**

Part 4 – Retrieving a list of library books sorted by ISBN

1. Import java.util.Comparator to LibraryGeneric.java.
2. Grab the code inside phase3.txt and add it in LibraryGeneric.java.
    a. copy and paste it inside the curly braces of the class.
3. The provided code includes a method for sorting an ArrayList of items. Both the type of items in the ArrayList and the order of the sort is generic. The ordering is specified by the Comparator object passed to the method. Most of this code has been implemented for you. The only missing part is for you to complete the implementation of `compare` method inside `OrderByIsbn` that implements `comparator` interface.
    a. the first step is to specify how many parameters `compare` method should receive and what is the type(s). You can consult JavaDoc if are in doubt: https://docs.oracle.com/javase/7/docs/api/java/util/Comparator.html
    b. You need to think about how to implement this method and what this method needs to return – again use JavaDoc (and class lectures) to accomplish this task.

4. **You last task is to write tests in LibraryGenericTest.java to test your `getInventoryList` method.**

Make sure to grab your code before your leave the lab, you will develop more functionality for the classes for your assignment.

For all your assignments, please start early and seek help early (either from the instructor or the TAs).