AP Computer Science Homework 13(Winter Break Assignment)

Due date: Monday, February 20, 2017
Instructor: Mr. Alwin Tareen

An Electronic Book Rental System for a Library

- The libraries of SmallTownX need a new electronic book rental system, and it is up to you to build it. SmallTownX has two libraries. Each library offers many books to rent. Customers can print the list of available books, borrow, and return books.

- Three classes have been given, `Book`, `Library`, and `LibraryTest`, that provide the functionality for the book database. You must implement the missing methods in the file `Library.java` to make these classes work.

Part A: Implement the `Library` class

- We need to build the class that will represent each library, and manage a collection of books. All libraries have the same hours: 9am to 5pm daily. However, they have different addresses and book collections (i.e., ArrayLists of `Book` objects).

- The files `Book.java`, `Library.java`, `LibraryTest.java` and `LibraryJUnitTest.java` have been provided for you. A `public static void main(String[] args)` method is included in `LibraryTest.java` which creates two libraries, and then performs some operations on the books. However, most of the necessary methods in the file `Library.java` are missing.

- Specifically, the methods you will need to define and implement are the following:
  - `public static String displayOpeningHours()`
  - `public String displayAddress()`
  - `public String addBook(Book novel)`
  - `public String borrowBook(String novelName)`
  - `public String returnBook(String textName)`

- Note that all of these methods return a `String`. If you examine the output of the program run, you will notice that every time some method is called, it returns a confirmation message.

- Do not include any `System.out.println()` statements to display a confirmation message to the output. These `String` messages are return values from each method, which are then printed out in the file `LibraryTest.java`.

- Be careful when comparing `String` objects. Use `string1.equals(string2)` for comparing the contents of `string1` and `string2`.

- You should get a small part working at a time. Start by commenting out the entire `public static void main(String[] args)`, then uncomment it line by line. Run the program, get the first lines working, then uncomment the next line, get that working, etc.

- You must *not* modify the code in `Book.java` and `LibraryTest.java`

- This is a much tougher assignment than what you are accustomed to. Start working on this project as soon as possible, and leave plenty of time for debugging. Also, I will be checking carefully for plaigiarism violations.

- The output when you run this program should be similar to the following:

```
Adding books to firstLibrary:
You have successfully added The Da Vinci Code
You have successfully added Le Petit Prince
You have successfully added A Tale of Two Cities
You have successfully added The Lord of the Rings

Library hours:
Libraries are open daily from 9am to 5pm.

Library addresses:
10 Main St.
228 Liberty St.

Borrowing The Lord of the Rings:
You have successfully borrowed The Lord of the Rings
Sorry, this book is already borrowed.
Sorry, this book is not in our catalog.

Books available in the first library:
The Da Vinci Code
Le Petit Prince
A Tale of Two Cities

Books available in the second library:
No books in our catalog.

Returning The Lord of the Rings to the second library:
Sorry, this book is not in our catalog.

Returning The Lord of the Rings to the first library:
You have successfully returned The Lord of the Rings

Books available in the first library:
The Da Vinci Code
Le Petit Prince
A Tale of Two Cities
The Lord of the Rings
```

- You are provided with the files `Book.java`, `Library.java`, `LibraryTest.java` and `LibraryJUnitTest.java` to develop this program.
- Write your code in the file `Library.java` in the area indicated by `// YOUR CODE HERE`.
- **Do not change any other part of the program.**
- On your `BlueJ` project window, you should see a button labelled `Run Tests`. Press this button to run the `JUnit` tests.
- You should see a `BlueJ: Test Results` window pop up. If everything is correct, you should see a green bar that indicates that your code has passed the `JUnit` tests. If your program is incorrect, you will see a red bar. You can click on the method name to get more information about the problem. Otherwise, just click on the `Close` button, and you can go ahead and upload this program to `Web-CAT`.

Part B: Submission

- Submit your Java program `Library.java` by uploading it to the Web-CAT automated grading platform.