6.092 Introduction to Software Engineering in Java
January (IAP) 2009

6.092: Assignment 4

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

Problem

We provide two classes, Book and Library, that together provide the functionality for the book database. You must implement the missing methods to make these classes work.

Step One: Implement Book

First we need a class to model books. Start by creating a class called `Book`. Copy and paste the skeleton below. This class defines methods to get the title of a book, find out if it is available, borrow the book, and return the book. However, the skeleton that we provide is missing the implementations of the methods. Fill in the body of the methods with the appropriate code. The `main` method tests the methods. When you run the program, the output should be:

```
Title (should be The Da Vinci Code): The Da Vinci Code
Borrowed? (should be false): false
Borrowed? (should be true): true
Borrowed? (should be false): false
```

Step Two: Implement Library

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

Create a class called `Library`. Copy and paste the skeleton below. We provide a `main` method that creates two libraries, then performs some operations on the books. However, all the methods and member variables are missing. You will need to define and implement the missing methods. Read the `main` method and look at the compile errors to figure out what methods are missing.

Notes

- Some methods will need to be *static* methods, and some need to be *instance* methods.
- Be careful when comparing Strings 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 the entire `main` method, then uncomment it line by line. Run the program, get the first lines working, then uncomment the next line, get that working, etc. You can comment a block of code in Eclipse by selecting the code, then choosing Source → Toggle Comment. Do the same again to uncomment it.
- You must *not* modify the `main` method.

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

```
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 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 book in catalog

Returning The Lord of the Rings:
You 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
```

Submission Instructions

Submit both files (`Book.java` and `Library.java`) via Stellar.

Good luck!

Book.class

```java
public class Book {

    String title;
    boolean borrowed;

    // Creates a new Book
    public Book(String bookTitle) {
        // Implement this method
    }

    // Marks the book as borrowed
    public void borrowed() {
        // Implement this method
    }

    // Marks the book as available, i.e., not borrowed
    public void returned() {
        // Implement this method
    }
    // Returns true if the book is borrowed, false otherwise
    public boolean isBorrowed() {
        // Implement this method
    }

    // Returns the title of the book
    public String getTitle() {
        // Implement this method
    }

    public static void main(String[] arguments) {
        // Small test of the Book class
        Book example = new Book("The Da Vinci Code");
        System.out.println("Title (should be The Da Vinci Code): " + example.getTitle());
        System.out.println("Borrowed? (should be false): " + example.isBorrowed());
        example.borrowed();
        System.out.println("Borrowed? (should be true): " + example.isBorrowed());
        example.returned();
        System.out.println("Borrowed? (should be false): " + example.isBorrowed());
    }
}
```

Library.class

```
public class Library {
    // Add the missing implementation to this class

    public static void main(String[] args) {
        // Create two libraries
        Library firstLibrary = new Library("10 Main St.");
        Library secondLibrary = new Library("228 Liberty St.");

        // Add four books to the first library
        firstLibrary.addBook(new Book("The Da Vinci Code"));
        firstLibrary.addBook(new Book("Le Petit Prince"));
        firstLibrary.addBook(new Book("A Tale of Two Cities"));
        firstLibrary.addBook(new Book("The Lord of the Rings"));

        // Print opening hours and the addresses
        System.out.println("Library hours:");
        printOpeningHours();
        System.out.println();

        System.out.println("Library addresses:");
        firstLibrary.printAddress();
        secondLibrary.printAddress();
        System.out.println();
        // Try to borrow The Lords of the Rings from both libraries
        System.out.println("Borrowing The Lord of the Rings:");
        firstLibrary.borrowBook("The Lord of the Rings");
        firstLibrary.borrowBook("The Lord of the Rings");
        secondLibrary.borrowBook("The Lord of the Rings");
        System.out.println();

        // Print the titles of all available books from both libraries
        System.out.println("Books available in the first library:");
        firstLibrary.printAvailableBooks();
        System.out.println();
        System.out.println("Books available in the second library:");
        secondLibrary.printAvailableBooks();
        System.out.println();

        // Return The Lords of the Rings to the first library
        System.out.println("Returning The Lord of the Rings:");
        firstLibrary.returnBook("The Lord of the Rings");
        System.out.println();

        // Print the titles of available from the first library
        System.out.println("Books available in the first library:");
        firstLibrary.printAvailableBooks();
    }
}
```