# Exercise NMCGJ: Making a Library



The libraries of a small town need a new electronic borrowing system. The town has two libraries, and each library offers a collection of books to borrow. Customers can print the list of available books, borrow, and return books.

## Problem

Make two classes, Book and Library, that provide the functionality of the book database. Implement the missing methods to make these classes work.

#### Implementation of Book

First we need a class to model books. It should provide methods to get the title of a book, find out if it is available, borrow the book, and return the book.

Create a class called Book. Use the skeleton below. Only the main method is provided, and creates a book. Define and implement the missing methods. When running the program, the output should be:

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

#### Implementation of 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: 9:00 to 17:00 daily, but they have different addresses and book collections (i.e., arrays of Book objects). You can assume that the maximal capacity of each library is 100 books.

Create a class called Library. Use the skeleton below. Only the main method is provided, and creates two libraries. Define and implement the missing methods. When running the program, the output should be:

```
Library hours:
Libraries are open daily from 9:00 to 17:00
Library addresses:
Main Street 10
Liberty Street 228
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 the 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
```

#### Notes

- Some methods need to be *static* methods, and others *instance* methods.
- Be careful when comparing String objects. Use string1.equals(string2) for comparing the contents of string1 and string2.
- Start by getting a small part working at a time. For example, comment the entire main method, then uncomment it line by line (in Eclipse: select some code and choose Source > Toggle Comment).
- Making a class diagram could be useful.
- Do NOT modify the main methods.

### Java classes

#### Book.java

```
public class Book {
   // Add the missing implementation to this class!
   // A small test of the Book class.
   public static void main(String[] arguments) {
      Book book = new Book("The Da Vinci Code");
      System.out.println("Title (should be The Da Vinci Code): "
         + book.getTitle());
      System.out.println("Available? (should be true): "
         + book.isAvailable());
      book.borrowed();
      System.out.println("Available? (should be false): "
         + book.isAvailable());
      book.returned();
      System.out.println("Available? (should be true): "
         + book.isAvailable());
   }
}
```

#### Library.java

```
public class Library {
   // Add the missing implementation to this class!
   // A small test of the Library class.
   public static void main(String[] args) {
      // Create two libraries
      Library firstLibrary = new Library("Main Street 10");
      Library secondLibrary = new Library ("Liberty Street 228");
      // 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:");
      Library.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 books from the first library
      System.out.println("Books available in the first library:");
      firstLibrary.printAvailableBooks();
   }
```

```
}
```