

CSCI 2540 Assignment 4

100 points

Due date: Tuesday, Feb. 14 (before class)

In this assignment, you will write a program to simulate an inquiry system of a small library. The program should operate as follows:

1. Read the library catalog from an input data file. The data file should be named **catalog.txt**.

There is one input line per book, and these lines have the following format:

<book_id> <book_title> <ISBN> <author_last_name> <category>

where <book_id> is a 5-digit positive integer (you may assume the leftmost digit is not a zero), <book_title> is a string of at most 30 characters with no embedded blanks (you can use “_” in between words in a title), <ISBN> is a string of 10 digits, <author_last_name> is a string of at most 10 characters, and <category> is a character (‘F’ if the book is fiction; ‘N’ if it is a non-fiction book). You may assume the catalog has no more than 100 books (a small library). A sample catalog is shown at the end of the specification.

Make sure to include exception handling code related to file input.

2. Read from standard input a customer’s inquiry with a given book_id and output the complete information of the book. Your program should allow the customer to continue to inquire about other books. When a zero input for <book_id> is entered, it means the end of the customer’s inquiry.

The output should include the book_id, book_title, ISBN, author_last_name, and “Fiction” or “Non-Fiction” for category, printed on a single line. A sample output is shown at the end of the specification.

3. Exception handling:

The following input error is possible and should be handled by your program:

<book_id> not in the category file.

You need to define a **BookNotFoundException** and handle it by writing an error message. This should be defined as a checked exception. It should include two constructors. One is the default constructor, and the other is one-parameter constructor and the parameter type is String.

Program Structure:

Your source code should be developed in three files: **Book.java**, **BookDemo.java**, and **BookNotFoundException.java**.

Book.java will contain the class definition for a book according to the requirements specified below. *BookDemo.java* will be the application program that runs the simulation of the inquiry. *BookNotFoundException.java* will define the checked exception.

Data Structure:

Each catalog item should be an object of the *Book* class. Define a separate instance variable of the appropriate type for the five pieces of information about each book. Instance variables should be maintained as *private* data. Besides constructor(s), the following methods are required for the *Book* class.

- A getter method for each instance variable.
- A *toString* method that takes no parameter and returns all the information of the book as a combined string, including *book_id*, *book_title*, *ISBN*, *author_last_name*, and “Fiction” or “Non-Fiction” for category.
- A static method *bookSearch* that takes three input parameters: (1) an array of *Book* objects that represent the entire catalog; (2) an integer specifying how many books are actually in the array; and (3) an integer representing a *book_id*. The method should search the array of books looking for the book with the given *book_id* as specified by the third parameter. The method should return the index within the array. If it cannot find the item, the method should throw a *BookNotFoundException* but it is not handled in this method. Instead it will be handled in the main method where it is called.

Sample Catalog file:

10001	Emma	0486406482	Austen	F
12345	My_Life	0451526554	Johnson	N

Sample interaction:

Enter book id: 12345

Book id: 12345, Title: My_Life, ISBN: 0451526554, Author: Johnson, Non-fiction

Enter book id: 10001

Book id: 10001, Title: Emma, ISBN: 0486406482, Author: Austen, Fiction

Enter book id: 0

Note: if you use Eclipse, the input file should be placed outside of src folder.

Submission instructions:

To submit your programs, you need to submit your programs electronically on Blackboard. Please also bring a hard copy of your programs to the class to submit.

For the ease of grading your assignment, **please use a named package for each of your assignment.** For example, for assignment 4, please create a new package and **name your package as assg4_yourlastname** (with the first letter of your last name in uppercase and the rest in lower case), such as `assg4_Smith`. You also need to include a statement such as “`package assg3_Smith;`” at the beginning of each of your .java file. **Please follow this naming convention exactly for all future assignments. You will be deducted points for not doing so.** When you submit your files to Blackboard, please submit your package folder (with source code only, i.e., .java files) as one zip file.