



## COMP 130 - Introduction to Programming with Java

**Class:** `Book.java`, `TestBook.java`

**Score:** ★★ ★ (Medium)

**(Book.java)** Write a class named **Book** that keeps track of book objects such that the instance data contains the book's **title** and **author** and a unique identification number, say **id**, that starts from 1 for the first book object and is incremented by 1 each time a new book with at least the author specified is created. All variables should be **private**. The required methods in the Book class are:

- **Three constructors:** One without parameters, that sets the title and author to "unknown" and id to 0; one with a single parameter that sets the author to the given value as parameter, sets the title to "unknown", increments
- The **set** methods for title and author.
- The **get** methods for title, author, and id.
- The **compareAuthor()** method that compares the current book object's author surname with a string given as a parameter and returns true if they match, and false otherwise.
- The **toString** method that returns a text including the book's title, author and id. Refer to the sample runs for the test program for details.
- The **getInitials** method that returns the **initial letters** of the author's first name(s) and last name, if the author's name is known (not equal to "unknown", or not null). Assume that there can be at most two names and one surname each separated by a single blank. Refer to the sample runs for the test program for details.

**(TestBook.java)** Write a **TestBook** class that tests the Book class. The program first asks the user the total number of books that s/he wants to enter. The user is then allowed to enter title and author information for each of the books. The program takes care of necessary object initializations and stores all the book objects in an object array of type `Book[]`. After getting the books' information, the program goes into a loop asking the user each time an author surname which is a String input; searches the whole Book array and displays all the books with the entered author surname. The information of a book is displayed using its id, title, **initials of author's names** and the surname, as specified in the sample runs below. **The input iteration stops when the author information is entered empty.**

### Sampe run 1

Enter the number of books: 1  
Type the title: Windows NT Server 4.0  
Type the author: Russel

To end the search process, bypass the question by typing the enter key!

Enter the author's surname to search: Russel  
Book No: 1 entitled "Windows NT Server 4.0" written by Russel

Enter the author's surname to search: Yemez  
No records found...

Enter the author's surname to search:  
(the program exists)

### Sampe run 2

Enter the number of books: 5  
Type the title: Java Software Solutions

Type the author: Lewis Loftus  
Type the title: Introduction to Java Programming with JBuilder  
Type the author: Yvet Daniel Liang  
Type the title: Windows NT Server 4.0  
Type the author: Russel  
Type the title:  
Type the author: Kenneth Rosen  
Type the title: Rapid Java application development using JBuilder 3  
Type the author: Yvet Daniel Liang

To end the search process, bypass the question by typing the enter key!

Enter the author's surname to search: Liang  
Book No: 2 entitled "Introduction to Java Programming with JBuilder" written by Y. D. Liang  
Book No: 5 entitled "Rapid Java application development using JBuilder 3" written by Y. D. Liang

Enter the author's surname to search: Loftus  
Book No: 1 entitled "Java Software Solutions" written by L. Loftus

Enter the author's surname to search: Rosen  
Book No: 4 entitled "unknown" written by K. Rosen

Enter the author's surname to search:  
(*the program exists*)