

CSE 1322L – Assignment 2 (Fall 2025)

Introduction

In this assignment, you will write a simple program to help keep track of publications in a database. This program must allow new publications to be added, and existing ones to be retrieved or removed. This system has very basic library functionality, allowing publications to switch between being available and unavailable as visitors borrow them.

Going forward, you will be asked to write methods. Method headers may be provided explicitly:

```
static boolean performer(String, int)
```

Or descriptively:

Performer: A static method which returns a boolean and has two parameters: a String and an integer.

In either case, the method's functionality will be described in English right after.

You will also be required to write classes. Remember that, **unless stated otherwise, class fields (attributes) must always be set to **private** while class methods (behaviors) and constructors must always be set to **public**.**

Requirements

The features described below must be in your program:

- A class called "Publication"
 - It has 6 fields:
 - A static integer field called "nextId", initialized at 10001
 - 2 integer fields named "publicationId" and "year", respectively
 - 2 String fields, named "title" and "author", respectively
 - A boolean field named "available"
 - **Publication(String, String, int):** Assigns nextId to publicationId, then increments nextId by 1. Initializes "available" with true. Initializes "title" and "author" with the first and second arguments, respectively. The last argument is used to initialize "year" with the following restrictions:
 - If the integer argument is 0 or negative, set "year" to 1970
 - Otherwise, use the integer argument to set "year"
 - It has getters for all fields **except "nextId"**
 - It has a setter for "available"
 - It overrides toString() to return the following string:

{title} by {author} ({year})
#{publicationId}
AVAILABLE/NOT AVAILABLE

- Regarding the string above, note the following:
 - Replace the curly braces and their contents with the contents of the appropriate field
 - The string must span 3 lines
 - The last line will either be just “AVAILABLE” if the “available” field is currently true or “NOT AVAILABLE” otherwise. Never both.
- A class named “Shelf”
 - It has a single field: an arraylist of type Publication, called “publications”
 - **void addPublication(Publication):** Inserts the argument at the end “publications”
 - **void removePublication(Publication):** Removes the argument from “publications”
 - **Publication getPublication(int):** If “publications” has a Publication whose publicationId matches the argument, return that Publication. If there is no match, return null
 - **String listByKeyword(String):** Searches “publications” for Publication objects whose title contains the argument.
 - If no match is found, returns “NO MATCH WITH KEYWORD”
 - Otherwise, concatenates the toString() of all Publication objects whose title contains the argument, returning said concatenation
 - The match must be case insensitive.
 - If there is a Publication with the title “Simple Programs”, another one with the title “System processes” and the user searches for “pRo”, both objects must be considered a match
 - Separate each toString() by at least 1 line break (“\n”)
 - **String listByAuthor(String):**
 - Searches “publications” for Publication objects whose author matches exactly the argument.
 - If no match is found, returns “NO SUCH AUTHOR”.
 - Otherwise, concatenates the toString() of all Publication objects whose author matches the argument, returning said concatenation
 - Separate each toString() by at least 1 line break (“\n”)
 - **String toString():**
 - If “publications” is empty, returns “NO PUBLICATIONS”.
 - Otherwise, concatenates the toString() of all Publication objects inside of “publications”, returning said concatenation
 - Separate each toString() with at least 1 line break (“\n”)
- Driver
 - Create a Shelf object, herein called “shelf”

- In a loop, implement the menu options below:
 - **1. Add publication:** Prompts the user for the necessary information to create a Publication object. This object is then passed to shelf's addPublication()
 - **2. View all publication:** Prints the toString() of all the publications currently stored in "shelf"
 - **3. Find by author:** Prompts the user for an author name, passing it to shelf's listByAuthor(), and printing the returned value
 - **4. Find by keyword:** Prompts the user for a keyword, passing it to shelf's listByKeyword(), and printing the returned value
 - **5. Make publication available:** Prompts the user for a publication ID. If no Publication exists with that ID, print an error message. If a Publication with that ID exists but it is already available, print an error message. Otherwise, set that Publication to be available and print a success message.
 - **6. Make publication unavailable:** Prompts the user for a publication ID. If no Publication exists with that ID, print an error message. If a Publication with that ID exists but it is already unavailable, print an error message. Otherwise, set that Publication to be unavailable and print a success message.
 - **7. Remove publication:** Prompts the user for a publication ID. If no Publication exists with that ID, print an error message. Otherwise, call shelf's removePublication() and print a success message.
 - **8. Quit:** Terminates the program

Deliverables

- Assignment2.java (driver)
- Shelf.java
- Publication.java

Considerations

- You will get partial credit for partial work, as long as the rubric permits it. Try to do as much as you can.
- Despite what the deliverables above say, you can submit all of your classes in a single file.
- Don't forget that you can add a line break in a string by inserting a "\n" in the string.
- This assignment requires you to make checks between 2 strings, for both full equality substring equality. You can find all string methods in the [documentation](#).
- If you get stuck while implementing a menu option, leave yourself a comment saying "TODO", followed by why you got stuck, and then go implement something else. At a later date, search for any TODOs you still have and try to fix them. Don't forget to erase the TODO once you have applied a fix.

Sample Output (user input in red)

[Publication Database]

1. Add publication
2. View all publications
3. Find by author
4. Find by keyword
5. Make publication available
6. Make publication unavailable
7. Remove publication
8. Exit

Enter option: **2**

NO PUBLICATIONS

1. Add publication
2. View all publications
3. Find by author
4. Find by keyword
5. Make publication available
6. Make publication unavailable
7. Remove publication
8. Exit

Enter option: **1**

Enter title: **The History of Databases - Part 1**

Enter name of author: **Morty Tittensor**

Enter year of publication: **2012**

Publication added

1. Add publication
2. View all publications
3. Find by author
4. Find by keyword
5. Make publication available
6. Make publication unavailable
7. Remove publication
8. Exit

Enter option: **1**

Enter title: **Principles of Programming**

Enter name of author: **Genie Ruskin**

Enter year of publication: **1999**

Publication added

1. Add publication
2. View all publications
3. Find by author
4. Find by keyword
5. Make publication available
6. Make publication unavailable
7. Remove publication
8. Exit

Enter option: **1**

Enter title: **Memory Management Demystified**

Enter name of author: **Fenton Beverley**

Enter year of publication: **-2025**

Publication added

1. Add publication
2. View all publications
3. Find by author
4. Find by keyword
5. Make publication available
6. Make publication unavailable
7. Remove publication
8. Exit

Enter option: **1**

Enter title: **Using version control like a professional**

Enter name of author: **Hope Marks**

Enter year of publication: **2020**

Publication added

1. Add publication
2. View all publications
3. Find by author
4. Find by keyword
5. Make publication available
6. Make publication unavailable
7. Remove publication
8. Exit

Enter option: **1**

Enter title: **The History of Databases - Part 2**

Enter name of author: **Morty Tittensor**

Enter year of publication: **2013**

Publication added

1. Add publication
2. View all publications
3. Find by author
4. Find by keyword
5. Make publication available
6. Make publication unavailable
7. Remove publication
8. Exit

Enter option: **3**

Enter name of author: **Morty Marks**

NO SUCH AUTHOR

1. Add publication
2. View all publications
3. Find by author
4. Find by keyword
5. Make publication available
6. Make publication unavailable
7. Remove publication
8. Exit

Enter option: **3**

Enter name of author: **Morty Tittensor**

The History of Databases - Part 1 by Morty Tittensor (2012)

#10001

AVAILABLE

The History of Databases - Part 2 by Morty Tittensor (2013)

#10005

AVAILABLE

1. Add publication
2. View all publications
3. Find by author

4. Find by keyword
5. Make publication available
6. Make publication unavailable
7. Remove publication
8. Exit

Enter option: 4

Enter name keyword: pro

Principles of Programming by Genie Ruskin (1999)

#10002

AVAILABLE

Using version control like a professional by Hope Marks (2020)

#10004

AVAILABLE

1. Add publication
2. View all publications
3. Find by author
4. Find by keyword
5. Make publication available
6. Make publication unavailable
7. Remove publication
8. Exit

Enter option: 6

Enter publication id: 1004

No publication with that id

1. Add publication
2. View all publications
3. Find by author
4. Find by keyword
5. Make publication available
6. Make publication unavailable
7. Remove publication
8. Exit

Enter option: 6

Enter publication id: 10004

Publication #10004 has been made unavailable.

1. Add publication
2. View all publications
3. Find by author
4. Find by keyword
5. Make publication available
6. Make publication unavailable
7. Remove publication
8. Exit

Enter option: 6

Enter publication id: 10004

ERROR: Publication #10004 is already unavailable.

1. Add publication
2. View all publications
3. Find by author
4. Find by keyword
5. Make publication available
6. Make publication unavailable
7. Remove publication
8. Exit

Enter option: 7

Enter publication id: 10001

Publication #10001 has been removed.

1. Add publication
2. View all publications
3. Find by author
4. Find by keyword
5. Make publication available
6. Make publication unavailable
7. Remove publication
8. Exit

Enter option: 2

Principles of Programming by Genie Ruskin (1999)

#10002

AVAILABLE

Memory Management Demystified by Fenton Beverley (1970)

#10003

AVAILABLE

Using version control like a professional by Hope Marks (2020)

#10004

NOT AVAILABLE

The History of Databases - Part 2 by Morty Tittensor (2013)

#10005

AVAILABLE

1. Add publication
2. View all publications
3. Find by author
4. Find by keyword
5. Make publication available
6. Make publication unavailable
7. Remove publication
8. Exit

Enter option: 8

Shutting off...