CS 116

Lab Assignment # 3: Sorting and searching a Library Database

- **Points: 3**
- **Submission**
    - **Deadline: Tuesday 02/23 11:59 PM**
    - **Submit on Blackboard under assignment "Lab3". Please make sure that you click the "Submit" button and not just "Save".**
- **Late Submission Policy**
    - **You can do a late submission until Thursday 02/25 11:59PM with a 5% penalty on the total points for this assignment.**
    - **After that solutions will be posted and no submission will be accepted**
- **Early Submission**
    - **You can also get 5% extra point (on your score on the assignment – Note this change from previous assignment!) for early submission if you submit by Monday 02/22 11:59PM.**
- **Getting help**
    - **From instructor during office hours in SB228 or by email.**
    - **By seeing one of the TAs during the listed TA office hours in room SB108. (check the course website http://cs.iit.edu/~jkorah/cs116/)**
    - **By visiting the ARC (Academic Resource Center).**
- **Academic Dishonesty Policy**
    - **Working with a partner: You can work with a partner as assigned by the instructor. Otherwise this should be considered individual work.**
        - **Even if you are working with a partner, you and your partner are required to make individual submissions.**
    - **Please note: In case two submissions are declared identical (and if you are not supposed to work together) the excuse: we worked together, does not hold and both submission will be treated according to ethics rules.**

Objectives:

The list below indicates the Java concepts needed for this exercise.

1. Calling static and non-static methods from a static method.
2. Reading data from a File.
3. Using while loops.

4. Develop the logical algorithm that will accomplish the required task.
5. Using methods of the String class
6. Saving objects in an array.
7. Packaging a class
8. Using enumerations.
9. Binary Search
10. Sorting algorithm

THEORY: Lexicographic Sorting

Just like numbers can be sorted, words or strings can also be sorted. The sorting of strings in an increasing or decreasing order is called lexicographic or dictionary based sorting. Therefore when we sort the words {cat, ball, bat,bats} in increasing order, we should get {ball,bat,bats,cat}. "ball" is smaller or lower than "bat" in the sorted list – when we compare their individual letters sequentially, "b" and "a" are identical. However, "l" is lexicographically smaller than "t". Hence "ball" is lower than "bat". Remember strings can also contain numbers. For example, when comparing "AX93" and "AX83", "AX83" is lexicographically lower than "AX93".

PROGRAMMING TASK : Building a database for a library

- **Please read all steps carefully first, and then start coding.**
- **You can use the solutions to practice exercises and any other help including lecture presentations and your text book.**
- **The current directory where your source code files are located should be a folder named <LastName>-<FirstName>-Lab3.**

In this programming task, you will further develop the library database from the Lab Assignment 2. Specifically, you will add code to sort the records in the

NOTE: You solutions for Lab 3 should have ALL the functionality that was specified in Lab 2.

NOTE: You can either continue to use the code that you developed for Lab assignment 2

OR

you can use the solution for Lab 2 that will be posted on the course website on Feb 18th . (recommended)

Programming Task specification:

1. **Change the BookRecord class** [0.25 pts] BookRecord.java is the service class of the java application. Add appropriate class attributed and methods for the following specifications:

a. **A string variable** to store the "Tag" information for the books. A tag is a UNIQUE 8 digit alpha numeric code (combination of letters and numbers) for each book. This information is provided in the input text file.

b**. An int variable** to store the length (number of pages) in the book. This information is provided in the input text file.

c. If you are using a non-default constructor, then modify it to add the tag and book length to the list of parameters

d. toString(): modify this method to also print out the tag and page length information.

e. equals(): You will modify the equals() class method that compares the instance variables of the two objects . Now, you will compare the tag and page length of the two books along with the values of the title, list of authors and genre of two objects to determine their equality. Again, DO NOT compare the record id.

h. Accessor and mutator methods: implement appropriate accessor and mutator methods for the tag and page length class attributes.

As before, keep the service class in a package called library.service.classes.

2. Change the client class: Place the client class in library.client.classes. You will make the following changes to the client class:

a. Modify the file read functionality[0.25pt]: The file text "books.txt", provided along with the assignment has lines with the following format:

title:genre:author-1,author-2,….author-m:tag:no-of-pages

Modify the code from lab 2 so that it also reads in the tag and no. of pages, along with the other values, and then creates the BookRecord objects. Remember that the text file may have duplicate records (records with the same title, authors, genre, tag and page length). As with Lab 2, your code should filter out the duplicate records.

b. Implement a class method called sortString() with the following signature:[0.75pt]

public BookRecord [] sortString(BookRecord [] myArray, int noRecords)

The method takes in the array of BookRecord object you wish to sort and also the number of objects in the array. You will implement the selection sort algorithm in this method to sort the BookRecord object array. Once the books.txt file has been read and the array of BookRecord objects has been created, you will call the sortString() method. This method will use selection sorting algorithm to re-arrange the BookRecord objects in the array in a lexicographic increasing order with respect to their tag values. Hint: Lexicographic selection sorting is not difficult as long you can figure out how to implement the comparisons of two strings. Go through the Java API for String class and figure out the method that you can use to do String comparisons.

An important point to be noted is that the array of BookRecord objects may not be completely filled. Therefore you need to set the indexes of the sorted and unsorted sub-arrays in your selection sorting algorithm appropriately.

c. Implement a second class method called sortPages() with the following signature [0.5pt]

public BookRecord[ ] sortPages(BookRecord [] myArray, int noRecords)

This method will take in an array of BookRecord objects and sort the objects in an increasing order with respect to their page length values. You will use a selection sort algorithm to accomplish this. The purpose for implementing this method is explained below.

d. Implement a third class method called searchTag() with the following signature [0.75pt]

public void searchTag(String tag)

This method will take in a tag value as a parameter. It should search the BookRecord objects in the array, using Binary Search, for a particular tag value and display the record (call the toString() method of the record). If such a record does not exist in the array, the method should print "No match found" and return. **You should implement binary search to find the object in the array. If you implement the sequential search algorithm, you will NOT get partial points!!**

c. Modify the interactive menu[0.5pt]: After reading the records from the text file and sorting it according to their tag values, you should prompt the user with a menu with the following options:

Select an option:
Type "S" to list books of a genre
Type "P" to print out all the book records
Type "T" to search for a record with a specific tag
Type "Q" to Quit

- Options "S": Remember from Lab 2 assignment that when the user selects this option, the set of genres are listed and the user is prompted to type in a particular genre. The books belonging to the selected genre are listed. You will modify the implementation from lab 2 to display the books of a particular genre in the increasing order of their page length. This means that books with shortest length are listed first, followed by the second shorted book and so on. You will use the sortPages() method to implement this feature.
- If the user types in the option "P", you will list all the book records in the array. Since you sorted (see description 2b above) the book records in an increasing (lexicographic) order, the book records should also be displayed in that order.
- **Option "T" is a new option that you will add to the menu from the lab assignment 2. This options allows the user to search for a book record that has a particular tag value. Remember the tag value is unique. When a user selects this option, your code should prompt the user to enter the tag value that he/she is looking for. Once the user enters the search term, your code will call the searchTag() method which will then display the results.**
- If the user types in the option "Q", you will quit the program. Remember you will continue to display the menu options to the user and execute the selections, until he/she selects "Q" to exit the program.

Submission instructions

- In your submission you must include

  a. The source code files and the compiled files for the program.
- Zip all files and name the zip file using your last name followed by your first name followed by the name of the assignment

  i.e. Doe_Jane_Lab3.zip
- Upload the file on assignment folder: Lab3 on Blackboard.

Sample Output:

```
>java library.client.classes.library books.txt 5
```

```
Resized the array from 5 to 10
Resized the array from 10 to 15
Found a duplicate
===================================
Tag:MNBV3456
Title:English landscaping and literature, 1660-1840
Genre: GENRE_LITERATURE
Authors:  E. Malins
No. of Pages: 980
===================================

Resized the array from 15 to 20
Found a duplicate
===================================
Tag:NJKG7456
Title:Nikola Tesla
Genre: GENRE_HISTORY
Authors:  Sean Patrick
No. of Pages: 987
===================================

Found a duplicate
===================================
Tag:HJKG2342
Title:Microfabricated microneedles, a novel approach to transdermal drug deliver
y
Genre: GENRE_ENGINEERING
Authors:  S. Henry D. V. McAllister M. G. Allen
No. of Pages: 378
===================================

Select an option:
Type "S" to list books of a genre
Type "P" to print out all the book records
Type "T" to search for a record with a specific tag
Type "Q" to Quit
S
Type a genre. The genres are:
GENRE_HISTORY
GENRE_SCIENCE
GENRE_ENGINEERING
GENRE_LITERATURE
GENRE_SCIENCE
===================================
Record No:10015
Tag:KJHG8909
```

```
Title:Gene Ontology
Genre: GENRE_SCIENCE
Authors:  M. Ashburner C. A. Ball J. A. Blake D. Botstein H. Butler
No. of Pages: 90
===================================

===================================
Record No:10010
Tag:HGHB8909
Title:Free radicals in biology and medicine
Genre: GENRE_SCIENCE
Authors:  B. Halliwell J. M. C. Gutteridge
No. of Pages: 234
===================================

===================================
Record No:10008
Tag:BGHF8976
Title:The comparative method in evolutionary biology
Genre: GENRE_SCIENCE
Authors:  P. H. Harvey M. D. Pagel
No. of Pages: 234
===================================

===================================
Record No:10011
Tag:LKJH2345
Title:Electron transfers in chemistry and biology
Genre: GENRE_SCIENCE
Authors:  R. A. Marcus N. Sutin
No. of Pages: 890
===================================

Select an option:
Type "S" to list books of a genre
Type "P" to print out all the book records
Type "T" to search for a record with a specific tag
Type "Q" to Quit
T
Input the tag of the book you want to search for:
LKJH2345
Found a match
===================================
Record No:10011
Tag:LKJH2345
Title:Electron transfers in chemistry and biology
```

```
Genre: GENRE_SCIENCE
Authors:  R. A. Marcus N. Sutin
No. of Pages: 890
===================================

Select an option:
Type "S" to list books of a genre
Type "P" to print out all the book records
Type "T" to search for a record with a specific tag
Type "Q" to Quit
T
Input the tag of the book you want to search for:
LKJH2344
No match found
Select an option:
Type "S" to list books of a genre
Type "P" to print out all the book records
Type "T" to search for a record with a specific tag
Type "Q" to Quit
P
===================================
Record No:10000
Tag:ABEW2345
Title:Thomas Jefferson and the Tripoli Pirates
Genre: GENRE_HISTORY
Authors:  Brian Kilmeade Don Yaeger
No. of Pages: 234
===================================

===================================
Record No:10008
Tag:BGHF8976
Title:The comparative method in evolutionary biology
Genre: GENRE_SCIENCE
Authors:  P. H. Harvey M. D. Pagel
No. of Pages: 234
===================================

===================================
Record No:10007
Tag:FDST9878
Title:Climate and atmospheric history of the past 420,000 years
Genre: GENRE_HISTORY
Authors:  J. R. Petit  J. Jouzel D. Raynaud N. I. Barkov J. M. Barnola
No. of Pages: 675
===================================
====================================
Record No:10010
Tag:HGHB8909
Title:Free radicals in biology and medicine
Genre: GENRE_SCIENCE
Authors:  B. Halliwell J. M. C. Gutteridge
No. of Pages: 234
====================================

====================================
Record No:10001
Tag:HGNH4567
Title:Component-oriented programming
Genre: GENRE_ENGINEERING
Authors:  C. Szyperski J. Bosch W. Weck
No. of Pages: 456
====================================

====================================
Record No:10005
Tag:HJGF7645
Title:A history and theory of informed consent
Genre: GENRE_HISTORY
Authors:  R. R. Faden T. L. Beauchamp N. M. King
No. of Pages: 654
====================================

====================================
Record No:10002
Tag:HJKG2342
Title:Microfabricated microneedles, a novel approach to transdermal drug deliver
y
Genre: GENRE_ENGINEERING
Authors:  S. Henry D. V. McAllister M. G. Allen
No. of Pages: 378
====================================

====================================
Record No:10006
Tag:JHGF9089
Title:The Feminist Companion to Literature in English Women Writers From the Mid
dle Ages to the Present
Genre: GENRE_LITERATURE
Authors:  V. Blain P. Clements I. Grundy
No. of Pages: 767
```

```
===================================

===================================
Record No:10013
Tag:JHKG2343
Title:Device electronics for integrated circuits
Genre: GENRE_ENGINEERING
Authors:  R. S. Muller T. I. Kamins M. Chan P. K. Ko
No. of Pages: 654
===================================

===================================
Record No:10015
Tag:KJHG8909
Title:Gene Ontology
Genre: GENRE_SCIENCE
Authors:  M. Ashburner C. A. Ball J. A. Blake D. Botstein H. Butler
No. of Pages: 90
===================================

===================================
Record No:10011
Tag:LKJH2345
Title:Electron transfers in chemistry and biology
Genre: GENRE_SCIENCE
Authors:  R. A. Marcus N. Sutin
No. of Pages: 890
===================================

===================================
Record No:10004
Tag:MNBV3456
Title:English landscaping and literature, 1660-1840
Genre: GENRE_LITERATURE
Authors:  E. Malins
No. of Pages: 980
===================================

===================================
Record No:10003
Tag:NJKG7456
Title:Nikola Tesla
Genre: GENRE_HISTORY
Authors:  Sean Patrick
No. of Pages: 987
===================================
===================================
Record No:10009
Tag:UYHG1223
Title:Human-computer interaction
Genre: GENRE_ENGINEERING
Authors:  J. Preece Y. Rogers H. Sharp D. Benyon S. Holland
No. of Pages: 889
===================================

===================================
Record No:10014
Tag:YUTY9098
Title:An outline of English literature
Genre: GENRE_LITERATURE
Authors:  G. C. Thornley G. Roberts
No. of Pages: 89
===================================

Select an option:
Type "S" to list books of a genre
Type "P" to print out all the book records
Type "T" to search for a record with a specific tag
Type "Q" to Quit
Q
Quitting program
```