

CENG 351
Data Management and File Structures
Fall 2018-2019
Programming Assignment 2

Due date: 02.01.2019, 23:59

1. **CENG LIBRARY:** To provide a better learning environment, Computer Engineering Department of Middle East Technical University has decided to build a digital library that contains all the course materials.

The project has started long ago, due to the shortage of funds however, the coders quit their job before finishing the books section of the library. At the end, as creative(!) teaching assistants of the department we come up with a free solution: Giving this important task to our prospective young talents (i.e. students) as their homework.

2. **GOOD NEWS:** The project is almost finished and added to COW. Only four functions in the file *LibraryTree.java* are needed to be written.

2.a. *public void addBook(CengBook book)*
2.b. *public CengBook searchBook(Integer key)*
2.c. *public void printPrimaryLibrary()*
2.d. *public void printSecondaryLibrary()*

3. **THE WORK:**

3.a. Books are to be stored in two B+ trees. One is a primary B+ tree, using non-negative integer *CengBook.key* as the comparison element. We know that keys are unique and can not appear more than once in our database. The other one is a secondary B+ tree, using *CengBook.year* as the comparison element. Note that several books may have the same year, but the B+ algorithms you have learnt in class do not consider duplicates. Therefore, we use year and *CengBook.key* together as a composite key in our secondary B+ tree.

3.b. The function *addBook* will be used to construct both primary and secondary B+ trees. It does not give any output. The input specifications are given in the specifications section.

3.c. The function *searchBook* will be used to find the corresponding book using the given key in the primary tree. This function should print the traversed path and the book that is being searched. If the book is not in the tree, the function still prints the traversed path but not the book since it does not exist in the database. Instead of the book, it will print a sentence to state that the book does not exist. The input and output specifications are given in the specifications section.

3.d. The functions *printPrimaryLibrary* and *printSecondaryLibrary* print the corresponding B+ trees in depth-first order. The input and output specifications are given in the specifications section.

4. THE FILES:

There are 17 files in CengLibrary project:

4.a. Graphical User Interface: CengLibrary has a primitive GUI to help you to check your work visually and see your mistakes. They will only be executed if the program calling parameter `<guiOptions>` is set to greater than 0. Note that grading will be done without using GUI. Files used only for GUI are:

CengGUI.java, GUIInternalPrimaryNode.java, GUIInternalSecondaryNode.java, GUILevel.java, GUIPrimaryLeafNode.java, GUISecondaryLeafNode.java, GUIDreeNode.java, WrapLayout.java

4.b. Main Files: The other 9 files are used to parse the input, and determine the structure of our database. These are:

CengLibrary.java, CengBook.java, LibraryNode.java, LibraryNodeLeaf.java, LibraryNodePrimaryIndex.java, LibraryNodeSecondaryIndex.java, LibraryNodeType.java, LibraryParser.java, LibraryTree.java

4.c. Files to be Changed: There are 3 files that you are allowed to make changes on. Of course you can experiment on the other files, but at the end of the day you should upload only these three files. So, it is strongly suggested to redownload the project and execute last tests on clean files. Files to be changed are:

LibraryTree.java, LibraryNodePrimaryIndex.java, LibraryNodeSecondaryIndex.java

5. SPECIFICATIONS:

5.a. Compile and Execute:

(Every file is needed to be in the same folder-no subfolders allowed)

```
>javac *.java
>java CengLibrary <order> <guiOptions> <guiFileName>
```

order: Order of the B+ tree.
(ie. Nodes should have n elements s.t. $orderCount \leq n \leq 2 \cdot orderCount$)

guiOptions: The option for GUI.
0: GUI is disabled.
1: only primary tree is used in GUI
2: only secondary tree is used in GUI
3: both trees are used in GUI

guiFileName: The path for the file to be used in GUI. Only parsed if GUI is enabled.

The file consists of the values for books in lines of the form:
Key|Year|Name|Author

A sample guiFile, "sampleGUIFile.txt" is also given.

Example with GUI:

```
>java CengLibrary 1 3 sampleGUIFile.txt
```

Example without GUI:

```
>java CengLibrary 2 0
```

5.b. Commands:

It may seem complicated at first glance, some example input and output pairs are also given as separate files to help you understand the input-output specifications easier.

add: This command is used to add books to the library.

```
>add|Key|Year|Name|Author
```

Example:

```
>add|2|2003|Database Management Systems 3. Ed.|Raghu Ramakrishnan
```

search: This command is used to search a book with a given key in primary B+ Tree.

```
>search|Key
```

Example:

```
>search|2
```

The output should contain all the visited internal nodes with their keys and the search result in the following form:

```
<index>
key
...                // all the keys in this internal node
key
</index>
...                // some more internal nodes to be traversed
<data>
<record>key|year|name|author</record>
</data>
```

Successful Search Example:

```
<index>
5
11
</index>
<index>
13
19
</index>
<data>
<record> 15|1991|AbstractBook|NoOne</record>
</data>
```

UnSuccessful Search Example:

```
<index>
5
11
</index>
<index>
13
19
</index>
No match for 25
```

print1: This command is used to print the primary tree in depth-first order.

```
>print1

<index>
key
...
key
</index>
...
<data>
<record>key|year|name|author</record>
...
<record>key|year|name|author</record>
</data>
...
```

print2: This command is used to print the secondary tree in depth-first order. (Note that instead of key, year is used as index perimeter)

```
>print2

<index>
year|key
...
year|key
</index>
...
<data>
<record>key|year|name|author</record>
...
<record>key|year|name|author</record>
</data>
...
```

quit: Just terminates the program. Already implemented.

```
>quit
```

6. SUBMISSION:

6.a. Archive three files with the command:

```
>tar -cvf <yourStudentID>.tar.gz LibraryTree.java, LibraryNodePrimaryIndex.java,  
LibraryNodeSecondaryIndex.java
```

For grading, your files will be extraced with the command:

```
>tar -xvf <yourStudentID>.tar.gz
```

(Since the codes will be black-box tested, if the extraction results within a folder, your work can not be graded)

6.b. Before submission, be sure to make a last test. Download clean codes from Cow. Copy your tar file into the directory that clean codes rest to replace 3 files with your work, and test on inek machines (**without** using any IDE)

```
>javac *.java  
>java CengLibrary <order> 0
```

6.c. Make sure your output format exactly matches the specifications. Your program should print the outputs on the terminal.

6.d. All the generic rules that are stated in the course syllabus also apply for this programming assignment.