CSE 2105 – Data Structures

2020 – 2021 Fall Semester Project

Write a program that performs the corresponding operations through the below menu, according to the selected option.

> *0. Exit*
> *1. Add student*
> *2. Delete student*
> *3. Find student*
> *4. List all students*
> *5. List distinct names*
> *6. List name counts*
> *7. About*
> *Enter your selection:*

You should ask until the user enters his selection correctly (between 0-7). When the user enters an option between 1-7, corresponding operation in the menu should be performed and the menu should be printed again after the operation ends. If the user enters 0, you should ask whether he/she really wants to exit the program. If he/she is sure, finish the program; otherwise, display the menu again. If the user enters an invalid option (not between 0-7), you should print a warning message and display the menu again.

Each student should (at least) has the following properties:

- student id
- name
- surname

To store students, you must implement <u>your own</u> "list", "binary search tree" and "hash" structures. (Instead of using pre-written Java API Class you should develop your own data structures). You should write your own code for data structures. You are allowed to use lab source codes that we examined so far; however it is <u>preferable</u> to write your own codes for your data structure. Your implementations should be <u>efficient</u> as possible.

Clarifications

1. This option adds the student to the list, tree and hash structures. You should get from the user student id, name and surname before adding student. New student should be added to the appropriate positions of the structures according to its student id (i.e student id is the key).

2. First get the <u>student id</u> from the user and (if found) delete this student from all structures.

3. First get the student id from the user and (if found) and print this student's properties to the screen. Also print the number of search levels (how many hops) that you found the student from the list, tree and hash.

4. If the user selects option 4, first ask him from which data structure he/she want to operate and print all the student's information according to selection by traversing related structure.

5. Print to the screen all the distinct "names" (use only names, not surnames) from your data structures. (i.e. not list duplicate names to the screen).

6. Show that how many of each "name" included in the data structures. (use only names, not surnames). For example you can print like following format:

emre : 2
sevcan : 3
zeynep : 1

7. Print the author(s) of the program (your student id, name and surname).

P.S.: You must prepare your project yourself (single person). You have to submit a "report" **(report is very important!, especially you have to explain your data structures usages, how and why)**, and "Java code" (your classes) of your program. (You can use any IDE you want.) Please upload your compressed(zip/rar) file (that includes <u>your report (as "pdf" file) and Java source code files (with ".java" extension) – not all the project files!</u>) to the **Microsoft Teams** page of the course to the appropriate area (<u>projects that are sent via e-mail or other different ways of sending, will not be accepted!</u>) before **15 January 2021, Friday.** (**Projects that not uploaded to Microsoft Teams page of the course until this time, will definitely not be accepted!**).