Homework 1, CSCI 430/530. Fall 2023.

This is an individual assignment.

Due date: Friday Sep 1

Note: *These programs have to be finally compiled and run on the Linux VMs (see video on D2L). Software built on an IDE sometimes does not port to a JVM; these will be tested on the VMs so make sure you test everything on the VM before submitting*

Q 1. Implement the following in Java:

a Create a data type `Person` with three string fields: `firstName, lastname` and `id`. The field `id` is unique to each person. Add accessor methods, a constructor with three parameters, and the `toString())` method.

b Create a class `MyMain` for the main program. Within the `main()` program, instantiate a linked list (use the Java `LinkedList` class) . Add the following functions(methods) (static methods within the class `MyMain`) that the main program will invoke:

- `store(<input stream>, <linked list>)` that reads the data for several persons from the input stream and stores the data in the linked list.
- `display(<output stream>, <linked list>)` that writes the data for all person objects in the linked list, on the output stream, one per line.
- `find(string sid, <linked list>)` that returns the index of the person object in the linked list, that has the same `id` value as `sid` (return -1 if no such person exists). This can be done as a simple search that goes sequentially through all the objects in the linked list.
- Create a data file with data for a few person objects.
- In the main method, call the `store()` and `display()` methods to read the data and display it. Invoke the find method a few times. Compile and test your code.

c Construct a different version of the above program as follows:
(1) create a data type `PersonList` that uses the Java `LinkedList`, and supports the methods(re-written appropriately) from part b.
(2) instantiate the `PersonList` object in the main program and invoke its methods (as per the appropriate syntax), so that program produces the same output as the one in part b.

d (Individual Reflection) Which of the two versions (part b and part c), in your opinion, represents a process-centered approach? Which one represents a data-centered approach? What additional features, in your opinion, if added to the `LinkedList` class might reduce the coding effort in part c? **Justify all your claims.**

Submission: Within your folder (named by your starid) inside the `StudentWorkFolder in CourseFiles (Y: drive)` create a folder `Hw1Q1`. For each part, do as follows:

- Question 1b. Within `Hw1Q1` create a folder `Hw1Q1b`. Upload the source code, data files, and the `.class` files from centos to this folder.
- Question 1c. Within `Hw1Q1` create a folder `Hw1Q1c`. Upload the source code, data files, and the `.class` files from centos to this folder.
- Question 1d. In the D2L dropbox, upload a word file with all your answers to questions.