CS2401 – Week 8-9

With this lab assignment, you are going to practice using someone else's code (part of the code is given to you). You will practice recursion and the manipulation of linked lists. We hope you enjoy this assignment!

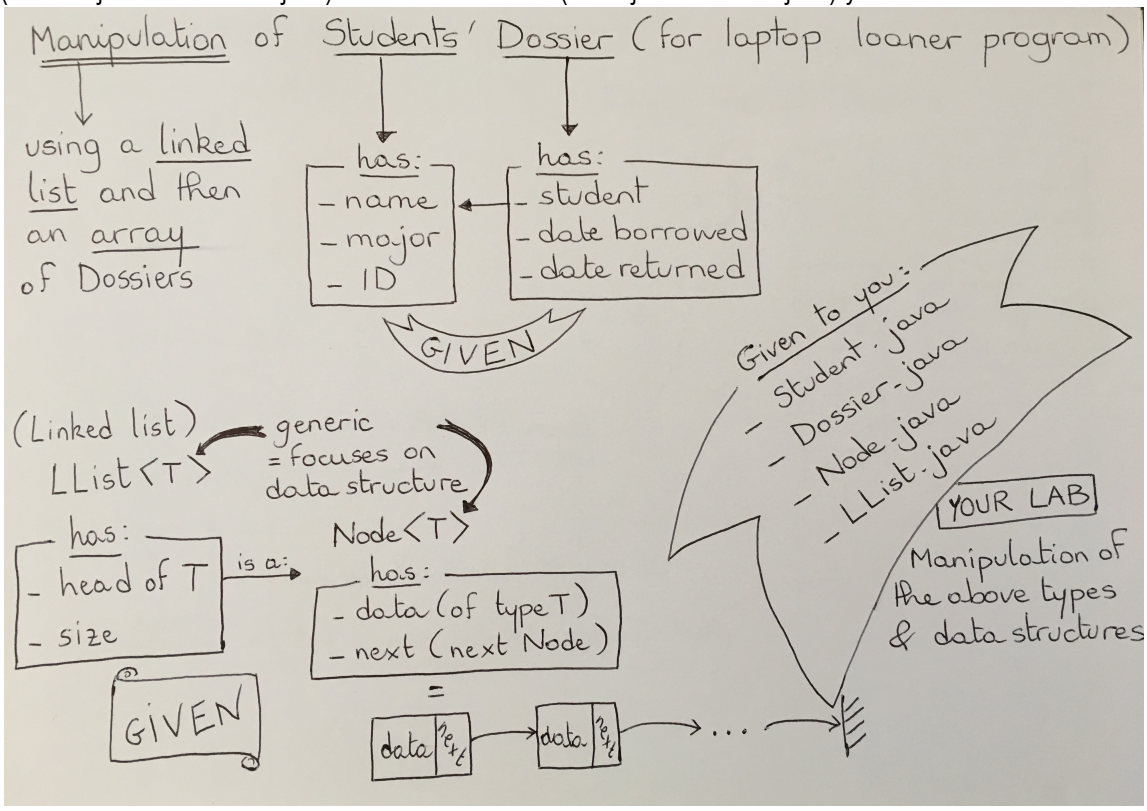
What is the scenario?

You are put in charge of managing a laptop loaner program at UTEP. Each day you receive information pertaining to this program. Information passed on to you can be of two types: dossiers about students who just entered the program, and updates on dossiers. When you receive information about a dossier, you create a new dossier and add it to a list of dossiers. When you receive an update, you have to look for the dossier to update in the list and you update it. At the end of the day, you transfer all information from your list into an array (easier to manipulate). You then need to be able to sort this array to make sure that all non-returned dossiers are in the first part of the array, and the returned dossiers are in the second part of the array.

So, how will this work? What will you have to do?

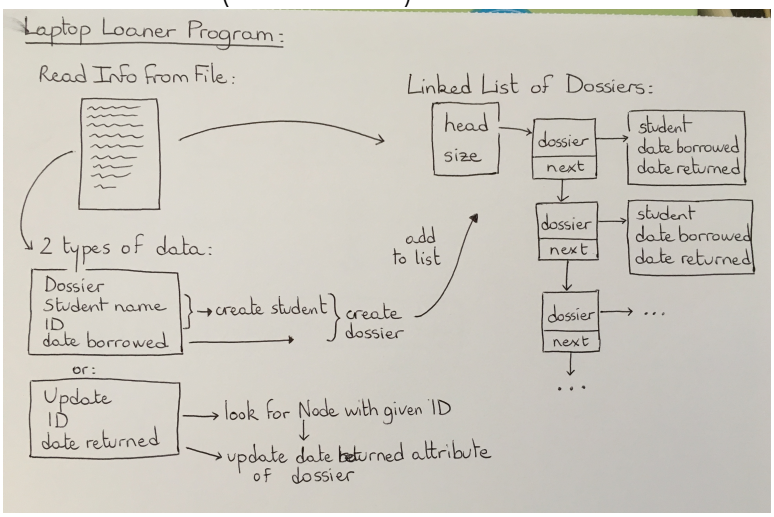
For the sake of this lab assignment, we will pretend that your data about dossiers and updates lies in a txt file. You will read it from there. See loanerInfo.txt for an example of such a txt file.

Four files are provided to you so that you can focus on the work at hand and not worry about the types (Student.java and Dossier.java) and data structures (Node.java and LList.java) you have to handle.

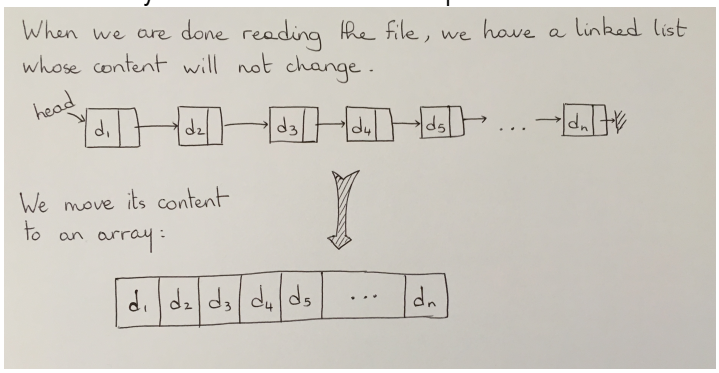


Your work resides entirely within the file LoanerProgram.java. In this file, you are expected to implement the following 4 methods.

- Method readStudentInfo:
  - Takes the name of a file (String filename)
  - Returns a linked list (LList<Dossier>) that contains the data read from the file filename



- Method findStudentWithGivenID:
  - Takes a node of dossier (Node<Dossier>) and a student ID
  - Returns the node of the input linked list whose student has the input ID
  - Note: this method is a submethod of readStudentInfo and is needed for when you want to update a dossier based on a student ID
  - Node: This method is expected to be RECURSIVE
- Method createArrayOfLoanerDossiers:
  - Takes a linked list of dossiers
  - Returns an array of all the dossiers in the input linked list



- Method sortByNotReturned:
  - Takes an array of dossiers
  - Arranges the input array in such a way that all non-returned dossiers are now at the start of the array and all returned dossiers are at the end of the array
  - Note: this method is expected to be inspired by the QUICK SORT algorithm

You then have to complete the main method, as prompted in 5 steps. In these 5 steps, you are expected to use the above methods as much as possible and relevant.

Grading:

5 pts	Code is properly indented
10 pts	Code is well documented
20 pts	Method readStudentInfo
15 pts	Method findStudentWithGivenID
10 pts	Method createArrayOfLoanerDossiers
20 pts	Method sortByNotReturned
20 pts	Your main method follows specifications and works well.

Due date: October 25th at 11:59pm

What to submit?
You should submit only LoanerProgram.java.

How to submit?
You should follow your TA submission guidelines: see piazza Resources section, under Lab Submissions.

Failing to follow submission instructions and guidelines will result in up to 15 points off your overall grade in this lab. So please pay attention.

Extra Credit Opportunities

If you decide to take on this extra credit opportunity, here is what you can do. Note that you do not have to do it all. There are multiple options to do just that: give you options.

[For 10 more points]
Expand the sortByNotReturned method into a new method sortByNotReturnedByDate.
Your new method should still aim to separate the non-returned from the returned dossiers. However, each part (non-returned and returned) should be sorted by oldest to newest date borrowed. Your method should be a quik-sort inspired algorithm

[For 10 more points]
You should design a method called LatestBorrowed that will take a linked list L of dossiers and an integer n, and return the dossiers of the latest n borrowed laptops.

[For 5 more points]
You should design a method called hasBorrowed that will take a student ID and return whether the given student has borrowed a laptop.

[For 10 more points]
You should design a method that identifies, among the dossiers of returned laptops, the student who kept a laptop the longest. This method should take an array of dossiers and return a student.