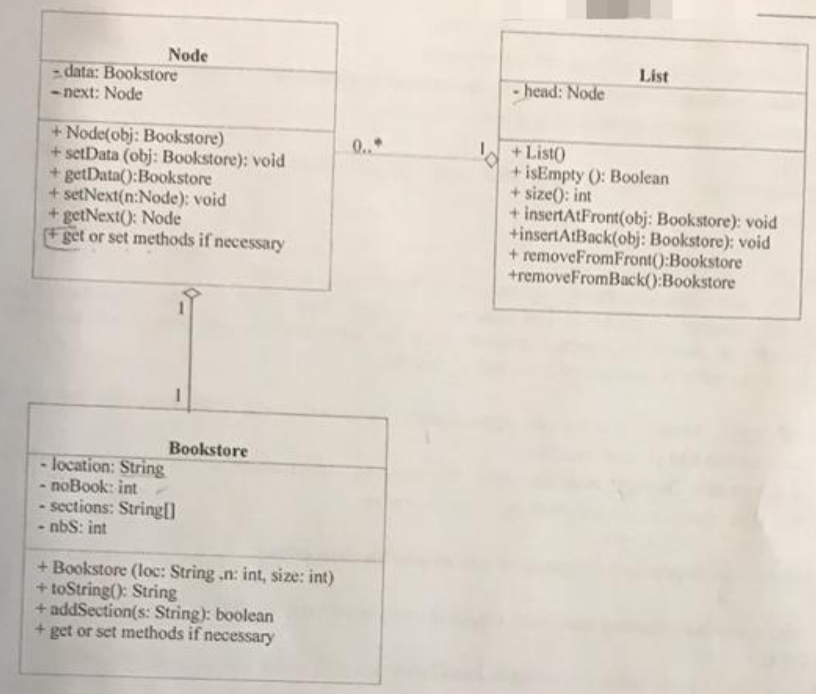
King Saud University
College of Computer and Information Sciences
Department of Computer Science

CSC 113 – Java II
Final Lab
Second Semester 1440/41 H

Name:
Lab#.



Problem Description:

Assume the city of Riyadh wants to keep track of all the bookstores in the city. A bookstore is considered as a "Big bookstore" if the number of books is 1000 or more, otherwise it is a "Regular bookstore". The city of Riyadh allows a maximum 5000 books in each bookstore. The city of Riyadh uses a **List of Bookstores** for that purpose as shown in the previous UML Diagram.

Notes and Instructions:

1. **Important:** Create a folder in the desktop with your name (FirstName_LastName) and save all your work on it.
2. The classes *List* and *Node* and file *"MoreBookstore.txt"* have already been implemented for you. You can find them in the folder "CSC113 final lab" on the C:\ drive.
3. You **must** make use of existing methods when appropriate.
4. The **order** of the list is not important.
5. After you finish, compress your folder and submit it in the LMS under "Lab" page.

Implement the following:

1 . **Implement Class Bookstore:**

Attributes

- ***location: String***: The location of the bookstore.
- ***noBook: int:*** Number of book in the bookstore.
- ***sections: String[]:*** An array containing the name of sections of the bookstore.
- ***nbS: int:*** the current number of sections in the bookstore.

Methods

- ***Bookstore(loc: String, n: int, size: int):*** A constructor to initialize the bookstore information:
  - *loc: is the location*
  - *n: is the number of book.*
  - *size: is the max number of sections the* bookstore *can have.*

  If the number of books is less than zero or more than 5000 an *InvalidNumberofBook* exception should be generated and caught in the same environment. You should allow the user to re-enter the number of book until it is valid.

- ***toString()*: String:** Returns an appropriately formatted string containing the bookstore information and all *section names.*
- ***addSection(s: String): boolean***: Adds the received section name to *section* array. Returns true if section is added successfully, false otherwise.

2 . Define a new **unchecked** exception of type ***InvalidNumberofBook.***

3 . Create a new **Application class** with a main method to perform the following:

NOTE:

Your main() should **catch the possible exceptions** that may occur during processing files.

a) **Create** a list for Riyadh bookstores named ***BoSList***.

b) **Add** new three bookstores to the list of Riyadh's Bookstores. Ask the user to enter: location, number of books, size. Also, ask the user how many sections he would like to add, then read their names.

c) **Read** the information of more bookstores from the **text file:** *"MoreBookstore.txt"*. Add the bookstores to the list of Riyadh bookstores. In the text file, the information for each bookstore is on three lines containing: location, noBook, and size in turn.

d) **Remove** one bookstore that is located at *"KSU"*, then print an appropriate message.

e) **Add** a new section named "Computer" to all **Big bookstores**.

f) **Copy** all **Regular bookstores** from the list to the first empty location in a new array named ***RArray***. Then, **print** the number of **Regular bookstore**.

g) **Create** an object file named *"RegularBookstores.dat"* and store all **Regular bookstores** a once.

h) Print the complete information of all bookstores in Riyadh.