BRAC University

Department of Computer Science and Engineering

CSE220: Data Structure

Task 1

Write a Java program that will keep the records of BRAC University Books.

Book **Class**

The program should contain a "Book" class with the following fields:

Name

Id

Author

Publisher

Book **Records Class**

You need to write a "BookRecord" class. In the "BookRecord" class, you have to declare an array "BookArray" of the instance of the "Book" class. The "BookRecord" class contains the following methods that can manipulate the "BookArray" with the instances of the Book class:

AddRecord: The *AddRecord* method adds a new Book record to the BookArray at the next available empty index of the array. If the array is full, then you should resize the array and add the new record. You will ask the user to give the name, author and publisher of the book, but not the id. Id will be automatically generated and you have to make sure that the id will be unique.

PrintRecord: The *PrintRecord* method will create a new array to sort the BookArray in alphabetical order according to their names. If there is a tie (i.e., if multiple Books have the same name), then sort them according to their id's. After sorting, this method prints the contents (Name,Id, Author, Publisher) of the array, each record in a line. Do not change the base array as we wanted to keep the array in a way it was initially filled in.

DeleteRecord: The *DeleteRecord* method asks the users to enter the Id number of a Book to be deleted. Then, delete that record and reports (by printing a message) the success and failure of the deletion. After completion of a deletion, your program will show a message "ID #### is deleted". If the given Id does not exist, your program will show a message "ID #### does not

exist". Once, a record is deleted, the DeleteRecord method also shift left all records that are to the right of the deleted record.

Edit Record: The *EditRecord* method asks the user to enter a Book Id and edits its corresponding information (name, publisher). If the given Id does not exist, your program will show a message "ID #### does not exist". EditRecord method should be able to edit one or more information of that Book.

Tester Class

Your program should have a "TesterClass" containing the main method. In the main method, your program should print the following message:

Enter 1 to Add a Record

Enter 2 to Print the Records

Enter 3 to Delete a Record

Enter 4 to Edit a Record

If the user chooses 1, then the program will add a new Book record to the BookArray. If the user chooses 2, then the program will print the Book list in alphabetical order. If the user chooses 3, then the program will delete the records of the given Id. If the user chooses 4, then the program will edit the records of the given Id.

Upon the completion of one operation (choosing one option among 1 – 4), your command line menu will ask the user whether s/he wants to continue the program. If the user press **"Y"/"y",** then the user will be given menu 1 – 4 again. Otherwise the user will quit the program.

Task 2

Update your code for task 1 so that every time user tries to end the program, it asks whether the user would like to save everything s/he has done in the current iteration or not. We are talking about 'save and exit' and 'exit' features. If the user decides to 'save and exit', you have to store every data related to record of the books your program holds in a text file so that when the user re-opens the program, s/he gets an option to load the backed up data or to initiate a fresh start.

Task 3

Now, update your *AddRecord* method so that user can record multiple books at once from a given text file (similar to the one you will be using to back up your data in task 2). The previous 'one book at a time' option will still be available. Design the method in a way so that the user get to decide whether s/he wants to record multiple books at a time from a text file or on book at a time.