KING SAUD UNIVERSITY COLLEGE OF COMPUTER AND INFORMATION SCIENCES Computer Science Department		
CSC 113: Introduction to Programming II	Homework#2	2nd

Write the code of the following classes:

Student

Attributes:

- **name:** name of the student.
- **id:** id of the student (int).
- **gpa:** the gpa of the student (double)**.**

Methods:

- **Student(String name, int id, double gpa)**
- **toString () :String** : returns a string containing all information of the student.

You can add any setters or getters if you need them

NotFoundException

It is a user defined exception that extends **Exception.**

Methods:

- **NotFoundException(String msg)**

Application

Methods:

- **addStudent (Student s):** void : adds the given student to the first empty location in the array ***studentList***.

  Add without checking the index beforehand. In case the array is full, you should print a message that indicates the unsuccessful addition operation while handling the caused exception **in the main**.

- **removeStudent(int id):** void : deletes the student with the specified id (replace the deleted one with the last element in the array or shift left). If the given **id** is not found, throw a **NotFoundException.**

- **Main() :**
  - Create an array of **Student** (***studentList***) that can contain 100 students at most.
  - Read the students' information from ***student.txt*** create an object for each student and add them to ***studentList*** .
    - The data written in ***student.txt*** is in the following form:

      name-id-gpa

      each line hold a student's informaion (**name**, **id** and **age** separated by '-')
  - Read the students objects from ***student.data*** and add them to ***studentList*** .
  - Display info of the students in ***studentList***.

Then display the following menu:

1) Add: add a new **student :**

- Ask the user to enter the student's information and add the student to ***studentList***.

If the addition was successful, print a confirmation message, otherwise, print a message that indicates the unsuccessful addition operation. Printing should take place **in main**.

2) **Delete:** Delete a student from ***studentList***, knowing its ***id***.

If the deletion was successful, print a confirmation message that the deletion was successful, otherwise, print a message that indicates the unsuccessful deletion operation. Printing should take place **in main**.

3) **Display:** Display the information of all students in ***studentList.***

4) **Exit:** Write the students in ***studentList*** in a new file ***updatedStudents.data*** then exit the program.

The menu should continue to appear until the user wishes to exit the program.

Note:

You can have any number of variable declared globally in class **Application**

You should handle all possible exceptions that can be caused by the program **in main.**