ICS 102 Lab Project (Term 191)

Design a Student class that has the following properties:

- private instance variables:
    - private int studentID
    - private String name
    - private double[] grade
- An appropriate constructor
- The following public methods:
    - public int getStudentID( )
    - public String getName( )
    - public double[ ] getGrades( )
    - public void addGrade(double grade) throws IllegalArgumentException
    - public void modifyGrade(double grade, int quizNum) throws IllegalArgumentException
    - public String toString( )
    - public boolean equals(Object obj) - comparison based on studentID only

Your **StudentDriver** program must have the following main menu:

1. Display Grade Info for all students
2. Display Grade Info for a particular student
3. Display quiz averages for all students
4. Modify a particular quiz grade for a particular student
5. Add quiz grades for a particular quiz for all students
6. Add New Student
7. Delete Student
8. Exit

Please select your choice:

Implement a well-structured Java program that uses the **Student** class and **StudentDriver** class to enable an instructor to maintain student grades of up to 4 quizzes. The grade information is kept in a text-file of the form:

```
91007   Ahmad   Said          50.0
91004   Hassan  Khan          45.5
91003   Suleiman Wasim         72.6
91002   Majed  Sameer          60.0
91006   Muhammad  Adel         85.5
91005   Muhsim Zuheir          70.0
91001   Muneeb  Abdullatif     30.0
```

where each line of the text-file contains a **unique** student ID, the student first and second names, followed by up to four quiz grades (The above diagram shows the grades for the first quiz only)

When your program starts, it will read all the information in the input file. It will create an array of Students objects. Then it will display the menu shown above.

Your program must loop as long as **option 8** has not been selected. It must display an appropriate error message if an invalid choice is entered. After executing each of the options 1 to 7, your program must pause and display the message: **"Press Enter key to continue . . ."** . Your program must display the main menu after pressing the Enter key. Each of the options must be implemented in separate static methods. The code for **"Press Enter key to continue . . ."** must also be implemented is a separate static method.

The options must have the following behaviors:

Option 1: Display Grade Info for all students

It displays the grade information of all students. The option must be implemented by reading directly from grades text-file. It then waits for the Enter key to be pressed before returning control to the main menu:

```
Please select your choice:  1


            StudentID       Student Name            Quiz01

               91007      Ahmad    Said              50.0
               91004      Hassan  Khan               45.5
               91003      Suleiman Wasim             72.6
               91002      Majed  Sameer               60.0
               91006      Muhammad  Adel             85.5
               91005      Muhsim Zuheir              70.0
               91001      Muneeb  Abdullatif          30.0

        Press Enter key to continue . . .
```

Note: The number of quizzes displayed depends on the number of quizzes in the grades text-file. Initially, if the file has no quiz grades; it has only student IDs and their corresponding names.

Option 2: Display Grade Info for a particular student

It prompts for and reads the studentID. It then searches for this studentID in the array of students' objects. If the ID is not found an appropriate error message is displayed, otherwise; the student information is displayed. In both cases, the option waits for the Enter key to be pressed before returning control to the main menu.

```
Please select your choice:  2
Enter studentID: 91006

 StudentID            Student Name          Quiz01     Quiz02    Quiz03    Quiz04
     91006         Muhammad  Adel             85.5       90.0      95.0      98.0

Press Enter key to continue . . .
```

```
Please select your choice:  2
Enter accountID: 91552

Error: Invalid student ID
Press Enter key to continue . . .
```

Option 3: Display quiz averages for all students

The option displays quizzes average of all students. Control is then returned to the main menu after pressing the Enter key.

```
Please select your choice:  3

StudentID       Student Name           Average

    91007       Ahmad    Said            65.1
    91004       Hassan  Khan             56.3
    91003       Suleiman Wasim          80.7
    91002       Majed  Sameer            60.6
    91006       Muhammad  Adel           92.1
    91005       Muhsim Zuheir           83.3
    91001       Muneeb  Abdullatif      51.0

Press Enter key to continue . . .
```

Note: The average depends on the number of quizzes. Your program must work correctly for any number of quizzes taken. An error message must be displayed if students have not taken any quiz.

Option 4. Modify a particular quiz grade for a particular student

It prompts for and reads a studentID, the quiz number, and the new quiz grade. If the quiz number is invalid, or the new quiz grade is negative or greater than 100 an appropriate error message is displayed; otherwise it searches for this studentID in the students array. If the studentID is not found an appropriate error message is displayed, otherwise; the student grade is updated in the array of objects.

```
Please select your choice:  4
Please enter studentID: 91003
Please enter quiz number to modify:  2
Please enter new quiz 2 grade:  80.0

Before grade modification: 91003   Suleiman Wasim      72.6     75.0     85.0     90.0      80.7
After grade modification: 91003   Suleiman Wasim       72.6     80.0     85.0     90.0      80.7

Press Enter key to continue . . .
```

```
Please select your choice:  4
Please enter quiz number to modify:  6
Please enter new quiz 6 grade:  70.0

Error: Invalid quiz  number

Press Enter key to continue . . .
```

Control is returned to the main menu after pressing the Enter key.

Option 5: Add quiz grades for the next quiz for all students

It prompts the user to enter student grades for quiz **n + 1**, where **n** is the current number of quizzes for each student. The program then prompts for and reads the new quiz grade of each student and update the quiz array of each student object.

```
Please select your choice:  5
Please enter quiz grades for Quiz#3

Please enter grade for student : 91007
60.5
Please enter grade for student : 91004
70.0
Please enter grade for student : 91003
85.0
Please enter grade for student : 91002
55.5
Please enter grade for student : 91006
95.0
Please enter grade for student : 91005
90.5
Please enter grade for student : 91001
66.5

Press Enter key to continue . . .
```

Note: An error message must be displayed when any invalid grade is entered.

Option 6: Add New Student

It prompts for and read the ID of the student to be added. It will check if a student with same id already exists in the Students array. If not, it will be added by reading the remaining information i.e. name and quizzes and added as a student object to the array of students.
If the student with same id already exists, an error message will be displayed.

Option 7: Delete Student

To implement option 7, search the Students array for the studentID of the student to be deleted. If found, delete it from the array by moving all students objects after this student one location to the left. If the studentID does not exist, display an error;

Option 8: Exit

Save all data to the file, then terminate the program.

Note:

- Your project must not use parallel arrays. IT MUST USE an array of Student objects.
- Your project must not use 2D-arrays.
- Use the clause **throws FileNotFoundException** or **throws IOException** for each method that performs File I/O.
- **The project must your own work.**
- **You must not share code with any other project group.**
- **Teams of 3 students will be formed. Choose your partners from any of my lab sections and send me your ids.**
- **The deadline is Friday, 13/December/2019.**
- **Project demo slots of 15 to 20 minutes per group will be announced later. All group members are required to attend the project demo together. A grade of zero will be given for any student not attending the Project Demo.**
- **Cheating in any form will result in a grade of F in the course.**