MIAMI DADE COLLEGE

School of Engineering and Technology

COP2805C –Java Programming 2

Project 2

Due on 4/6/2017

Submissions are due before class. Late penalties start after 6:14 PM.

Design a class **Student** that contains the following members:

1- String fields **firstName, lastName** and **status**.
2- char field **letterGrade**.
3- double fields **grade1, grade2, grade3**
4- double field **average**
5- A two-argument constructor that takes **firstName** and **lastName** as parameters.
6- **computeAverage** method. Take into account that you will have different number of grades depending on the student.
7- **computeStatus** method (If **average** is < 70, the status will be "Failing". Otherwise, the status is "Passing")
8- **computeLetterGrade** method (If average >= 90 letterGrade is 'A', If average >= 80 letterGrade is 'B', If average >= 70 letterGrade is 'C', If average >= 60 letterGrade is 'D', else letterGrade is 'F')
9- **get** and **set** methods for **all** fields.

Design a class **StudentList** that contains the following members:

1- An **ArrayList** of Student objects **students**.
2- **readStudents** method that prompts the user for an input file name (use JFileChooser) and reads the contents of the input file into **students**. You can expect the file to be a text file with the following format:

```
firstName|lastName|Grade1|Grade2                 //Student 1
firstName|lastName|Grade1|Grade2|Grade3         //Student 2
firstName|lastName|Grade1                       //Student 3
firstName|lastName                              //Student 4
    …
```

Sample input file contents:

```
Michael|Corleone|78.6|99.7
Luca|Brasi|90.5|100|100
```

Phillip|Tattaglia|78.1
Kay|Adams

Notes:

- Use **String.Split()** method to parse the input.
- The file could have any length; therefore, you cannot make assumptions about how many students you'll find in the file.
- You can assume that the information for every student will include firsName, LastName and between 0 and 3 grades.
- Need to populate the **average**, **status** and **letterGrade** fields as appropriate.

3- **saveStudentsToDB** method that prompts the user for an DataBase file name (use JFileChooser) and writes the contents of **students** to the DB. The database will contain the table **StudentsTbl** with the following columns: **ID (ignore), FirstName, LastName, Grade1, Grade2, Grade3, Average, Status** and **LetterGrade.**

4- **findStudent** method that prompts the user for a student name and last name and shows a message indicating that the student was either found or not found in the DB.(use JOptionPane with a text field(s) and OK and Cancel buttons). Continue asking the user until the user presses **Cancel**.

5- **writeStudents** method that prompts the user for an output file name (use JFileChooser) and writes the contents of the **StudentsTbl** from the DB to the output file with the following format:

Name	Grade 1	Grade 2	Grade 3	Average	Letter Grade	Status
Michael Corleone	100.00	100.00	100.00	100.00	A	Passing
Phillip Tatagliaz	60.00	60.00	60.00	60.00	F	Failing

6- **writeSortedStudents** method that prompts the user for an output file name (use JFileChooser) and writes the contents of the **StudentsTbl** from the DB to the output file in ascending order of average (use **order by** SQL clause) with the following format:

Name	Grade 1	Grade 2	Grade 3	Average	Letter Grade	Status
Michael Corleaone	100.00	100.00	100.00	100.00	A	Passing
Julio Perez	60.00	60.00	60.00	60.00	F	Failing

Create a class **TestStudents** to test your work. This class will have a main that looks exactly like this:

```
public static void main(String[] args) {
      StudentList studentList = new StudentList();

      studentList.readStudents();
      studentList.saveStudentsToDB();
      studentList.writeStudents();
      studentList.writeSortedStudents();
      studentList.findStudent();
}
```

Place the following header on top of your files:

```
/*--------------------------------------------

 Student ID

 COP 2805C –Java Programming 2

 Spring - T Th 6:15 PM - 9:30PM

 Project # 2

Plagiarism Statement: I certify that this assignment is my own work and that I have not copied in part or
whole or otherwise plagiarized the work of other students and/or persons.

------------------------------------------------------------*/
```

Submission guidelines: Send your code files (**Student.java**, **StudentList.java** and **TestStudents.java**) and as attachments to my email **wmurill1@mdc.edu**, with the subject **Adv. Java Project 2**