Lab 5
100 Points
WMU GPA-O-Tron
Readings 12-13
Lab5classID

In our fifth lab of the semester we will create an application that enables the user to enter a student's WIN (no dashes), first name, last name, and semester GPA. If the student's GPA is 2.0 or above his/her information is written to a **sequential file named StudentsGoodStanding.txt** If the student's GPA is below 2.0 his/her information is written to a **sequential file named StudentsAcademicProbation.txt**

Once all information is entered, the program will display both lists **read from the files with appropriate headings.** The lists that are output **must be read from the files and not data in volatile memory.**

You can use either CLI (recommended) or JOptionPane for this lab. If you choose JOptionPane all output must be GUI.

Lab Parameters

We will use **Exercise 8A and 8B on page 725** of the textbook as our starting point. However, you are creating **one application and not two as discussed in the book.**

You should also add specific Exception handling to your application as you deem appropriate.

You should also add any **appropriate logic checks to control input and the flow of the program**. For example, You will need to determine when the user is done entering students so you can display the results.

For this lab, you will **have one class**. It should be named as such:

Lab5classID.java

This is your main application class. You determine what logic and exceptions should be included here.

At a minimum you need to do the following:

- Get all the students' information (WIN, first name, last name, and GPA)
- Write to the appropriate files (and close them): StudentsGoodStanding.txt and StudentsAcademicProbation.txt

- Read from the appropriate files (and close them): StudentsGoodStanding.txt and StudentsAcademicProbation.txt
- Display student information from each file in separate lists (**Probationary Standing** and **Good Standing**) with GPA context noted in 8B (exceeds or falls short of 2.0).

Make sure to add some design and include WMU's name, header information for the lists, and other important visial information in the output.

Challenge Parameters

For those who want to create a more robust application, consider the following in this order.

1. Sort by GPA in the display of each list.
2. Allow the user to choose to sort by GPA or Last Name in the display of each list.
3. Allow the user to re-run the application, **add** to the file entries, and re-display the list with chosen sort parameter.
4. Use a Student class to encapsulate critical data and logic from the main application
5. Allow the user to search for a GPA and display all student with that GPA and higher (or lower) in a single list.
6. Any other functionalities you deem appropriate to your program.

Important items to note

- **Use the following Path settings in your lab**
  - Paths.get("StudentsGoodStanding.txt");
  - Paths.get("StudentsAcademicProbation.txt");
- Use NetBeans for this lab and turn in the project folder
- Follow variable and file naming conventions
- Use appropriate data types
- Document every file that you create and/or change
- Include your classID as appropriate

Deliverables

This lab (entire project folder) should be turned in as a **.zip file** named with your Lab number and classID. For example, if your classID was **bjones4242** (Bubba Jones), the file would be:

Lab5bjones4242.zip

Refer to the **Documentation Guide** at for guidance on comments and lab preparation.

When you are finished, make sure to upload AND submit your lab in eLearning.