Programming Project 2

Jade Pearl

CMSC 215 6382 Intermediate Programming

Professor Romerl Elizes

**Approach**

As I moved through the project I made sure to:

1. Carefully read all materials in the instructions to ensure that I understand what it is asking
2. Come up with a plan/pseudo algorithm to help plan out what to write in each class. This plan includes the UML diagram which was really helpful
3. Code each class starting with Student and then onto the subclasses that use it: Undergraduate and Graduate. I then moved on to Project2 (the main method last)
4. Come up with test plans to make sure everything in the program is being used properly.
5. Polish off anything I notice could look better or be simplified.

**UML Diagram**

Below is the UML Diagram I created for this project. I used a flowchart maker so that the diagram is nice and neat compared to makeshift tables with drawn lines in Microsoft Word like I had for my first project:

A screenshot of a computer

Description automatically generated

**Test Plans**

Below are each of the 5 test cases I carried out to make sure that the project worked properly:

**Test 1: Sample input given from Project 2 Instructions**

Input**:**

**A screenshot of a computer

Description automatically generated**

Result: Normal result expected. The only one eligible is the student pursuing the Masters degree because their GPA was higher than the threshold GPA.

A computer screen with blue text

Description automatically generated

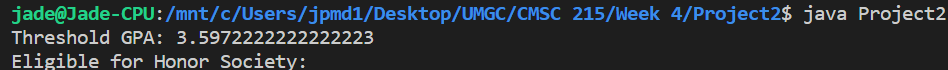
**Test 2: One Student**

Input:

A screenshot of a computer

Description automatically generated

Result: No student listed. The threshold is higher than the average gpa (which in this case is that one student’s gpa) because it is meant to be the midpoint of the average gpa and the 4.0 which will always be higher than the average itself. So if one student participated, there would be no honors!



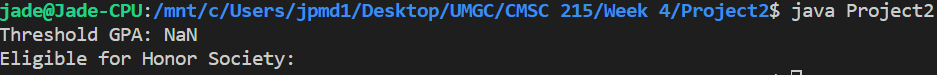
**Test 3: No Students/Blank input file**

Input:

A screenshot of a computer

Description automatically generated

Result: NaN gpa since totalGPA has a value of 0 and there is no data to read in



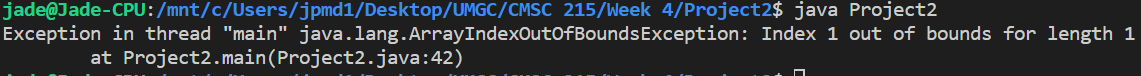
**Test 4: Syntactically incorrect file**

Input:

A screenshot of a computer

Description automatically generated

Result: Error due to incorrect formatting which is expected since in the main method, I split each line with a “ “ as a delimitor. If there are no spaces, there are no “pieces” of the string that can be assigned.



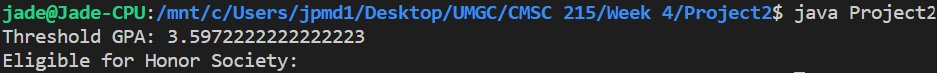
**Test 5: Many duplicates of the same student**

Input:

A screenshot of a computer

Description automatically generated

Result: Treated the same as one student. All of the duplicates have the same GPA and the threshold will seemingly always be higher than the average gpa, so none of the clones of the student will be eligible for honors either. This shows that calculations of the GPA are done correctly.



**Lessons Learned**

There were a lot of obstacles doing this project. The biggest obstacle was mainly trying to get behind the logic of how to put the main method together and what I needed to do to read each student in from the text files. Overall, in this project I learned:

* More about how to use subclasses and how they work when you make objects stemming from a subclass of a class.
* More interaction with String functions, especially split which is what I used to pick out each piece of data from the text files.
* How to ask myself the right questions when I am troubleshooting (developed better debugging skills)
* Better practice with throwing and catching for exceptions. At first I could not properly use the statement where I made a new Scanner object based on a File object because it wanted me to use an exception block.
* Improvement of previous skills and understanding of classes and their relationships and how to chart them in a UML diagram