Lab 1-b writeup

I. Team members: Lucas Greenelsh (greenels)

II. Initial decisions:

   A. I determined Java to be the programming language due to my level of familiarity with the language.

   B. I wrote the program using Intellij IDE and Vim text editor

III. I chose to create a Student class that would represent individual entries. I chose to build a simple arraylist of Students that could be easily appended to and iterated over. For the implementation of part B, I chose to modify my existing program to read two different files and fill two different arraylist structures with instances of two different objects (Student and Teacher).

IV. Task log:

   A. Initial design: Lucas Greenelsh, 9/16/2020 3:00pm - 4:00pm, 60 minutes.

   B. Student implementation: Lucas Greenelsh, 9/16/2020 4:00pm - 4:10pm, 10 minutes.

   C. Schoolsearch implementation: Lucas Greenelsh, 9/16/2020 5:00pm - 6:00pm, 60 minutes.

   D. Testing: Lucas Greenelsh, 9/16/2020 4:00pm - 6:00pm, 120 minutes.

   E. Test script: Lucas Greenelsh, 9/19/2020 12:00pm - 1:00pm, 60 minutes.

   F. Writeup: Lucas Greenelsh, 9/20/2020 6:00pm - 6:30pm, 30 minutes.

   G. Part B design: Lucas Greenelsh, 9/23/2020 3:30pm - 4:00pm, 30 minutes.

   H. Teacher implementation: Lucas Greenelsh, 9/16/2020 7:30am - 7:45am, 15 minutes.

   I. Modifying schoolsearch file to read from two different files and while maintaining original functionality: Lucas Greenelsh, 9/24/2020 - 8:00am - 10:00am, 120 minutes.

   J. Adding functionality for requirements NR1 - NR5: Lucas Greenelsh, 9/24/2020 12:00pm - 2:00pm, 120 minutes.

   K. Modification of test script for additional testing: Lucas Greenelsh, 9/24/2020 3:00pm - 4:00pm.

V. Notes on testing: The majority of my testing was done in concert with the coding of the program. This is why the time logs for the two processes have an overlap. My tests yielded approximately 12 bugs and the majority were uncaught number format exceptions which were easily corrected. Testing for part B yielded few bugs, all of which were from duplicate data being reported. These were addressed by using a handful of boolean return values in conditional statements to control the flow of the program.

VI. Design modifications for part B: In order to make my existing code perform the same look-ups on two seperate data sources, I needed to add an additional class (Teacher) in which I could store the teacher data. In addition to this new Teacher class, I also modified my Student class to have fewer data fields as the same data as before would now just be divided between two relations. I also had to modify some of my methods to take an additional arraylist structure as a parameter and iterate over that arraylist.

VII. I implemented the new search commands by mostly augmenting the existing options. For example, in the menu options:

`"T[eacher]: <lastname>\n"` became `"T[eacher]: <lastname> [G[PA]]\n"`

The change to this menu option resulted in the average GPA of the specified teacher’s students being reported when the user included the optional G flag.
I only had to create completely new menu option:

```
"C[lass]: <number> T[eacher]|S[tudent]\n"
```

Selecting this menu option with the specified mandatory input would result in either a list of students assigned to the specified room number or the name of a teacher assigned to the specified room number being reported.