Griffin Aswegan, Steven Bradley
CSC 365 - Professor Emilia Buckalew
September 29th, 2018
Lab 1 Part 2 Writeup

Initial Decisions

As per the lab spec, we used Java as our chosen language. We developed in two separate environments, depending on the person - Griffin used Eclipse, and Steven used IntelliJ. We also decided to use Git/Github as a source control, both for safety as well as easy of access for both team members.

Program Information/Notes

We decided the best course of action would to be reading in the entire file, constructing a linked list of the information, and then using that list to search and parse through information. We decided a linked list would be best, as in order to search you need to tab through every element in order to find the correct information, negating the speed benefits of using an array. We created a class to contain every bit of information, since all students will have the same information as well as that information is consistent and always in the same order. Combining the two, we have a simple Linked List of Student class objects, as well as a simple Linked List of Teacher class objects.

We decided to keep our program from part 1 and just adjust it to fit the new code. The biggest changes were adding a new Teacher class, similar to the Student class, and reading that in from a separate file. We had to adjust some commands to understand that the teacher's name wasn't in the Students file anymore, but it did not take long. Mostly, the biggest pointes affected were initial setup (opening the teachers file and the list file instead of just one, plus the addition of the teacher class) as well as some of the internal workings of the commands (most notably, the "Teacher" command, and any command that printed out teacher's names.

There are also a couple of new commands: Grade has been modified - to use the command as specified in Part 1 R7, the S[tudent] flag can be used, while the T[eacher] flag adheres to the new NR4 requirement. One or the other of these commands must be specified in order to use the program. The second command "P", or "GPA", grabs all of the GPAs (anonymized) and sorted based on one of three values: the Grade the student is in, the Teacher the student has, or the Bus route that the student takes. Finally, the last command, "Enrollment" just displays the number of students in each classroom.

Some examples (and remember the colon - it's important):

"GPA: Bus" will display all GPAs sorted by bus (see [1] for example output).

"p: t" will display all GPAs sorted by teacher (see [2] for example ouput).

"Grade: 4 Student" will display the grade as it was before [3].

"G: 4 teacher" will display the teachers teaching grade 4 [4].

As a small aside, all commands and arguments for the program are case insensitive, meaning "Enrollment", "enrollment", "E", and "e" will all execute the enrollment command mentioned above.

Finally, for the GPA command, we decided to only display all of the GPAs individually - we believe that averaging the scores together displays less information, while also performing data analysis, which is something the program was explicitly supposed to not perform.

Task Log

Refactor of Current Code to adapt for a separate Teacher File:

Steven Bradley - 1 Hour (Wednesday, 3:10 PM to 4:10 PM)

Addition of Requirements NR1, NR2, and NR4

Griffin Aswegan - 1 Hour (Wednesday, 3:10 PM to 4:10 PM)

Addition of Requirement NR3, NR5

Steven Bradley - 0.5 Hours (Thursday, 2:30 PM to 3:00 PM)

Testing, Refactoring, Merging, and Adding Tracability

Griffin Aswegan & Steven Bradley - 1 Hour (Wednesday, 3:10 PM to 4:10 PM)

Testing Notes

See the readme for information on how to run our test. We have one file, labeled “tests.txt” which tests each command’s syntax and checks output for correctness.

Mostly our issue with testing was merge conflicts. Getting every little detail sorted out took a bit of time, but it got done pretty quickly. There were some exceptions thrown, mostly revolving around not checking for tokens properly inside of a command. Other than that, most everything worked as intended.

Final Other Notes

For clearer output, if any additional command line arguments are specified when being run (for example, running “java schoolsearch asdf”), the program will run in “test” mode, which removes the output that lists the commands and what they do. Running it with no arguments displays the list. This was added to help with clarity while testing.