CSC 365

Lab 1 Writeup

Team Members: Melanie Huang, Bella Boulais

Initial Decisions:

- Programming language: Python
- Development Environment: VS Code

Internal Architecture:

We decided to use a dictionary to store the student information. We thought this was the best data structure to use because we could easily get all the data types from a student by sending in the key of the student's last name. When we want a specific element from the student information, we could ask for the index of teacher, or the student gpa. We passed in the student file and checked that the file was not empty. For each function, we checked that the student last name that was passed in was in our file. If it was not, then an empty string would be printed.

For each search option, we made the key of the dictionary whatever the search was. For example, for "Student: <last_name>", the key was the last name and the values were a list of the rest of the students information. If there were two students with the same last name, the value would make a list of lists. For "Bus: <number>" the key was the bus number and the value was a list of lists of students with that bus number. For "Info" we made the key the grade and the value a list of lists of all the students within that grade, etc.. This made lookup for each search option simple and fast. We also realized we only needed to store the student information that needed to be printed, so that made returning the student information simpler, since we could just return the whole value.

Task Log: (Requirement, Team member, time taken)

R2 & R3 Bella and Melanie 2 days

- Completed as each search option was completed

R4 Melanie 10 min
R5 Melanie 20 min
R6 Melanie 10 min
R7 Melanie 10 min
R8 Melanie 10 min
R9 Bella 30 hr
R10 Bella 10 min
R11 Bella 10 min
R12 Bella 10 min
R13 Melanie 30 min

E1 Bella 30 min

Testing (suit & output): Bella & Melanie

- Wrote test cases and recorded outputs for the requirements we worked on as noted above
- Time taken: 1.5 hours (including writing, performing, and fixing bugs)

Testing:

Both of us tested the code after completing the program requirements R2 - R12:
Time taken: 2 hours (including writing, performing, and fixing bugs)
Bugs found: ~ 1 hr to fix

1. When implementing Grade: <number> High, if High was spelled wrong it would choose the low option automatically
    a. Made sure if High or Low was spelled wrong it returned to the command line prompt
2. Wasn't casting GPA's to floats if imputed as integer (i.e. 3 instead of 3.0)
    a. Casted every gpa to float when reading in lines to ensure it will show a float if printed
3. Variables in `if __name__ == "__main__":` conflicting with variables in main() (i.e. grade & grade)
    a. Changed variables in `if __name__ == "__main__":` so that they don't conflict and raise errors with main variables

Cases tested:

- Students with same last name in all the instances
- Student name, bus, grade, classroom, teacher name, gpa doesn't exist
- Multiple Students with same high/low GPA
- Check empty list
- File name not found
- Format of line is incorrect