Miriam Lam
CSC 365 - Introduction to Databases
Lab 1, Part a

• List of team members

Miriam Lam

• Initial decisions: programming language, development environment

Python because it is the easiest and I am familiar with it. Python also has nice built in data structures that I am not sure how to use in other languages. (Set, Dict)

• Notes on your chosen internal architecture: what data structures you used, for what purposes

I used a 2D array because I thought of the data as a table, and the lectures implied that the tuple was a row, and the attribute was the column. And there would be no indexing errors because each category was filled for each person (no missing GPA, etc).

• Task log. For each task to be completed, list the name of the task, the student(s) performing it, start time, end time, total person-hours it took to complete. Choose the granularity wisely. You do not have to document every method or function.

- Setting up IDE
  - 10 minutes
- Setting up the switch statement/UI
  - 20 minutes
- Converting students.txt to 2D array
  - 15 minutes
- First function (student last name)
  - 20 minutes
- Each successive function aside from maxGPA
  - 10 minutes
- Debugging
  - 30 minutes

• Notes on testing. When, who, how long, how many bugs found, how long it took to fix them

In my “for student in students”, I mixed up student and students, which was easily fixed with a print statement.
For the ones with flags, I realized I had to check if there was a flag before checking what the flag was. And I had to check if there was no flag in the switch statement as well.

• Final notes (anything else you want to share with me about your implementation)

I don’t feel very confident that this is a good way to look up data, considering that it is O(n) for all of these functions since I look through every student. I’m sure real databases have better complexity. Would it have made a difference if I put the data into a dict() because dict would

require key,value? I think I did the same thing in an array and treating the index as a key. 0=studentlastname always. I also assumed that the requirement E1 just meant that **any** errors in the program would just be ignored and just asks for a new command