Lab 1-1
Cole Grigsby
Avinash Sharma
4/5/17

Initial Decisions:

We decided to use Python 2 to implement this lab. This way, we can easily write methods for each functionality and run it easily from a unix command line without compiling.

Internal Architecture:

For our Student object, we created a field representing each column of the text file to easily clean and retrieve the data being search for. We read in the students.txt file once using the readline function and split the data by commas into tokens and stored each token in their corresponding field within our Student object. When reading the file in, we created all of the Student objects and stored them in an array local to the runProgram function.

Tasks:

- Read in file and parse into array of student objects (initial set up) - Avinash Sharma
    - Started and finished April 3rd in lab (30 min)
- Each interactive command - Cole Grigsby, Avinash Sharma
    - Met on Tuesday in library to work on it (2.5 hours)
- Testing - Cole Grigsby, Avinash Sharma
    - While writing commands we wrote the test cases for them (1 hour during command writing, 15 mins to run all and store tests.out)
- Writeup - Cole Grigsby, Avinash Sharma
    - Started April 3rd in lab (10 min)
    - Finished April 4th in library (25 min)

Testing:

We wrote test cases as we designed methods for each command to test their functionality. After completing the commands, we tested everything together. We found a few small formatting errors that we were able to fix quickly such as, parsing our input using split by whitespace instead of split by a single space. We didn't have too many bugs, but we had an issue with passing parameters within a function in our gradeTop and gradeLow functions that we spent about 10 minutes to fix.

Final Notes:

We wrote a function to get the results for each command from the user input and printed out the results of those functions within our runProgram function.