Lab 1: SchoolSearch

Team members: Sara Bilich and Makenna Johnstone

Initial Decisions:

We chose to use Java to develop the program because it is a high-level language that would make the development process faster and easier to organize. We developed the program in Atom, a simple, lightweight IDE.

Internal Architecture:

We created a class called *Student* to encapsulate all of the information pertaining to a particular student. Within the main method of the SchoolSearch program the contents of students.txt is kept as an ArrayList of Student objects. We chose to create a new class to hold the information about a student because it nicely organized the data into discrete blocks with a predictable structure. An ArrayList was chosen to hold all of the Student objects because it is easily traversed and are relatively lightweight.

Task Log:

Created git repository - Sara Bilich - 1/8/18 - 5 minutes
Wrote Student.java - Sara Bilich - 1/8/18 - 15 minutes
Created stubs for all necessary files - Sara Bilich - 1/8/18 - 10 minutes
Wrote parsing logic in SchoolSearch.java - Sara Bilich - 1/8/18 - 40 minutes
Created write-up doc and wrote "Initial Decisions" & "Internal Architecture" - Sara Bilich - 1/8/18 - 15 minutes
Diagnosed and fixed Java versioning issue - Makenna Johnstone - 1/10/18 - 35 mins
Wrote switch statement, wrote search options prompt and parsed input from user - Makenna Johnstone - 1/10/18 - 1 hr
Implemented the first half of the search functions - 1/11/18 - 1.5 hr
Implemented the second half of the search functions - Makenna Johnstone - 1/11/18 - 3 hrs
Tested code with examples through the command line - Makenna Johnstone - 1/11/18 1 hr
Fixed output formatting and added cases for error handling - Sara Bilich - 1/15/18 - 2 hrs
Wrote and tested test cases for tests.txt and tests.out - Sara Bilich - 1/15/18 - 1.5 hrs

Testing Log:

1/15/18 - Sara Bilich - 2 hrs to find and fix - at least 10 different bugs, all of which were NullPointerExceptions or ArrayIndexOutOfBoundsExceptions

Final Notes:

Our implementation of lab 1 has a simple internal structure and tries to focus on providing useful and meaningful responses to invalid input. We aimed to pinpoint the faults in a query so that the user can safely recover from their mistake and make a successful query the next time around.