Ryan Hartzell
Anthony Delgado

CSC 365 Lab 2 Report

## Initial Decisions

We decided to use Java as our language for this assignment, as it was the language we felt most comfortable using.

The environments used were IntelliJ and Vim.

## Notes on Internal Architecture

We created a Student class to hold information for each Student (Last name, First name, Grade, GPA, Bus, Teacher last name, and teacher first name).

We used a HashSet as our structure to hold objects of type Student. This was for efficiency.

## Task Log

| Task Name | Student Performing | Start Time (M/DD, H:m) | End Time | Total Hours Spent on Task |
|---|---|---|---|---|
| Build Data Structures | Ryan Hartzell | 9/19, 12:00pm | 9/19, 12:30pm | 0.5 |
| Implement File I/O | Ryan Hartzell | 9/19, 12:30pm | 9/19, 1:00pm | 0.5 |
| Implement Search Options | Ryan Hartzell | 9/19, 8:00pm | 9/20, 9:30pm | 1.5 |
| Test Suite | Anthony Delgado | 9/21, 12:30pm | 9/21, 1:30pm | 1 |
| Report Write-up | Anthony Delgado | 9/19, 12:00pm | 9/21, 12:30pm | 0.5 |
| Modification to Retain Functionality | Ryan Hartzell | 9/25, 8:30pm | 9/25, 10:00pm | 1.5 |
| Implementing New Search Options | Anthony Delgado | 9/25, 11:00pm | 9/26, 12:00am | 1 |
| Implement Analytics | Anthony Delgado | 9/26, 11:30am | 9/26, 1:30pm | 2 |
| Amend Report | Ryan Hartzell | 9/26, 12:30pm | 9/26, 1:30pm | 0.5 |

## Notes on Testing

Informal Testing (Trying to Compile/Run commands): Ryan Hartzell found 2 bugs during the implementation of search options; it took approximately 10 minutes to fix them.

## Code Modification

We modified our code by adding a Teacher class so that both text files would be read in and stored as a set of objects. Additionally, we then had to add another file parsing method and slightly edit the main interactive loop functions to accommodate having to draw data from two sets.

## Additions to Query Language

C[lassroom]: <number>

Now outputs both students and teachers for a given classroom.

G[rade]: <number>

Now outputs both students and teacher for a given grade.

E[nrollment]

Outputs enrollment broken down by classroom.

A[verage]: <number> | B[us] | G[rade] | T[eacher]

Inputting any of the three text options (B, G, T) will output average GPAs broken down by each unique instance of that option. i.e. “A: G” will output the average GPA in each grade