Victor Delaplaine
Jaipreet Hundal
Kyle Detwiler

Writeup1-2

Initial Decision:
-The group continued to use our Java implementation that was previously used in part1. This allowed us to keep our core architecture that had success and join that with new functions created to bring in the teachers data.

Architecture:
-The key changes that were made was the implementation of the Student class and the hashmap for teacher names. The map was built on the key-value pair of (classroom, teacher's name), allowing us to pair any student with the correct teacher. This was possible because each teacher had a unique classroom. The creation of the Student class allowed for more efficient data storage, and quicker access of values by pulling from an object instead of indexing a list.

Task Log:
-**Victor:** Tasked with creating the parsing tools to read in and read out the text file. Each line in the file would be stored into a list to create easy access and comparison when filtered through the streams.
Time: 5pm - 6:30pm
Total Hours: 1.5
-**Jai:** Tasked with creating testing scripts and Student class
Time: 6pm - 7pm
Total Hours: 1
-**Kyle**: Tasked with creation of the write-up as well as the pairing operation (construction of hashmap)
Time: 5pm - 6pm, 9am-10am
Total Hours: 2

Testing:
-Jai was responsible for testing and encounters 2 bugs. One bug came from naming convention since the work and code development was split amongst three people. In addition to that, the creation of the hashmap needed to be tweaked due to hashcode incorrectly storing classrooms.

Part 1 Changes:
-Very few modifications were made to the previous code. Aside from the added code discussed above, our “grade” command and our “c” command were changed to accommodate the correct output. Our query in light of the implementation of the hashmap pairing architecture created an extra step, using the classroom of the student as the key to pull the teacher information.

Examples:

```
HashSet<Pair<String, String>> pairs = new HashSet<>();
Map<Integer, Set<Pair<String, String>>> gradeToTeachers = new HashMap<>();
Map<Integer, Integer> classToTotStudents = new HashMap<>();
```