Thoughts:
I'm pretty happy with how this project went. It can take a wide range of group sizes, and can take any number of students. The hill climbing algorithm is effective and fast, though there are a few improvements that could be made. The main problem with this program is that it gets stuck at local optima very easily due to the very bare bones rules for accepting or denying a swap.
This could be fixed by instead making changes based on probability, and increasing or decreasing the chance that a change is accepted if it meets certain conditions. For example, it could discourage pairing students that dislike one another, or it could encourage pairing students who share similar dislikes for a certain student. Ultimately, the program is still more than good enough for the needs of a teacher.

Resources Used:
https://www.w3schools.com/java
Used for general java help.

https://www.tutorialspoint.com/how-to-read-the-data-from-a-csv-file-in-java
Used to learn how to extract data from a csv file

https://www.geeksforgeeks.org/generics-in-java/
Used to assist in using generics.

https://en.wikipedia.org/wiki/Hill_climbing
Used to research Hill climbing.

chatgpt was used exclusively for help in determining the algorithm I would use to organize students optimally.
Link to the chatgpt conversation: https://chatgpt.com/share/670c27e1-7da4-800b-b6f5-79fa9808b56c

Scalability:
My code would scale linearly with the number of students, as all parts of the algorithm that change with respect to the number of students are O(n) complexity. The time to fully extract the csv file data is linearly proportional with the number of lines in the csv file, which itself is linearly proportional with the number of students. Additionally, the number of random swaps done during the hill climbing algorithm is just the number of students multiplied by 1000, so it as well scales linearly.