**Abstract:**

This project will allow two users to play a game of checkers against each other. There will be an eight by eight board, with each person starting with twelve checkers. The users’ checkers will be placed on the same diagonal path, allowing them to jump over, and therefore remove, the other user’s checker. A checker can only move in a diagonal manner, and can only move one space, unless an opponent checker is in the space, then it can jump over the checker. Multiple jumps are allowed if the checker lands in a space where another jump is available. King checkers are made when a user moves their checker to the farthest row from where they started. A king checker can move both directions, while a normal checker can only move away from where each user started individually. The first person who jumps all of the other users checkers will be the winner.

**Team Members:**

Phillip Neidlinger, Noah Schuler, Zackery Kim, Ashley Simpson

**Location of Repository:**

https://github.com/Ashley00simpson/CSCI-325-checkers

**Program Description:**

The checker class will set up an object that has two color options, a size, and a direction in which the checker can travel. The king class will allow the checker to move both directions along with changing the checker’s appearance. The board class will set up a two dimensional array that will house twenty-four checkers in the correct places. The main function will interact with the users to allow the users to make moves. The main function will output the board, ask for alternating input from the users, determine if the input is valid, and update and re-display the board again. Also, when all the checkers are gone for any user, the game will end and a user will be displayed.

**UI Description:**

The users will be playing the game of checkers in the command line in IntelliJ.

**Division of Labor Description:**

The main ways the labor will be divided up will be into four parts of creating a checker class, creating a king class, creating a board class, and creating a main function.