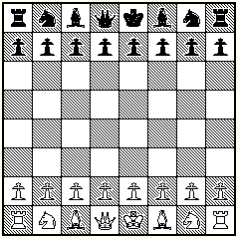
Chess Coding Exercise

Pretend that you taking over a software project that has only just started and may be on the wrong track. The project’s long-term goals are to build a fully functional chess game and you have been hired to pick up where others have left off. You won’t be writing it all at once and the first sprint will have a narrow focus on some basic functionality of the Chessboard and some simple movements of a Pawn.

All simulations take place on a chess board (class name: ChessBoard) that is a grid consisting of length X, and height Y – both of which are integers. Chess pieces can be placed on the board at a given (x,y) coordinate pair with (0, 0) being in the lower left-hand corner of the board as seen in the following illustration:

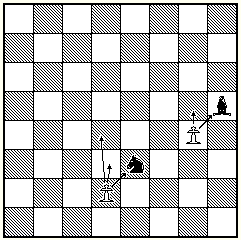
(7, 7)



(0, 0)

Pieces are either Black or White. Black pieces typically start at row x=7 and x=6, whereas white pieces typically start at rows x=0 and x=1. That said, you can set up a board with many initial configurations to replay famous chess games (that last bit might be a paradox).

Additionally, Pieces can be given two commands: move and capture (we will ignore capture for this exercise). Each piece has unique movements, but we are going to focus on commands for pawns. For our limited implementation, Pawns can only more forward one space (toward their opponents side of the board) and can only capture in a forward and diagonal direction as seen in the next illustration.



Your task is to get all unit tests found under the Tests folder passing. Since you plan to be on the project long term, think about how you would implement the solution, what other test coverage might be necessary and what you would do to make future features easier to implement.

Good luck, and please reach out to us if you have any questions!