For this project you will write a class called Othello that allows two people to play text-based Othello

(https://en.wikipedia.org/wiki/Reversi). The Othello game is a strategy board game. In this game, two players take turns placing their colored pieces on an 8x8 board. The objective is to capture the opponent's pieces and have most of your own pieces on the board at the end of the game.

\*\*Rules: \*\*

\* The game is played on an 8x8 board.

\* Players take turns placing their pieces on the board, starting with black.

\* To capture pieces, a player must place their piece adjacent to an opponent's piece, forming a straight line of adjacent pieces (horizontal, vertical, or diagonal) with their piece at each end.

\* Multiple chains/directions of pieces can be captured all at once in a single move, and the captured pieces are converted to the capturing player's color.

\* The game starts with four pieces placed in the middle of the board, forming a square with same-colored pieces on a diagonal. Once a piece is placed, it cannot be moved to a new square.

\* If a player cannot make a valid move (a capturing move), their turn passes to the other player.

\* The game ends when neither player can move, and the player with the most pieces on the board wins. A tie occurs if both players have the same number of pieces.

For a better understanding of the rules, you can play the game at this site: https://www.eothello.com/

\*\*Game Board: \*\*

The game board is represented by a 10x10 grid as figure 1 shown below.

\* Edge: \* (star)

\* Black piece: X

\* White piece: O

\* Empty space: “. “ (dot)

Each position on the board could be represented by a (row, column) pair. For example, at the beginning, white pieces

are at position (4,4) and (5,5) and black pieces are at (4,5) and (5,4).

Your code for the game must define the class and methods described below, but you are encouraged to define other methods or classes that may be useful for the game. All data members must be \*\*private\*\*

\*\*Player: \*\*

The Player class represents a player in the game. It contains the following information:

\* Player name (string)

\* Piece color (string): Either "black" or "white"

\*\*Othello: \*\*

The Othello object represents the game as played. It contains information about the players and the board. Name the

board as "board", so we could access each position value on the board by \*\*self.\_board[row][column]\*\*. It must include those methods (but may have more):

\* print\_board(self): print out the current board, including the boundaries

\* create\_player(self, player\_name, color): creates a player object with the given name and color ("black" or "white") and adds it to the player list

\* return\_winner(self): returns "Winner is white player: player’s name" when white player wins the game, and returns “Winner is black player: player’s name" when black player wins the game, and returns "It's a tie" if black and white player has the same number of pieces on the board when the game ends.

\* return\_available\_positions(self, color): returns a list of possible positions for the player with the given color to move on the current board.

\* make\_move(self, color, piece\_position): puts a piece of the specified color at the given position and updates the board accordingly, then return the current board(as a 2d list). make\_move is an internal method and is meant to be called by play\_game, but for testing purposes it should be able to be used alone. You could assume that we will only pass valid position to this method.

\* play\_game(self, player\_color, piece\_position): attempts to make a move for the player with the given color at the specified position. If the position the player wants to move is invalid, the function should not make any move and

\* return "Invalid move", and print out this message "Here are the valid moves:" followed by a list of possible positions. If no valid moves exist, then the returned list is empty. If the position is valid, the function should make that move and update the board. If the game is ended at that point, the function should print "Game is ended white piece: number black piece: number" and call the return\_winner method.

Your python file must be named \*\*Othello.py\*\*

For the testing purpose, the above methods should be able to be called alone, which means we could use the methods like this as an example: