

Your assignment is to create a connect four game. The game consists of two people who alternately drop their unique game piece down a column of the gameboard. Once a player has 4 pieces in a row either horizontally, vertically or diagonally, then they win the game.

You will need to use the following interfaces for your implementation.

ConnectFourPlayerInterface and ConnectFourGameInterface

This is important because I am also contributing to your solution and have written the class

ConnectFourDriver which will create an instance of your class and run the game. Of course, I wouldn’t know how to do that (which methods to call) if you weren’t using these interfaces.

You will notice that the constructor for a player can accept either a string that will refer to the name of the player or no parameters meaning the player name should default to “Group <your group number>”.

You will need to implement the functionality of the game and create a player class to do so. Your game should play indefinitely until a player selects not to play again.

An example execution of the program may look like this,

>$ java ConnectFourDriver

Bob select a single char game piece

a

Player select a single char game piece

b

Welcome to Connect Four!

.......

.......

.......

.......

.......

.......

Player 1: Bob select a column

0

.......

.......

.......

.......

.......

a......

Player 2: Player select a column

1

.......

.......

.......

.......

.......

ab.....

Player 1: Bob select a column

0

.......

.......

.......

.......

a......

ab.....

Player 2: Player select a column

1

.......

.......

.......

.......

ab.....

ab.....

Player 1: Bob select a column

0

.......

.......

.......

a......

ab.....

ab.....

Player 2: Player select a column

1

.......

.......

.......

ab.....

ab.....

ab.....

Player 1: Bob select a column

0

.......

.......

a......

ab.....

ab.....

ab.....

Bob has won the game

Would you like to play again? y/n

n

Bob (Player 1) has 1 wins and Player (Player 2) has 0 wins.

>$

**Grading:**   
Correctness: You can lose up to 20% if your solution is not correct   
Quality: You can lose up to 20% if your solution is poorly designed   
Testing: You can lose up to 20% if your solution is not well tested   
Explanation: You can lose up to 40% if you cannot explain your solution during the grading session