A Tic-Tac-Toe Class

```
Author: Graham Mitchell
Filename: TicTacToeClass.java
```

Write a class to handle the basics of a two-player game of Tic-Tac-Toe. The required methods are below. If you've done things correctly, save a copy of the files provided below into the same folder as `TicTacToeClass.java` and compile and run them to play or test your game.

Files Needed

- TicTacToeClass.java - starter code
- TicTacToeRunner.java - a runner to play TicTacToe. Or, write your own for extra credit.
- TicTacToeTester.java - a brutal tester to make sure your isWinner() method is *always* correct
- TicTacToeGUI.java - a different runner

The only tricky part about the game is determining if a given person has won. The more straightforward way to do it takes 8 **if** statements.

Instance Variables	
`board`	a two-dimensional array of chars
`turns`	an integer keeping track of the number of turns played this game
Constructors	
`TicTacToeClass()`	the default constructor, which just creates the two-dimensional array and fills each slot with `' '` (a blank space) and initializes the other attributes
Accessors	
`boolean isWinner(char p)`	returns `true` if the letter passed in currently has three in a row. That is, `isWinner('X')` will return true if X has won, and `isWinner('O')` will return true if O has won
`boolean isFull()`	returns `true` if nine turns have been played and `false` otherwise
`boolean isCat()`	returns `true` if all nine spaces are filled AND neither X nor O has won
`boolean isValid( int r, int c )`	returns `true` if the given row and column corresponds to a valid space on the board
`int numTurns()`	returns the numbers of turns played so far
`char playerAt( int r, int c )`	returns the character representing the piece at the given location. Should return either `' '`, `'X'`, or `'O'`.
`void displayBoard()`	displays the current board on the screen
Modifiers	
`void playMove(char p, int r, int c)`	allows the given player to place their move at the given row and column. The row and column numbers are 0-based, so valid numbers are 0, 1, or 2

```
(...a game already in progress)

       X   0
       0 X X
```

```
          X O

'O', choose your location (row, column): 0 1

        X O O
        O X X
          X O

'X', choose your location (row, column): 2 0

        X O O
        O X X
        X X O

The game is a tie.
```