Name: _____

Swift: Basic Classes

Part 1: Class Definitions

Below are a few definitions of classes. Below, fill in the tables which list the members of each class.

```
class ChessMove {
	let piece: ChessPiece
	let toX: Int
	let toY: Int
	let color: Bool
	init(piece: ChessPiece, toX: Int, toY: Int, color: Bool) {
		self.piece = piece
		self.toX = toX
		self.toY = toY
		self.color = color
	}
}
class ChessPiece {
	let name: String // "King", "Queen", "Knight", etc.
	let color: Bool // false == white, true == black
	var x: Int
	var y: Int
	init(name: String, color: Bool, x: Int, y: Int) {
		self.name = name
		self.color = color
		self.x = x
		self.y = y
	}

	func possibleMoves() -> [ChessMove] {
		// Code to compute moves omitted.
		return ...
	}
}

class ChessBoard {
	var pieces: [ChessPiece] = []
	var isWhiteTurn = true
```

```
	init() {
		setupBoard()
	}
	func setupBoard() {
		pieces = [ ... ] // List of initial pieces omitted.
	}
	func isLegal(move: ChessMove) -> Bool {
		// Code to compute complicated chess rules omitted.
		return ...
	}

	func perform(move: ChessMove) {
		if isLegal(move: move) {
			// Do things here to move the piece.
		}
	}
}
```

ChessMove

Member Name	Type
piece	ChessPiece
toX	Int
toY	Int
color	Bool

ChessPiece

Member Name	Type
name	String
color	Bool
x	Int
y	Int
possibleMoves	() -> [ChessMove]

ChessBoard

Member Name	Type
pieces	[ChessPiece]
isWhiteTurn	Bool
setupBoard	() -> ()
isLegal	(ChessMove) -> Bool
perform	(ChessMove) -> ()

Part 2: Class Structure Proposal

For the above problem, I sort of gave a rough proposal for how one could use classes to structure the code for a chess game. In this problem, I leave it up to you to propose a class or multiple classes to structure the code for a tic-tac-toe game. This an open-ended problem (just like real programming), so there isn't a single right answer, but there are definitely good and bad answers. For every class you need to list out the instance variables and functions, including parameters and return types. However, you do **not** need to actually implement the methods (similar to what I did above).

```
// This is only one of infinite possible right answers…
class TicTacToeBoard {
      var isXTurn = true

      var topLeft = ""// "" for empty, "x", for x, "o" for o.
      var topMiddle = "" // "" for empty, "x", for x, "o" for o.
      var topRight = "" // "" for empty, "x", for x, "o" for o.

      var middleLeft = "" // "" for empty, "x", for x, "o" for o.
      var middleMiddle = "" // "" for empty, "x", for x, "o" for o.
      var middleRight = "" // "" for empty, "x", for x, "o" for o.

      var bottomLeft = "" // "" for empty, "x", for x, "o" for o.
      var bottomMiddle = "" // "" for empty, "x", for x, "o" for o.
      var bottomRight = "" // "" for empty, "x", for x, "o" for o.
```

```
    func isLegal(x: Int, y: Int, player: String) -> Bool {

    }
    func possibleMoves(forPlayer: String) -> [(x: Int, y: Int)[ {

    }
    func performMove(x: Int, y: Int, player: String) {

    }
}
```