**Connect Four**

You know the game. If not, go play Connect Four online.  
In making this game, we will use arrays (an array or arrays) to keep track of player positions.

1. Start writing the method that will draw the game board out.  
   Draw the vertical lines.  
   Draw the horizontal lines.  
   Main method you will use is *drawLine(x1,y1,x2,y2)* . This draws a line from point x1,y1 to point x2,y2.  
   You will come back to draw out the game pieces later.
2. Decide on what type of 2d array you will use.  
   A 2d array of type String? int? bool? How will you keep track of empty and player pieces on the board?  
   Create the 2d array.  
   In the constructor, fill the array with some random data. You can fill the entire 2d array randomly or just manually set a few locations to some values. This will allow you to test your draw method.
3. Go back to your draw method and try to draw out the game pieces.  
   The main method you will use is *fillOval(x,y,r)*. Remember that the x,y is not the center of the oval. It is the top left ‘corner’ of the oval.
4. Write a method named *private int checkForWin()* . It will eventually return 0, 1, or 2 to tell us if someone has won the game. But for now, only fill it with *System.out.println(“Checking for win”) and return 0.*
5. Write the game code. After *Start Game* is pressed, it should be player 1’s turn and when buttons are pressed you should ‘drop’ pieces down if there is room in that column. Maybe when a column is full you should disable the button so it cannot be used. After a piece is dropped, you should call the draw method and then *checkForWin()*.
6. Complete the *checkForWin()* method.   
   This one isn’t too bad, but it will require almost a screen of code.  
   Consider the various ways a player can win – there are four different directions.   
   Maybe you can split this up into 4 sub methods to help you?  
   Maybe you are good at math logic and loops and can write just one clever loop?
7. Test out check for win carefully, especially the *boundary conditions*. This is where a win starts or ends along an edge. You have four edges to test out for each of the four possible winning directions. Not a 2 minute task!
8. Make sure to control the GUI properly, reset the board between games, etc.  
   Make methods for some of these tasks to keep your code clean and organized.

For those that want a little more challenge and respect on the streets:  
  
**Board Size Options**  
Give your menu an ‘adjust grid size’ feature that lets the player play on regular, large, or super large board. Make your draw code adjust by doing a little more simple math. Now you have more variety and long, more challenging games.  
  
**Filled Board Game Slots**  
Give your game an option where when you start the game, a few board locations are already filled with pieces. This will give the game a little more variety so player’s can’t just do the same moves every time.  
  
**Game Playback**Remember the state of the board at every move. Add buttons that allow players to ‘go back’ to previous moves.