SUMMARY
In this assignment you will write a program to play the game Connect Four.

DEADLINE
This assignment is due on Wednesday, Nov 13 at 11:00 pm.

DESCRIPTION
Connect four is a popular game played by dropping checkers into a vertical grid with six rows and seven columns. The checkers can be dropped in any of the seven columns (permitted they are not full). Once dropped, the checker falls to the lowest unoccupied space. The winner is the first player to occupy four consecutive spaces in a straight line, either horizontal, vertical, or diagonal.

SPECIFICATIONS
In this project you will write a program to play Connect Four using a two-dimensional array of integers to represent the grid and a Picture object to display the game. You will create a class called ConnectFour to represent the game. The class PlayConnectFour (provided) will allow the user to play the game. When run, this program will display the game board and ask the user to specify a column. The user is always player 1 (red checkers) and goes first. After the user enters a column the opponent will select a column for their piece.

For this project you will create a simple naive opponent that selects a column at random. Then the program will update and repaint the board and ask the user to select another column. This will repeat until the game ends in a win, loss, or draw.

To help with this project a Referee class is provided. The referee will look at your grid and tell you whether the game is a win, loss, draw, or is not finished.
Create a new class called ConnectFour.

Fields:

- `private int [][] grid`
- `private Referee ref`
- `private Picture display`
- `private static final Color [] CHECKER_COLORS = {Color.GRAY, Color.RED, Color.BLACK}`

Constructors:

- `public ConnectFour()`
  - This constructor should initialize the grid, Referee, and display. It should also draw and show the initial display.

Methods:

- `public void updateDisplay()`
  - This method should update the display picture (from the grid) and repaint.

- `public void dropChecker(int column, int player)`
  - This method should drop a checker for player `player` in column `column`.

- `public int getWinner()`
  - This method will simply ask the referee to find the winner. It will return
    - 1 for player one (user) wins
    - 2 for player two (opponent) wins
    - 3 for a draw
    - 0 if the game is not finished

- `public void opponentMove()`
  - This method will carry out an opponent move. The opponent should choose a column at random and drop a checker in that column.
**SOME DETAILS**

**Grid**

The grid will be stored as a two-dimensional array of integers. (each location represented by `grid[x][y]`) Each position in the grid is either 0, 1, or 2. 0 means there is no checker in that space. 1 means there is a player one (red) checker in that space. 2 means there is a player 2 (black) checker in that space. You will use the grid to keep track of the state of the game. You will also use the grid to update the display Picture.

**Display**

The display will be a picture object 700 pixels wide by 600 pixels high. You should begin by painting the entire picture yellow. Then you can use the graphics class to draw circles in the correct places for checkers or missing checkers. Refer to Lab 6 and the methods `fillOval` and `drawOval` for how to draw the circles. Each circle should be evenly spaced and have a diameter of 70 pixels. After using `fillOval`, use `drawOval` to draw a black edge on the circle.
EXAMPLE RUN

> java PlayConnectFour
Column: 1
Column: 2
Column: 2
Column: 5
Column: 4
Column: 3
Column: 3
Column: 2
Column: 5
Column: 5
Player 1 Wins!
>

END DISPLAY
SUBMISSION

Submit the file

    ConnectFour.java

with your method added by the deadline given above. Place all files for submission in a directory in your cs account. Submit the entire directory with the following command:

    submit102 assign2 MY_DIRECTORY

For example, if you named your directory assignment-02, and your current working directory contained the assignment-02 directory, you would submit with the following command:

    submit102 assign2 assignment-02