**Java**

**Tic Tac Toe**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Part 1: Create a board, and a method to display the board.**

In the beginning of the program, create a 3 x 3 array of chars. Fill it with the

character '-'.

Create a separate method: public static void printBoard(char[][] board)

It should print out all of the characters in the parameter in a grid format using row

major order. Display the row and column numbers so the user knows which space is

which.

Here is are two examples of what the method could print:

Before anyone plays: Somewhere in the middle of the game:

1 2 3 1 2 3

------------- -------------

1 | | | | 1 | X | O | X |

------------- -------------

2 | | | | 2 | | O | |

------------- -------------

3 | | | | 3 | O | | |

------------- -------------

Remember: Before a space is taken up there is a '-' in the array, you want to show it as a space though.

You can assume that all of the chars in the array have the value '-','X', or 'O'.

**Part 2: Allow two users to play the game.**

In the main method you will need a variable to remember whose turn it is. This could

be a char, or a boolean.

Create a method that places an x or an o inside the array only if that space was

previously blank:

/\*\*

\* places the specified character in the specified space in the given

\* array and returns true if that space was previously blank.

\* Otherwise returns false and does not change the array

\*

\* @param xo the char to place. Either 'X' or 'O'

\* @param row the row to place xo

\* @param col the column to place xo

\* @param board the array that represents the tic tac toe board

\* @return true if the char was successfully placed, false if not

\*/

public static boolean place(char xo, int row, int col, char[][] board)

Create a loop in the main method that:

1) Prints the board

2) Prints whose turn it is

3) Asks the current player to enter a row and a column. They should NOT

need to enter x or o, your program should automatically switch between the two players' turn's. The turn should not switch if the method place returns false.

At this point two people should be able to play the game, but not check who wins.

**Part 3: Check to see who wins.**

Create a method to check to see if someone has won the game yet:

/\*\*

\* Checks the given tic tac toe board to see if a player has won

\* @param

\* @return

\* 'X' if there are three X's in a row

\* 'O' if there are three O's in a row

\* '-' if no one has won yet and there are still some empty spaces

\* 'T' if no one has won yet and all of the spaces are taken up

\*/

public static char checkWin(char[][] board)

Call this method in the loop you have created to play the game every time a player

makes a move.

If the method returns anything besides a '-', tell the players who wins or if it was a tie.

**Part 4: Reset the board and keep score.**

Create a method that resets the board to its original state. Use it after someone

wins the game.

When a new game starts the computer should randomly decide who moves first.

Create a scorekeeper.

**Part 5: You have a few choices for this one.**

**Either make a computer player.**

Keep what you have, but give the user the option to play against the computer.

Also give the user the option to set the computer to easy, medium, or hard.

Easy could be:

The computer player just picks a random row and a random column

Medium could be:

The computer player blocks you when there are two in a row

Hard could be:

The computer blocks you and tries to win. You would have to actually give it

some kind of strategy here. Just think about how you decide where to place a

piece.

**Or make this game in an Applet.**

Make another program for this instead of changing your current one.

There is an example program attached to show how to use the mouse.

**Do both things if you have time.**