Tic Tac Toe

For this project, you will write a program that allows users to repeatedly play the game of Tic Tac Toe against the computer. See **http://en.wikipedia.org/wiki/Tic-tac-toe** for a description of the game of Tic Tac Toe

You should develop your project using Eclipse. Create a new project in Eclipse called **TicTacToe<lastname>** where <lastname> is replaced by *your* last name. For example, if your last name is Turing, your project should be named TicTacToeTuring.

The TicTacToe class

Your program should have one class file containing a main method and several other methods. You should name this class `TicTacToe`. The program should allow the user to repeatedly play games of Tic Tac Toe against a computer opponent. The user should also be allowed to choose whether or not they want to make the first move. The program will display the state of the board before asking for the human player's move and at the end of the game. The game is over when either the computer or the human is able to play three of their symbols in a row. The game is also considered over when no further moves can be made – this is, of course, a tie.

There should be a total of eleven methods in your program: `main`, `playGame`, `determineFirstPlayer`, `initializeGameBoard`, `processComputerMove`, `processHumanMove`, `displayGameBoard`, `isDone`, `isComputerWin`, `isHumanWin`, and `isTie`.

A starter version of the TicTacToe class is being provided for you. It includes empty versions of the above 11 methods. You should not have to modify the names, parameters or return types for these methods. This starter file also includes declarations for two "global" class-level variables:

1. keyboard - a Scanner object that can be used from any method in the program.
2. board – a char array of size 9 that represents the game board. Positions 0-2 are the top row of the game, 3-5 are the middle row, and 6-8 are the bottom row. There are three valid values for the cells of this array: ' ' (a space) indicates no player has claimed this position, and 'X' and 'O' indicate that the computer ('X') or human ('O') have played that position during a game.

The following is a brief description of the eleven methods needed for the program.

- `main` – The main method should not be very long. It should repeatedly allow the user to play games until the user indicates they don't want to play again. The actual playing of the game is handled by calling the `playGame` method.

- `playGame` – This method has been provided for you in its entirety. **No changes should be made to this method**.

- `determineFirstPlayer` – This method asks the user who should go first. It returns a ‘c’ or an ‘h’ depending on who will move first. Regardless of who moves first, the computer is always ‘X’ and the human is always ‘O’.

- `initializeGameBoard` – This method set the board array to all spaces

- `ProcessComputerMove` – This method randomly selects an **empty** board position and places an ‘X’ in the corresponding position of the `board` array.

- `ProcessHumanMove` – This method prompts the user for a valid, empty board position. The user should be repeatedly prompted for a position until they enter a number that is in range (0-8) and is not already occupied.

- `DisplayGameBoard` – This method prints out the current game board.

- `isDone` – This method returns true if the game is over or false if not. The game is over if one of the following three methods (`isComputerWin`, `isHumanWin` or `isTie`) returns true.

- `isComputerWin` – This method returns true if three cells in a row contain an X; otherwise it returns false.

- `isHumanWin` – This method returns true if three cells in a row contain an O; otherwise it returns false.

- `isTie` – This method returns true if all 9 cells have been filled with O’s or X’s and there is no winner.

Sample run 1

```
Welcome to Tic-Tac-Toe!!
Would you like to play a game? (enter 'yes' or 'no'): yes

Who should move first? (c=computer h=human): c
The computer is X, the human is O.

The computer chooses cell 0.

 X |   |
---+---+---
   |   |
---+---+---
   |   |

Enter an empty position number (0-8): 0
```

```
Invalid move.

 X |   |
---+---+---
   |   |
---+---+---
   |   |

Enter an empty position number (0-8): 1
The computer chooses cell 3.

 X | O |
---+---+---
 X |   |
---+---+---
   |   |

Enter an empty position number (0-8): 2
The computer chooses cell 8.

 X | O | O
---+---+---
 X |   |
---+---+---
   |   | X

Enter an empty position number (0-8): 5
The computer chooses cell 6.
The computer wins! Humans are not very bright.

 X | O | O
---+---+---
 X |   | O
---+---+---
 X |   | X

Game over!
Would you like to play again? (enter 'yes' or 'no'): no
Goodbye!
```

Sample run 2

```
Welcome to Tic-Tac-Toe!!
Would you like to play a game? (enter 'yes' or 'no'): yes
```

```
Who should move first? (c=computer h=human): h
The computer is X, the human is O.


   |   |
---+---+---
   |   |
---+---+---
   |   |

Enter an empty position number (0-8): 0
The computer chooses cell 8.

 O |   |
---+---+---
   |   |
---+---+---
   |   | X

Enter an empty position number (0-8): 2
The computer chooses cell 4.

 O |   | O
---+---+---
   | X |
---+---+---
   |   | X

Enter an empty position number (0-8): 1
The human wins! The human must have cheated.

 O | O | O
---+---+---
   | X |
---+---+---
   |   | X

Game over!
Would you like to play again? (enter 'yes' or 'no'): yes

Who should move first? (c=computer h=human): c
The computer is X, the human is O.

The computer chooses cell 5.

   |   |
---+---+---
   |   | X
---+---+---
   |   |

Enter an empty position number (0-8): 12
Invalid move.

   |   |
---+---+---
   |   | X
---+---+---
   |   |

Enter an empty position number (0-8): 4
The computer chooses cell 2.

   |   | X
---+---+---
   | O | X
---+---+---
   |   |

Enter an empty position number (0-8): 8
The computer chooses cell 6.

   |   | X
---+---+---
   | O | X
---+---+---
 X |   | O

Enter an empty position number (0-8): 1
The computer chooses cell 7.

   | O | X
---+---+---
   | O | X
---+---+---
 X | X | O

Enter an empty position number (0-8): 3
The computer chooses cell 0.
A tie! The human got lucky.

 X | O | X
```

```
---+---+---
 O | O | X
---+---+---
 X | X | O

Game over!
Would you like to play again? (enter 'yes' or 'no'): no
Goodbye!
```

Instructions on how to submit your project and extra credit opportunities will be available shortly

Submitting your project

The project is by the end of day on **TBD**. Of course, early submissions are welcome.

At that time you need to hand in a **printout** of the following (stapled together):

1. Your source code for `TicTacToe.java`
2. Sample output from running your program. Your output example should include as many games as necessary to demonstrate the various features of and conditions in you program.

Also, you need to **email** your *eclipse project* to me at **stlauterburg@salisbury.edu** by end of day on the above due date. Create a .zip archive file of the **entire project directory**. You'll find this directory in your Eclipse workspace directory. The .zip file should be named the same as your project, i.e., TicTacToe<lastname>.zip. Unfortunately, once you have a .zip file, you will need to rename it to replace the .zip extension with something like .117. This step is necessary to get around email firewall restrictions. The email subject line should clearly indicate what the email contains, e.g., "COSC 117 Project 3 Submission".