

**[11] 25 points**use files *TicTacToe.java* and *TicTacToeBoard.java*

Write a program that allows two people to play a game of “tic-tac-toe”, with the computer acting as a “referee.” Here is what your program might look like in action:

```
Welcome! Tic-Tac-Toe is a two player game.  
Enter player one's name: Foo  
Enter player two's name: Bar
```

Precisely duplicate the text shown in this PDF. There's (almost <sup>1</sup>) nothing to invent.

<sup>1</sup> After "Enter player xxx's name: " you have to read and process their response.

Players take turns marking a square. Only squares not already marked can be picked. Once a player has marked three squares in a row, he or she wins! If all squares are marked and no three squares are the same, a tied game is declared. Have Fun!

```
Game board:  
|   |  
|   |  
|   |
```

The 4-line `String` inside the green box is the current state of the board. It is returned by the `TicTacToeBoard` template class's `toString()` instance method.

```
It is Foo's turn.  
Pick a row between 1 and 3: 1  
Pick a column between 1 and 3: 1
```

The program class tells the template class "try to play an 'X' in row 1, column 1," because only the template class knows whether that square is available. Since the template class is validating square availability, let's also have it validate the user's inputs' magnitudes, so that all user input validation is done in one place.

```
Game board:  
| X  |  
|   |  
|   |
```

So the program class passes the user's row and column input to a template class instance method, and that instance method updates the board's state and returns some status information to the program class. As we develop the program class, we'll figure out the details.

```
It is Bar's turn.  
Pick a row between 1 and 3: 2  
Pick a column between 1 and 3: 2
```

```
Game board:  
| X  |  
|  O |  
|   |
```

It is Foo's turn.  
Pick a row between 1 and 3: 2  
Pick a column between 1 and 3: 2  
**ILLEGAL CHOICE! TRY AGAIN...**  
Pick a row between 1 and 3: 2  
Pick a column between 1 and 3: 1

Illustrates the program class's behavior when the template class's instance method returns a status that says "that input was invalid."

```
Game board:
| X   |
| X O  |
|      |
```

It is Bar's turn.  
Pick a row between 1 and 3: 3  
Pick a column between 1 and 3: 1

```
Game board:
| X   |
| X O  |
| O    |
```

It is Foo's turn.  
Pick a row between 1 and 3: 3  
Pick a column between 1 and 3: 3

```
Game board:
| X   |
| X O  |
| O  X |
```

It is Bar's turn.  
Pick a row between 1 and 3: 1  
Pick a column between 1 and 3: 3

```
Game board:
| X  O |
| X O  |
| O  X |
```

Game Over - Bar WINS!!!

Illustrates the program class's behavior when the template class's instance method returns a status that says "we have a winner."

In solving this problem, create a **TicTacToeBoard** class that contains a two-dimensional array representing the current state of a tic-tac-toe board. Initially, every element of this array could contain a space character, to indicate it is empty. You should also define a **toString** method in this class to make it possible for a client program to print the board as shown in the above output.

Your main program will alternately let each of the two players make a move, but not allow any move that is illegal — either because a particular square is already occupied, or because a value less than 1 or greater than 3 has been input for the row or column position. After each turn is made the computer should check if any row, column or diagonal contains three 'X' marks or three 'O' marks; if so, a winner should be declared and the game should end at that point.

You may also add the following optional method to **TicToeBoard.java** if you like:

```
// returns true if the board is a tie (optional)  
public boolean isTie() {...}
```