

**Bonus Programming Project01—Individual Project**(Full Score: **0~100** points)

(Due in class, 3/31/Thursday, Midnight at Moodle)

Your name:	Score:
------------	--------

**Modifications to the Multithreaded Tic-Tac-Toe Program** The programs (check Programming Project #2) in Figs. 24.13 and 24.15 implemented a multithreaded, client/server version of the game of Tic-Tac-Toe. Our goal in developing this game was to demonstrate a multithreaded server that could process multiple connections from clients at the same time. The server in the example is really a mediator between the two client applets—it makes sure that each move is valid and that each client moves in the proper order. The server does not determine who won or lost or whether there was a draw. Also, there is no capability to allow a new game to be played or to terminate an existing game.

The following is a list of suggested modifications to Figs. 24.13 and 24.15:

- a) Modify the TicTacToeServer class to test for a win, loss or draw on each move in the game. Send a message to each client applet that indicates the result of the game when the game is over.
- b) Modify the TicTacToeClient class to display a button that when clicked allows the client to play another game. The button should be enabled only when a game completes. Note that both class TicTacToeClient and class TicTacToeServer must be modified to reset the board and all state information. Also, the other TicTacToeClient should be notified that a new game is about to begin so that its board and state can be reset.
- c) Modify the TicTacToeClient class to provide a button that allows a client to terminate the program at any time. When the user clicks the button, the server and the other client should be notified. The server should then wait for a connection from another client so that a new game can begin.
- d) Modify the TicTacToeClient class and the TicTacToeServer class so that the winner of a game can choose game piece X or O for the next game. Remember: X always goes first.
- e) If you would like to be ambitious, allow a client to play against the server while the server waits for a connection from another client.
- f) (for **MORE bonus points**) How about designing a 4x4 Tic-Tac-Toe?

**Submission:**

A **compressed folder-file** that contains:

- Your COMPLETE project/ source file(s) (\*.java files)
- A Complete Report on how to use (compile, run and test) the code