Answer Key Assignment 3

Part 1

a. **Double (datatype in Java):** The Double datatype denotes a double precision 64-bit floating point type (Otherwise known as IEEE 754 floating point type).

b. **Short (datatype in Java):** The Short datatype denotes a 16 bit signed integer type.

c. **Character array:** An array is a container type of fixed number of values of a single type that allows indexed access with contiguous storage. A character array is an array where the individual datatype is character.

d. **Re-compilation:** Conversion of source code to byte code after it has already been compiled once before.

e. **Loop index:** This is a number that is related to the number of loops executed and can hence be used to identify individual loop executions. Usually this is directly the index variable used to keep track of the loop iteration.

Part 4

**EC1.** What conditions will the code in Parts II and III not detect? Describe exactly why this happens, and how the code could be fixed (if needed) so it does detect these conditions.

**Correct Answer:**

Does not detect a tie. A tie occurs when every row, column, and diagonal (major diagonals only) contains at least one X and O. To detect ties, the scoreTTT method should be modified so that in addition to checking every row, column, and diagonal for a winner (all values equal to X or O), it also checks to see if any are missing an X or O. This can be implemented by initializing a boolean state variable tie=true at the beginning of the scoreTTT method. Then, before checking each row/column/diagonal, set two boolean state variables (hasX and hasO) equal to false. Each time you encounter an X or O inside your loop, set hasX=true, or hasO=true, respectively. Then, after finishing the loop, if either hasX or hasO are false, the row/column/diagonal did not have an X or O, so set tie=false. If tie is still true after checking all rows/columns/diagonals, it is not possible for either X or O to win, so a tie has occurred.

*2pts for "tie" or "stalemate"
3pts for describing stalemate condition
5pts for explanation of how to fix it. Explanations may differ.*

**Common Partial Credit answers:**

(2pts) Does not break after a winner. To fix, add some logic to promptUserTTT that checks to see if a winner has been declared after calling scoreTTT. The reason this answer is not worth more points is because scoreTTT already detects an winner.
(4pts, if explained thoroughly) Does not detect if the board is full. To fix, add some logic to promptUserTTT that checks for this condition (If step >= n^2, or using a loop that walks over the TicTacToeArray searching for an empty space). This answer is a special case of the correct answer, since a full board without a winner is a tie.

(2pts) if the student assumed that this question was asking about user input validation, and described conditions they didn't check for.

EC2. What can you say (supported by analysis) about the size N of the TicTacToe array? (Hint: How much longer will it take to play the game as N increases by a factor of two?)

The size of the TicTacToe array follows N^2. The length of time required to play the game is bounded above by the board size, or N^2. The best case time for a game is 2N-1, since it is possible for the first player to win by filling a single row or column of the game board.

6pts for knowing board size (or max length of game) follows N^2
4pts for 2N-1, with explanation that it is the best case time