ASSIGNMENT #5 SOLUTION (Part Two)

```
/**
Board can represents 2D 3*3 array for TicTacToe game.
It can check if someone wins or a cat's game.
It can check if a square has been chosen.
It can also mark an X or O from the player's choice.
*/
class Board
   {
           private int [][] myBoard = new int [3][3];
           public Board()                                                          //Create a 3 by 3 array and use for a tic tac toe board
             {
                     for (int row = 0; row < 3; row++)
                       {
                               for (int column = 0; column < 3; column++)
                                 {
                                         myBoard [row] [column] = 0;
                                 }
                       }
             }
```

```
public void markFirst(int row, int column)                         // markFirst makes places a 2 accumulation for X
  {
        myBoard [row] [column] = 2;
  }

public void markSecond(int row, int column)                        // markSecond makes places a 1 accumulation for O
  {
        myBoard [row] [column] = 1;
  }

public boolean elementMarked(int row, int column)          //    elementMarked returns a true if the space has been taken
  {
        if (myBoard [row] [column] == 0)        return false;
        else      return true;
  }

/*
Win constructor checks if someone wins.
Here are the meanings of each return type
    o   'None' means no winner;                                    o   'Second' means O won;
    o   'First' means X won;                                       o   'Cat' means a Cat's game.
*/
```

```
public char win()
  {
         char winner = 'None';
         int catCheck = 1;

         for (int column = 0; column < 3; column++)                                         // Check the columns
           {
                  int accumulation = myBoard [0] [column] * myBoard [1] [column] * myBoard [2] [column];
                  if (accumulation == 8)                                                    // 2*2*2 = 8, a win for X
                    {
                           winner = 'First';
                           break;
                    }
                  if(accumulation == 1)                                                // 1*1*1 = 1, a win for O
                    {
                           winner = 'Second';
                           break;
                    }
           }

         if (winner != 'None')      return winner;
```

```
for (int row = 0; row < 3; row++)                                                    // Check the rows
  {
        int accumulation = myBoard [row] [0] * myBoard [row] [1] * myBoard [row] [2];
        if (accumulation == 8)
          {
                winner = 'X';
                break;
          }
        if (accumulation == 1)
          {
                winner = 'Second';
                break;
          }
  }

if (winner != 'None')    return winner;

int accumulation = myBoard [0] [0] * myBoard [1] [1] * myBoard [2] [2];              // Check one diagonal
if (accumulation == 1)   winner = 'Second';
if (accumulation == 8)   winner = 'First';
```

```
        accumulation = myBoard [0] [2] * myBoard [1] [1] * myBoard [2] [0];                    // Check the other diagonal
        if (accumulation == 1)    winner = 'Second';
        if (accumulation == 8)    winner = 'First';

        if (winner == 'None')                                                     // If nobody's won, Check for a cat's game
          {
                for (int row = 0; row < 3; row++)
                  {
                        for (int column = 0; column < 3; column++)
                          {
                                catCheck *= myBoard [row] [column];
                          }
                  }
                if (catCheck != 0)          winner = 'Cat';                                // any empty space is a zero
          }

        return winner;
  }

public String toString()                                                         //toString enables printing out of the board
  {
        String printBoard = "";
        char XorO;
```

```
          int position = 49;                                              // In ASCII, 49 stands for number 1

          for (int row = 0; row < 3; row++)
            {
              for (int column = 0; column < 3; column++)
                {
                  if (myBoard[row] [column] == 1)
                      XorO = (char) (myBoard [row] [column] + 78);          // In ASCII, 79 stands for an O: (78+1)
                  else
                      if (myBoard[row] [column] == 2)
                          XorO = (char) (myBoard [row] [column] + 86);      // In ASCII, 88 stands for an X: (86+2)
                      else
                          XorO = (char) (position);

                      position++;

                  printBoard = printBoard + XorO + " ";
                }
              printBoard = printBoard + "\n" ;                              // starts a new line at the end of a row
            }
          return printBoard;
      }                                                                             // The end of String
  }                                                                                 // The end of class
```