CSCI 2170 Homework 3 Due: midnight, Monday, March 2nd
(This homework counts 200 points)

This homework is modified based on problem 2 on page 660 of the textbook

A software company that develops games has just hired you! Before working on the next version of *Medieval Menace* they have given you the task of implementing the tic-tac-toe game in C++. Tic-tac-toe consists of 3x3 game board of squares whether each square is either empty, has an X marker, or has an O marker. Two players represented by an X or an O play the game. The objective is for one player to get three Xs or three Os in a row first.

Design, implement, and test a TTT ADT as a C++ class that represent a tic-tac-toe game board as well as the X and O markers. Your class should provide suitable observer and mutator methods for modifying the game board and displaying the state of the game. An example ttt.h file is attached. You may download this file from the course web page as well.

Use your class to create a game that prompts for player X and player O to place 'X' or 'O' markers at specified locations on the game board. After each move, your program should display the current game board to the user. Your program should also check after each move if there is a winning configuration of the game board. If so, the game should complete by indicating which player won.

Name the header file of the class: **ttt.h**, the implementation file: **ttt.cpp**, and the client program **main.cpp**.

To see how this program should run, you can download the executable program for this assignment from the course page. Since the executable is compiled and generated in the ranger system, it will only run properly within the ranger environment.

Electronically submit the three programs in D2L Dropbox named "Homework 3". You are only required to submit the three source files: **ttt.h, ttt.cpp** and **main.cpp.**

```
#ifndef  TTT_H
#define  TTT_H

/* Minimally, you need these operations for the TTT class
   You are free to add additional methods if needed */
class TTT
{
  public:
     TTT();
     // Initialize each square of the gameboard to ' '

     void Display() const;
     // Display the 3 x 3 gameboard on screen

     char CheckWon() const;
     // Check whether one player won:
     // if player 'X' wins, return 'X',
     // if player 'O' wins, return 'O',
```

```
    // if no player wins yet, return 'N'.

    bool Assign(int x, int y, char player);
    // If the gameboard position row x and col y already has a piece on it, operation "Assign" fails,
return false
    // Otherwise, assign player ('X' or 'O') at gameboard position row x and col y; then return true;
    // x (IN): row number to assign the next piece
    // y (IN): column number to assign the next piece
    // player (IN): 'X' or 'O'

    bool BoardFull() const;
    // Check to see if there is any blank square left on the gameboard (to continue to play).
    // Returns true of false depending on whether the gameboard is full or not

  private:
    char gameBoard[3][3];  // The game board
};
#endif
```