Agenda / Learning Objectives:
1. Understand the benefits of pair programming.
2. Run the following command to get the tic-tac-toe program (note the
dot):
   
   \texttt{cp \sim ctse/cs211/tic-tac-toe .}
3. Think about how to approach in answering question 9 from ch.5 of the
   Absolute C++ textbook. We can first sketch out our ideas with a
   pseudocode. (The solution code will be released later on today but let’s
   think about how it can be done first without looking at the answer.)

9. TIC-TAC-TOE (from chapter 5 of Absolute C++)
This program is to ask for moves alternately from players X and O. The display should be board-like:

\begin{array}{ccc}
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9 \\
\end{array}

The players enter their moves by entering a number corresponding to the place to be marked. After each
move, the program displays the changed board. After several moves the board may appear:

\begin{array}{ccc}
X & X & O \\
4 & 5 & 6 \\
O & 8 & 9 \\
\end{array}

This can be done with an array of one dimension to hold the game, a playGame function that calls a play
function, which in turn calls a display function and a scoring function.

The play function has a \texttt{char} parameter, with the values \texttt{'X'} and \texttt{'O'} for permissible arguments.
The class has private functions \texttt{score()} and \texttt{display()}, and a \texttt{char} array as a data member.

The main function declares the array to hold the game, and initializes the array. The main function then
calls \texttt{playGame()}. The \texttt{playGame()} function alternately calls \texttt{play('X',array)} and
\texttt{play('O',array)}. The function \texttt{play} prompts players for input, modifies the data in \texttt{array},
calls \texttt{displayArray()}, then runs the \texttt{score()} member which returns \texttt{'X'} if player X wins,
\texttt{'O'} if player O wins, \texttt{'T'} for tie, or \texttt{'\null'} (NULL) for no winners. The \texttt{score()} function indicates
the number of plays remaining.

One thing used here that is not covered in the text but works on all systems having a standard C/C++
library. The library call \texttt{int system(char[]str)} function executes any shell command. The
argument is any \texttt{cstring} that contains the shell command. (A shell command is any command you
use to do something at the system's prompt, $, such as \texttt{dir, ls, clear} or \texttt{cls}, etc.)

This program uses \texttt{system("cls")}, because most Windows based IDEs run programs in a DOS
window. The respond to “cls” by clearing the screen and placing the cursor in the upper left hand
position of the screen. If you are using Linux or some variant of UNIX™ you should use
system("clear"), since the UNIX and Linux shell command to clear the screen is "clear". If
neither of these works for you, you must read the manual for your system and consult a local expert.

Take care not to allow the overwriting of a position already occupied by an opponent with one of your
own. This again points out the necessity for complete testing.