

Programming Assignment 1

Due: Wednesday, September 12, 11:55PM

Objective: To review basic C++ programming, including functions and arrays.

Description:

You will write the code for a console based game of Tic-Tac-Toe in C++. If you do not know how to play Tic-Tac-Toe, the goal is to get three across a row, column, or diagonal. Your game will be constructed for play between a human player and a computer player.

The game board can be simulated with a two dimensional array. A player chooses a square by selecting its number and that square is marked with that player's symbol (X or O). Play then alternates between the players. When someone makes a move that wins the game, play stops. If all nine squares have been taken and no one has won, the game is a draw.

Make sure you show the game board after the final play. For simplicity, it is okay to have the human player go first and always be X.

Sample Output:

A sample output of a program that plays the game is shown here (with user input in bold):

```
0 | 1 | 2
-----
3 | 4 | 5
-----
6 | 7 | 8
Enter your move: 4

0 | 1 | 2
-----
3 | X | 5
-----
6 | 7 | 8

Computer's turn

0 | 1 | 2
-----
3 | X | 5
-----
6 | O | 8
Enter your move: 9
That is not a valid move.
Enter your move: 3

0 | 1 | 2
-----
X | X | 5
-----
6 | O | 8

Computer's turn
```

```

0 | O | 2
-----
X | X | 5
-----
6 | O | 8
Enter your move: 5

```

```

0 | O | 2
-----
X | X | X
-----
6 | O | 8
X wins!

```

Requirements:

Your program must have **input checking** to prevent the user from making incorrect or invalid moves. Every user input must be a value between 0 and 8, and cannot be the value of a square that is already marked. If an invalid value is entered, an appropriate message should be given, and the user prompted to enter a new value.

You should use **appropriate commenting and indentation** in your code, as discussed in your previous programming course. Place your **name and email address** in a comment near the top of your code. Your name handwritten in ink at the top is not acceptable.

You should use at least **two functions** (in addition to main) in this program: one to print the game board and one to check if someone has won the game. Feel free to use additional functions if you feel they are appropriate. Do not use global variables: all variables and structures should be defined in a function and passed to other functions as parameters.

Computer Player:

The simplest implementation for the computer player is probably random choice. Generate a random number between 0 and 8, and check if that square is already marked. If it is not marked, the computer marks that square. If it is marked, generate a new random number and try again.

If you want to have a more intelligent computer player, you could do some research on Tic-Tac-Toe strategies. For instance, the computer's strategy could follow these rules:

1. Play in a square that wins the game
2. Play in a square that blocks the player from winning the game
3. Play in the middle square
4. Play in a corner
5. Play in any other available position

Options:

You can add additional functionality or improve the game in any way that you like, as long as it meets the basic program requirements and does implement a human vs. computer game of Tic-Tac-Toe. Possible enhancements include:

- Let the human decide who should make the first move (X always goes first).
- Allow for multiple games, and keep a record of the number of games won by each player.

- Allow a player to concede a game if they feel that they cannot win.

Submission:

All source code required to build your project should be submitted through Sakai. If there is more than one file in your program, all files should be zipped into a single archive file for submission. Do not submit entire Code::Blocks or Visual Studio projects. Only submit source code (.c, .cpp or .h files).

Every file submitted must have your name in a comment at the top of the file. Do not expect us to derive your name from your Sakai account. We may do a batch download and build from Sakai, which may lose identifying names that are not in the source files.

It would also be helpful to put a comment in your code indicating the compiler or IDE that you used to build the project. 95% of your code will work fine on any compiler, but there are always a few programs that have some compiler dependent code. In case we have problems building your code, it would be helpful to know which compiler you used.