

CSCI 1100 – 2017 Computer Science I

Assignment 4 Due: June 15, 2017 at 5:00pm Submit on Brightspace

Remember that assignments are intended to be your own work!

It is okay to ask for help, to talk with TAs about a problem, or talk with your classmates about a problem.

It is not okay to submit a solution that is substantially the same as somebody else's solution.

Remember the golden rule of programming plagiarism: talk it out, but take no notes. Walk away for 30 minutes. Then, whatever you can remember after that break is yours; use it.

If you are in doubt, ask before submitting!

Note that because this is the final assignment, due the day after our final class, there are **no extensions** possible after the posted due-date and time.

Question 1.

Write a program that simulates a game of tic-tac-toe. You can see a description (and history) of the game on Wikipedia:

<https://en.wikipedia.org/wiki/Tic-tac-toe>

You can also see a live, playable demonstration of the game by Googling for the phrase “tic tac toe”.

The game board is represented by three rows of three squares each. Your program must use an internal representation of this board, prompt the user for inputs at each player's turn, and update the game board accordingly. After each player takes their turn, a representation of the game board should be printed to output.

There will always be two players, indicated separately by “X” and “O” marks on the board. You should allow the user to enter row and column numbers to specify where each player would like to make their mark in turn. Players may not make a mark on an already-used square. Continue prompting the user for player input until all squares are occupied with a mark - it will be up to the players to determine who won the game, if anyone! Your program should force the user to enter valid numerical input (i.e., out of range row/column values), continuing to prompt for the same player move until valid input is given. You may assume that the user will always input integer values: you do not need to account for arbitrary strings or non-integer numbers. The program will end only after the last move is made.

You should print out text that visually represents the board after each turn. The exact representation is up to you; however, you may wish to re-use part of your code (cite your original code, if you do this!) from Exercise 5 of Lab 7.

Make sure that it is made *clear* to the user how they should provide input: use meaningful and well-formatted messages in your prompts.

For instance, the first three turns of your program could produce output that looks something like this:

Let's play tic-tac-toe!

```
-----
| | | |
-----
| | | |
-----
| | | |
-----
```

Enter two valid integers representing the row and column of the move for Player X:

1 1

```
-----
|X| | |
-----
| | | |
-----
| | | |
-----
```

Enter two valid integers representing the row and column of the move for Player O:

1 1

Enter two valid integers representing the row and column of the move for Player O:

2 5

Enter two valid integers representing the row and column of the move for Player O:

2

2

```
-----
|X| | |
-----
| |O| |
-----
| | | |
-----
```

Enter two valid integers representing the row and column of the move for Player X:

3

3

```
-----
|X| | |
-----
| |O| |
-----
| | |X|
-----
```

Bonus Question.

Add to your program from Question 1 so that you can determine the winner (either X's or O's) given any state of the game board. At the end of each move, check whether there is a winner – if so, print the game board, announce the winning player, and end the game. At the end of the game, if there is still no winner, announce a draw.