Name:________________________________ Partner: ___________________________

Python Activity 40: Tic Tac Toe - Board

Bringing together all we've learned this semester into one big project!

Learning Objectives
Students will be able to:
Content:

- Name the input and expected outputs from `Board` and `TTTboard` objects
- Describe how **inheritance** and **abstraction** is beneficial with a specific example

Process:

- Write code that uses `Board` and `TTTboard` objects objects
- Design code that uses inheritance and abstraction effectively

Prior Knowledge

- Python concepts: user-defined classes, inheritance

Concept Model:

CM1. List some board games (in real life) that require a *grid* and *letters*:

What do these games have in common? How are they different?

If we were to implement two of these games using *inheritance*, what attributes/methods might we need in a *parent class*? What about the *child classes*?

Explain how we can use *abstraction* to implement Tic-Tac-Toe:

CM2. a.What is the *initial state* of a Tic-Tac-Toe game?

b. What is an example *"middle of the game" state* for a Tic-Tac-Toe game?

Critical Thinking Questions:

1. *Follow along in class lecture,* and fill out the *Class Object Model* below for the `Board` class: (*Hint:* Be sure to include each method's return type, and the names of any parameters!)

`class Board`

Attributes:

Methods:

2. Given the Class Object Model above, what features of this `Board` class are missing if we wanted to implement a `Tic-Tac-Toe Board` game? (*Hint:* It may be easier to answer this question after you answer questions 3 & 4).

3. How might we *initialize* a `Tic-Tac-Toe Board`? <u>Describe</u> your algorithm: (*Hint:* Refer to CM2a).

4. What are some actions a `Tic-Tac-Toe Board` needs to support for middle-of-the-game play? (*Hint:* Refer to CM2b).

5. Given a 3 x 3 tic-tac-toe board, with the following `row,col` values, describe an algorithm to do the following:

0,0	0,1	0,2
1,0	1,1	1,2
2,0	2,1	2,2

a. *Horizontal win:* Check if each *row* has the same value stored in it:

b. *Vertical win:* Check if each *column* has the same value stored in it:
(*Hint: What might you change from the previous question?)*

c. Check if the *primary diagonal* (i.e., from upper left corner to lower right) has the same value:

d. Check if the *secondary diagonal* (i.e., from upper right corner to lower left) has the same value:
(*Hint: what do the column numbers and row numbers have in common?*)

6. *Follow along in class lecture,* and fill out the *Class Object Model* below for `TTTboard`:
(*Hint:* Be sure to include each method's return types, and the names of any parameters!)

`class TTTboard`

Attributes:

Methods:

Application Questions: Use Python to check your work

(Lab 9 – Boggle is a really good application of these concepts!)