Modularization Design

LAB 01 : TIC-TAC-TOE

Due Saturday at 5:00 PM MST

This week, we will implement a simple game of Tic-Tac-Toe.

Program Description

Tic-Tac-Toe, otherwise known as noughts and crosses, is a game played on a 3x3 grid where one player is represented with Xs and the other is represented with Os. The X player puts a mark in one of the nine grid locations. The O player then selects one of the eight remaining grid locations. The game continues until either one player gets three in a row or three diagonally.

Write a program to represent the Tic-Tac-Toe game. The game begins with an empty board:

```
Enter 'q' to suspend your game. Otherwise, enter a number from 1 to 9
where the following numbers correspond to the locations on the grid:
 1 | 2 | 3
---+---+---
 4 | 5 | 6
---+---+---
 7 | 8 | 9

The current board is:
   |   |
---+---+---
   |   |
---+---+---
   |   |
X>
```

From here, it is X's turn. The X player selects any of the 9 squares by typing a number from 1...9 where 1 represents the upper left corner, 3 represents the upper right corner, and 6 represents the center square. Our user will go for the center:

```
X> 5
   |   |
---+---+---
   | X |
---+---+---
   |   |
O>
```

If the player enters a 'q', then the game is saved and will be reloaded the next time the program is run. The program will then exit after the game is saved.

If a player wins or if all the squares are taken, then the program quits and the saved game is cleared.

Start with...

Please start with the following file: Lab01.py. This file has the header for the program, a couple of constants which will make things easier, and a function which will determine if someone has won the game.

Assignment

Your assignment is to create a working Python program to match the program description. This program must have more than one function. The following are needed to turn this in:

1. Your source code
2. A demonstration video

Source Code

Submit your source code as a file attachment in I-Learn (not as a link to a Google doc or a GitHub share). It must be possible to open this file in Python and execute it without any additional work. This file must have a `.py` extension and be based off the template Lab01.py. You might want to double-check this before pressing [Submit]. In other words, open a fresh Python window and paste the code in to make sure it works as you expect.

At the top of your program, include a comment answering these five questions:

```
# 1. Name:
#      -your name-
# 2. Assignment Name:
#      Lab 01: Tic-Tac-Toe
# 3. Assignment Description:
#      Play the game of Tic-Tac-Toe
# 4. What was the hardest part? Be as specific as possible.
#      -a paragraph or two about how the assignment went for you-
# 5. How long did it take for you to complete the assignment?
#      -total time in hours including reading the assignment and submitting the program-
```

Demonstration Video

Record a short video demonstrating the execution of your program. The video must be very short; no video longer than one minute will be accepted. This means you might need to practice once or twice before recording the video to make sure that you demonstrated everything that is necessary. Also, please make sure you have your face in the video at some point in time. The easiest way to do this is to record the video with "picture-in-picture" turned on.

Your demonstration video must cover the following test cases:

1. Start with a blank file and play a couple of turns.
2. Quit the game in the middle. Then run the program again picking up the game where it was left off.
3. Make X win. Show that the next game starts with a blank board.

After the video is recorded, provide a voice-over mentioning what test case you are covering. Post your video on some streaming service (like YouTube) and provide a link in your assigment submission.

Assessment

Your grade for this activity will be according to the following rubric:

	Exceptional 100%	Good 90%	Acceptable 70%	Developing 50%	Missing 0%
Code Quality 40%	Perfection! The code is extremely easy to understand and very straightforward	Professional and efficient	A few obvious mistakes were made	Readable	Little effort was spent on style. The code looks thrown together or is missing

Functionality 40%	All the test cases execute perfectly and are demonstrated in the video	Everything works, but there are minor cosmetic defects	One test case fails to execute as expected	At least one test case works as expected	Code does not run, is missing, or does not resemble a working solution
Reflection 20%	The reflection component of the assignment completely and concisely describes what went well and what went poorly	It is clear that thought went into the reflection component of the assignment	Each reflection question is addressed, but there is no evidence of introspection	At least one reflection question is answered	There is no reflection in the assignment submission

Note that all the code submissions throughout the rest of the semester will be using a similar process.