**Outline**

Develop a functioning Tic Tac Toe game using basic Python concepts. Extend the game to verify user input and incorporate modularity. Use random number generator to implement a basic AI.

**Objectives**

* Use correct terminology to describe programming concepts;
* Describe the types of data that computers can process and store (e.g., numbers, text);
* Explain the difference between constants and variables used in programming;
* Use variables, expressions, and assignment statements to store and manipulate numbers and text in a program

**Materials**

* Python3 Development Environment at: https://repl.it/
* Python Tutorial at: <http://www.letslearnpython.com/learn/>

**Level 1: Initial Board Output and User Input**

The game board may be implemented may be implemented using the Python List construct. A single row may be represented as follows.

# Definitions of the Game Board Setup

x\_mark = " X "

o\_mark = " O "

blank = " "

# Setup and display of a single row

boardRow = [blank, x\_mark, o\_mark]

print (boardRow)

1. Create a three row representation of the game board by using / modifying the above code.   
   (Hint: Think about lists of lists.)

User input may be collected using the Python code as follows.

# Note: Numbers must be converted using the int() function

Print (“Make a move…”)

xORy = input("X or Y =”)

rowMove = int(input("Row = "))

colMove = int(input("Col = "))

# The move can be added to a row as follows:

boardRow [colMove] = xORy

print (boardRow)

1. Extend your program to collect user input and add the move to your three row game board.   
   Use / modify the sample code.
2. Define (in words) the types of user input that would be an invalid move. List each input example as a separate point.
3. Extend your program to add code that checks for valid user input. Use / modify the sample code provided below.

# If statements can be used to check that the range of input values are correct

if (colMove > 2 ) :

print (“Column value must be between 0 to 2. Please try again “)

if ((xORy != x\_mark ) or (xORy != y\_mark)) :

print (“Mark must be either X or Y. Please try again “)