In [ ]:

```
board = [' ' for x in range(10)]  # in this function I decleard Board by using Lo

def insertLetter(letter,pos):   # pos  = position letter = X and O
    board[pos] = letter

def spaceIsFree(pos):
    return board[pos] == ' '

def printBoard(board):      # this is the board interface availabe for user
    print('  '+ board[1] + ' | ' + board[2]+ ' | ' + board[3])
    print('______________')
    print('  '+ board[4] + ' | ' + board[5]+ ' | ' + board[6])
    print('______________')
    print('  '+ board[7] + ' | ' + board[8]+ ' | ' + board[9])

def isBoardFull(board): # this function used for checking board is full or not
    if board.count(' ') > 1:
        return False
    else:
        return True

def IsWinner(b,l):   # in this function we we computer deciside who is the winner
    return ((b[1] == l and b[2] == l and b[3] == l) or
    (b[4] == l and b[5] == l and b[6] == l) or
    (b[7] == l and b[8] == l and b[9] == l) or
    (b[1] == l and b[4] == l and b[7] == l) or
    (b[2] == l and b[5] == l and b[8] == l) or
    (b[3] == l and b[6] == l and b[9] == l) or
    (b[1] == l and b[5] == l and b[9] == l) or
    (b[3] == l and b[5] == l and b[7] == l))

def playerMove():  # this function is used for players move
    run = True
    while run:
        move = input("please select a position to enter the X between 1 to 9 : ")
        try:
            move = int(move)
            if move > 0 and move < 10:
                if spaceIsFree(move):
                    run = False
                    insertLetter('X' , move)
                else:
                    print('Sorry, this space is occupied')
            else:
                print('please type a number between 1 and 9 : ')

        except:
            print('Please type a number')

def computerMove():    # this function is used for computers move
    possibleMoves = [x for x , letter in enumerate(board) if letter == ' ' and x
    move = 0

    for let in ['O' , 'X']:
        for i in possibleMoves:
            boardcopy = board[:]
            boardcopy[i] = let
            if IsWinner(boardcopy, let):
                move = i
                return move

    cornersOpen = []
    for i in possibleMoves:
        if i in [1 , 3 , 7 , 9]:
            cornersOpen.append(i)

    if len(cornersOpen) > 0:
        move = selectRandom(cornersOpen)
        return move

    if 5 in possibleMoves:
        move = 5
        return move

    edgesOpen = []
    for i in possibleMoves:
        if i in [2,4,6,8]:
            edgesOpen.append(i)

    if len(edgesOpen) > 0:
        move = selectRandom(edgesOpen)
        return move

def selectRandom(li):
    import random
    ln = len(li)
    r = random.randrange(0,ln)
    return li[r]

def main():   # this is the main  function
    print("Welcome to the game!")
    printBoard(board)

    while not(isBoardFull(board)):
        if not(IsWinner(board , 'O')):
            playerMove()
            printBoard(board)
        else:
            print("sorry you loose!")
            break

        if not(IsWinner(board , 'X')):
            move = computerMove()
            if move == 0:
                print(" ")
            else:
                insertLetter('O' , move)
                print('computer placed an o on position' , move , ':')
                printBoard(board)
        else:
            print("you win!")
            break
```

```python
    if isBoardFull(board):
        print("Tie game")

while True:  # by using this function user can play again again
    x = input("Do you want to play again? (y/n) : ")
    if x.lower() == 'y':
        board = [' ' for x in range(10)]
        print('--------------------')
        main()
    else:
        break

# thank You.... :)

# code by = Subhan Mullani
```

```
Do you want to play again? (y/n) : y
--------------------
Welcome to the game!
    |   |
______________
    |   |
______________
    |   |
please select a position to enter the X between 1 to 9 : 5
    |   |
______________
    | X |
______________
    |   |
computer placed an o on position 1 :
  O |   |
______________
    | X |
______________
    |   |
```

In [ ]:

In [ ]: