CMPUT 175 lab 3

Fall 2023

Class methods

```
# Methods are functions defined within a class.
class Car():
    # the __init__() method is called when an instance of the Class is created
    # it initializes the atributes of the instance with provided arguments if any is passed in
    def __init__(self, color, weight, capacity):
        self.color = color
        self.weight = weight
        self.capacity = capacity

    # methods always have self  as the first parameter.
    def run(self, direction, speed):
        # do something
        return

# Invoke a method using an instance of that Class and a dot operator
# pass in the arguments as you would a normal function, ignoring the self parameter

car1 = Car("red", 2000, 4000)
car1.run("east", 2)
```

Special Variable- `__name__`

To understand how ` if __name__ == “__main__”:` works and its use in python, please go through the following link:

https://www.freecodecamp.org/news/if-name-main-python-example/

Please ask TA if you don’t understand anything.

List

```
# List is a built-in Python data structure.
# to declear a list, use square brackets
list_1 = []

# use append() and insert() to add to the list
list_1.append("apple")
print(list_1) # ["apple"]

list_1.insert(0, "orange") # insert "orange" to index 0
list_1.insert(1, "pear")
print(list_1) # ["orange", "pear", "apple"]

# use pop() and remove() to remove from a list
# pop() remove the element at specified index and returned it. Default is the last element
print(list_1.pop(0)) # "orange"
print(list_1) # ["pear", "apple"]

# remove() removes the first occurance of the specified element
list_1.remove("pear")
print(list_1) # ["apple"]
```

https://www.w3schools.com/python/python_lists.asp

List

```
# Lists can contain any type of data structures, including itself.
ticTacToe = NumTicTacToe()
list_1.append([1, 2])
list_1.append(ticTacToe)
print(list_1) # ["apple", [1, 2], <NumTicTacToe object>]

# items in a list are ordered and mutable
# use indicies to access and change each element
# 0 is the first element and -1 is the last element
print(list_1[0]) # "apple"
print(list_1[-2]) # [1, 2]

# change an element""" """
list_1[1] = "banana"
print(list_1) # ["apple", "banana", <NumTicTacToe object>]

# duplicates are allowed
list_1.append("apple")
print(list_1) # ["apple", "banana", <NumTicTacToe object>, "apple"]
```

https://www.w3schools.com/python/python_lists.asp

List of Lists

```
# to create a list of lists, you can use a for-loop
list_2 = []
for _ in range(4):
    inner_list = []
    for _ in range(4):
        inner_list.append(0)
    list_2.append(inner_list)

print(list_2) # [[0,0,0,0], [0,0,0,0], [0,0,0,0], [0,0,0,0]]

# or use list comprehension
list_3 = [[0]*4 for _ in range(4)]

print(list_3) # [[0,0,0,0], [0,0,0,0], [0,0,0,0], [0,0,0,0]]

# indexing a nested list
print(list_2[0][0]) # fist row first column
print(list_2[2][1]) # third row second column
```

Notes on Lab 3

drawBoard() method: Refer to lab 1 slides if you need help with string formatting

Do not hardcode as board sizes can vary

isWinner() method: Check the sum of every row, column, main diagonal, and anti-diagonal

Do not hardcode the checks as board sizes can vary