Osama Ibrahim (181842)

Theoretical:

Tuples: a type of non-primitive data structures and a built-in data type in python, its used to store multiple items in a single variable, tuple’s collection is ordered and unchangeable and allow duplicate values. items are indexed, first item has index 0, the second item has index 1. Finally, tuples are written between round brackets (). On the other side, lists also store multiple items in a single variable. It’s written between square brackets []. lists are ordered and allow duplicate values. Tuples differ from lists that tuples items are unchangeable while list items are changeable.

2- Break statement: usually used inside the loop and its usage is to get out of the loop, loop is immediately terminated and program transfer to the next statement following the loop.

continue statement used inside the loop and skips the current iteration and moves to the next one

pass keyword is a null statement, it returns no operation, nothing happens when pass statement is executed.

3- The word self: the first parameter of methods that represents the instance of the class. To call attributes and methods of a class, the programmer needs to use self. All the methods of the class have self as the first parameter.

4- docstrings: strings used right after the definition of a function, method, class. They are used to document our code. And are written between "".

5- Multiple Inheritance: when a class is derived from more than one base class. The derived class inherits all the features of the base case.

CODES:

1- def count\_vow(string):

vowels=0

for i in string:

if(i=='a'or i=='i' or i=='e' or i=='u' or i=='o'):

vowels+=1

print("no. of vowels: " + str(vowels))

count\_vow("celebration")

2- def findSum(n):

if n<=1:

return n

else:

return n + findSum(n-1)

print(findSum(5))

3- def Fibonacci(n):

if n < 0:

print("invalid input")

elif n == 0:

return 0

elif n == 1 or n == 2:

return 1

else:

return Fibonacci(n-1) + Fibonacci(n-2)

1- Default constructor VS parameterized constructor

def init (self) // def init(self, ')

2- Class VS Object

Class Course

EtHacking = Course ()

Password Generator:

import random

lower = "abcdefghijklmnopqrstuvwxyz"

upper = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"

numbers = "0123456789"

string = lower + upper + numbers

length = int(input("please enter length of password "))

password = "".join(random.sample(string, length))

print("Password:", password)