Name: rawan osama

Question 1:

1.Tuples are non-primitive data structure. Tuples and lists are the same in stored multiple items in a single variable. Tuples are ordered, unchangeable and allow duplicated values. Tuples are differed from lists that the tuples items are unchangeable, but the list items are changeable. Lists are ordered and allowed duplicate values. Tuples written in ( ) brackets but lists written in [ ] brackets.

2. pass: is a null statement, returns no operation and nothing will happen when it is executed

break: is used inside the loop to get out of the loop.

continue: used inside the loop to skip the current iteration and move to the next one

3. self is the first parameter of method that represent the instance of the class. All methods of the class have self as the first parameter.

4.docstring are strings used right after the definition of a function, method and class. They are used to document our code.

5. multiple inheritance is the class derived from more than one base class.

Example:

Class class1:

Pass

Class class2:

Pass

Class multiclass(class1,class2):

pass

Question 2:

1.false

2.false

3.true

4.true

5.false

Question 3:

1.

def count\_vowels(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("number of vowels are: " + str(vowels))

count\_vowels("celebration")

2.

def sum(x):

if x<=1:

return x

return x + sum(x-1)

print(sum(5))

3.

number = 10

n1, n2 = 0, 1

print("Fibonacci Series:", n1, n2, end=" ")

for i in range(2, number):

n3 = n1 + n2

n1 = n2

n2 = n3

print(n3, end=" ")

print()

4.

sample\_dict={'a':100,'b':200,'c':300}

value=200

if value in sample\_dict.values():

print(str(value) + " is in the dictionary")

else:

print(str(value)+ " not found ")

5.

from cmath import pi

class circle :

    def \_\_init\_\_(self , radius, color) :

       self.radius = radius

       self.color = color

    def getradius (self):

       return self.radius

    def setradius(self,radius):

      self.radius=radius

    def getarea(self):

        return pow(self.radius , 2) \* pi

    def getcolor(self):

        return self.color

    def setcolor(self,color):

        self.color=color

circ=circle(1.0,'red')

class Cylinder(circle):

    def \_\_init\_\_(self , height, radius,color) :

       self.height = height

       self.radius=radius

       self.color=color

    def getheight (self):

       return self.height

    def setheight(self,height):

      self.height=height

Question 4:

1. import string

import random

upper\_letters = list(string.ascii\_uppercase)

lower\_letters = list(string.ascii\_lowercase)

digits = list(string.digits)

def password():

passwordLength = input('Enter password length: ')

while True:

try:

passwordLength = int(passwordLength)

if passwordLength < 8 :

print("Sorry password must be at least 8 chars")

passwordLength = int(input('Enter password length : '))

break

except:

print('Please enter numbers only ...')

passwordLength = int(input('Enter password length: '))

password = []

random.shuffle(upper\_letters)

random.shuffle(digits)

random.shuffle(lower\_letters)

for i in range(round(passwordLength \* .4)):

password.append(upper\_letters[i])

for i in range(round(passwordLength \* .3)):

password.append(digits[i])

for i in range(round(passwordLength \* .3)):

password.append(lower\_letters[i])

password = "".join(password[0:])

print(f"Your password is : \n {password}")

password()

2.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content=

"width=device-width, initial-scale=1.0">

<title>Dark Mode</title>

<style>

body{

padding:0% 3% 10% 3%;

text-align:center;

}

h1{

color: #eb0d51;

margin-top:30px;

}

button{

cursor: pointer;

border: 1px solid #555;

text-align: center;

padding: 5px;

margin-left: 8px;

}

.dark{

background-color: #2f2f2f;

color: #fff;

}

</style>

</head>

<body>

<h1>Ethical Hacking</h1>

<button onclick="myFunction()">Switch </button>

<script>

function myFunction() {

var element = document.body;

element.classList.toggle("dark");

}

</script>

</body>

</html>

Question 5:

1.constructor:

def \_init\_ (self)

Parameterized constructor:

def \_init\_(self,data)

2. class:

Class personName

Object:

Rawan = personName()