1. What is the result of the code, and explain?

>>> X = 'iNeuron'

>>> def func():

print(X)

>>> func()

o/p iNeurron

variable x decalred the ‘ineuron’ and passing in function

2. What is the result of the code, and explain?

>>> X = 'iNeuron'

>>> def func():

X = 'NI!'

>>> func()

>>> print(X)

In python have interpreter its line by line execute the code so first X= ‘iNeuron’ sotred in memory and print statement is outside of function so its execute the output of ‘iNeuron’

3. What does this code print, and why?

>>> X = 'iNeuron'

>>> def func():

X = 'NI'

print(X)

>>> func()

>>> print(X)

NI

iNeuron

The inside the function is first priority so execute the inside function print statement then only its execute the outside of function print statement.

4. What output does this code produce? Why?

>>> X = 'iNeuron'

>>> def func():

global X

X = 'NI'

>>> func()

>>> print(X)

NI

In function declare the global keyword so python interpreter consider the execute of X= ‘NI’ in print statement

5. What about this code—what’s the output, and why?

>>> X = 'iNeuron'

>>> def func():

X = 'NI'

def nested():

print(X)

nested()

>>> func()

>>> X

iNeuron

‘iNeuron’

With in function of print statement execute the X= ‘iNeuron’ because X= ‘Ni’ is only decalare of func() so nested() function consider the X values is iNeuron and the X value also print in ‘iNeuron’

6. How about this code: what is its output in Python 3, and explain?

>>> def func():

X = 'NI'

def nested():

nonlocal X

X = 'Spam'

nested()

print(X)

>>> func()

iNeuron

iNeuron

nested function already created so nested() called that inner fuction X= ‘iNeuron’.The nested function is defined the scope of fun().