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

>>> X = 'iNeuron'

>>> def func():

print(X)

>>> func()

‘iNeuron’ func() is a simple function printing value of X. value of X was given before calling the function so latest value of X will be printed.

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

>>> X = 'iNeuron'

>>> def func():

X = 'NI!'

>>> func() - Nothing

>>> print(X) - ‘iNeuron’

Func neither have return nor print so return nothing.

3. What does this code print, and why?

>>> X = 'iNeuron'

>>> def func():

X = 'NI'

print(X)

>>> func() - ‘NI’

>>> print(X) - 'iNeuron'

Becaue value of X was internally updated but that value was a local value. Global value of X is still ‘iNeuron’

4. What output does this code produce? Why?

>>> X = 'iNeuron'

>>> def func():

global X

X = 'NI'

>>> func() nothin

>>> print(X) ‘NI’

func() neither printing nor printing anything so return nothing but it change global value of X and print ‘NI”

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

>>> X = 'iNeuron'

>>> def func():

X = 'NI'

def nested():

print(X)

nested()

>>> func() ‘NI”

>>> X ‘iNeuron’

func() internally provide value X and print it with nested function so func() print value of C.

Outside Value of X will remail same so print ;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() return ‘Spam’ as internally value of X updated