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

>>> X = 'iNeuron'

>>> def func():

print(X)

>>> func()

Output iNeuron

Variable X containing ‘iNeuron’ and a fuction by name func() just printing variable X

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

>>> X = 'iNeuron'

>>> def func():

X = 'NI!'

>>> func()

>>> print(X)

Output will be

‘iNeuron!’ because X contains ‘iNeuron’ global and then in function func() the value of X is ‘Ni!’ local variable and first function is called and local variable X contains the ‘Ni!’ and doesn’t affect the global variable value.

3. **What does this code print, and why?**

>>> X = 'iNeuron'

>>> def func():

X = 'NI'

print(X)

>>> func()

>>> print(X)

Output will be

NI

iNeuron

Global variable of X contains ‘iNeuron’ and local variable in the function func() X contains ‘NI’ and running the function funct() and printing local variable X and then printing global variable X

**4. What output does this code produce? Why?**

>>> X = 'iNeuron'

>>> def func():

global X

X = 'NI'

>>> func()

>>> print(X)

Output will be

NI

In the function funct() the value is reassigned from local to global variable so the value it contains ‘NI’

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

>>> X = 'iNeuron'

>>> def func():

X = 'NI'

def nested():

print(X)

nested()

>>> func()

>>> X

Output wll be

NI

iNeuron

first the function is run and it will print the local variable value ‘NI” then printing the value of X that is global variable and it contains the value ‘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()

Output will be

Spam

Because the global variable value is changed in the function by ‘Spam’ and so function will be print ‘Spam’