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

>>> X = 'iNeuron'

>>> def func():

print(X)

>>> func()

Output: ineuron

Here we define a variable x holding a string ‘ineuron’ , then we define function called func() and print the value of x by calling the function.

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

>>> X = 'iNeuron'

>>> def func():

X = 'NI!'

>>> func()

>>> print(X)

Output: ineuron

Here we get same output as previous problem because x is a globally defined variable so that value will be taken and given at the output.

3. What does this code print, and why?

>>> X = 'iNeuron'

>>> def func():

X = 'NI'

print(X)

>>> func()

>>> print(X)

Output: ineuron

Ineuron

Here same output is printed twice with the global value of x choosen to print.

4. What output does this code produce? Why?

>>> X = 'iNeuron'

>>> def func():

global X

X = 'NI'

>>> func()

>>> print(X)

Output: NI

Here NI is printed at the output as the same x variable is defined again inside the function as global and assigned with a new value.

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: 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: syntax error : no binding for nonlocal ‘x’ found