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

>>> X = 'iNeuron'

>>> def func():

print(X)

>>> func()

**Answer:**

**iNeuron**

**it simply prints X value**

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

>>> X = 'iNeuron'

>>> def func():

X = 'NI!'

>>> func()

>>> print(X)

**Answer:**

**iNeuron**

**it simply prints X value. since we don't have any print statement inside the function. it doesn't print X that is local to function**

3. What does this code print, and why?

>>> X = 'iNeuron'

>>> def func():

X = 'NI'

print(X)

>>> func()

>>> print(X)

**Answer:**

**NI**

**iNeuron**

**it simply prints X values that are inside and outside function. since we have print statement inside and outside the function. it prints X that is local to function and outside of the function**

4. What output does this code produce? Why?

>>> X = 'iNeuron'

>>> def func():

global X

X = 'NI'

>>> func()

>>> print(X)

**Answer:**

**NI**

**global keyword make the variable X global. it will override the X that is outside of the function**

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

>>> X = 'iNeuron'

>>> def func():

X = 'NI'

def nested():

print(X)

nested()

>>> func()

>>> X

**Answer:**

**iNeuron**

**iNeuron**

**Here, we dont have any global keyword. nested function simply prints the X value this is local to the function. next, func function doesnt have any print statement and X is local to that function. Hence X prints 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()

**Answer:**

**Spam**

**nonlocal keyword says that X is neither local nor global. first, func is called that doesnt have any print statement and followed by nested function that has X value as Spam.**