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

>>> X = 'iNeuron'

>>> def func():

print(X)

>>> func()

**Output:**

**iNeuron**

**Explanation:**

**X is in the global scope, that means that global variable can be accessed inside the function**

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

>>> X = 'iNeuron'

>>> def func():

X = 'NI!'

>>> func()

>>> print(X)

**Output:**

**iNeuron**

**Explanation:**

**Print(X) statement will print ‘iNeuron’ value, func() won’t have an output, because it doesn’t have print or return statement. Local X variable just declared in the function scope.**

3. What does this code print, and why?

>>> X = 'iNeuron'

>>> def func():

X = 'NI'

print(X)

>>> func()

>>> print(X)

**Output:**

**NI**

**iNeuron**

**Explanation:**

**Func() will print X variable that declared in it’s own local scope and print(X) statement will print X variable of a global scope.**

4. What output does this code produce? Why?

>>> X = 'iNeuron'

>>> def func():

global X

X = 'NI'

>>> func()

>>> print(X)

**Output:**

**NI**

**Explanation:**

**Global keywork tells the function to use global variable X. Since it uses global variable, it gets overridden within the function, that’s why print(X) statement outputs ‘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:**

**NI**

**‘iNeuron’**

**Explanation:**

**Nested() function is declared inside the func() along with the local variable X, then nested() function gets executed inside the func() function. Since nested() function doesn’t have any X variable itself, it prints the variable which is in outer scope, which is the local scope of func() function. The global variable remains intact.**

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:

Spam

Explanation:

**Nested() function is declared inside the func() along with the local variable X=’NI’. There is a ‘nonlocal’ keyword inside the nested() function, that means it should use the variable X outside it’s scope. Since nested() function uses variable outside it’s local scope, it overrides variable X with the new value ‘Spam’, then gets executed inside the func() function. Print(X) statement outputs a new value, because it was overridden.**