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

X = ‘iNeuron’

def func():

print(X)

func()

**Ans. Output= iNeuron and func() will execute and print the value of X.**

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

X = ‘iNeuron’

def func():

X = ‘NI!’

func()

print(X)

**Ans. Output= iNeuron because the scope of X=’NI!’ is inside the func() only that’s it takes value if** **globally defined variable type.**

3. What does this code print, and why?

X = ‘iNeuron’

def func():

X = ‘NI’

print(X)

func()

print(X)

**Ans. Output1 = NI & Output2 = iNeuron. For printing value of X inside function it will print locally** **assigned value of X and for outside function it will print the value of globally defined variable.**

4. What output does this code produce? Why?

X = ‘iNeuron’

def func():

global X

X = ‘NI’

func()

print(X)

**Ans. Output= NI because we had defined X as global variable inside function so whatever changes** **we’ll do it will reflect globally.**

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

X = ‘iNeuron’

def func():

X = ‘NI’

def nested():

print(X)

nested()

func()

X

**Ans. Output will be as iNeuron but 2 times because the scope of X=’NI!’ is inside the func() only that’s it takes value if globally defined variable for nested function.**

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()

**Ans. Output = Spam as it X defined as nonlocal so due to this X is considered as global in this case.**