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

>>> X = 'iNeuron'

>>> def func():

print(X)

>>> func()

iNeuron since X is global variable, function accessed it

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

>>> X = 'iNeuron'

>>> def func():

X = 'NI!'

>>> func()

>>> print(X)

iNeuron; though local variable X = ‘Nl!’ is provided since print(X) is mentioned outside the

function, printed the global variable value

3. What does this code print, and why?

>>> X = 'iNeuron'

>>> def func():

X = 'NI'

print(X) NI!

>>> func()

>>> print(X) iNeuron

print(X) inside the function accessed the local variable of function, so Nl!

print(X) outside the function accessed the global variable of function, so iNeuron

4. What output does this code produce? Why?

>>> X = 'iNeuron'

>>> def func():

global X

X = 'NI'

>>> func()

>>> print(X)

NI!

Though print(X) is mentioned outside the function, since global is used inside

the function followed by local variable, local replaced the global variable value.

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

>>> X = 'iNeuron'

>>> def func():

X = 'NI'

def nested():

print(X)

nested() iNeuron, accessing the global variable

>>> func() No output is generated, since func() doesn’t have any operation

>>> X it access the global variable, with print(X) we can see 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()

**SyntaxError:** no binding for nonlocal 'X' found