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

>>> X = 'iNeuron'

>>> def func():

print(X)

>>> func()

We are saving 'iNeuron' as X and defining a function called func() and we are asking the function to print(X)

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

*The global variables are access in side the functions in python. But we can not access function variable out side function.*

*# Since x is golbal variable we are able to print it out side of the function solution = 'iNeuron'*

>>> X = 'iNeuron'

>>> def func():

X = 'NI!'

>>> func()

>>> print(X)

3. What does this code print, and why?

>>> X = 'iNeuron'

>>> def func():

X = 'NI'

print(X)

>>> func()

>>> print(X)

*The global variables are access in side the functions in python. But we can not access function variable out side function.*

*# X is updated with 'NI' which is local to function and its immutable. its name space is with in the function solution = 'NI!', 'iNeuron'*

4. What output does this code produce? Why?

>>> X = 'iNeuron'

>>> def func():

global X

X = 'NI'

>>> func()

>>> print(X)

*since the X in side function is made Global, it will be accesible out side of the function too.*

*#now X will have new value.*

*#solution : 'NI!', '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

*the nested() function will print 'iNeuron', Then func() does not display anything,*

*# and x ='NI' is not accessible out*

*#side the function.*

*#Solution : '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()

*#Nonlocal variables are used in nested functions whose local scope is not defined.*

*#This means that the variable can be neither in the local nor the global scope. it print the updated value from nested*

*#function*

*#Sol : 'spam'*