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

```
>>> X = 'iNeuron'

>>> def func():

print(X)

>>> func()
```

Ans iNeuron is the output of above code because X is global variable and global variables can be accessed within the function

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

```
>>> X = 'iNeuron'

>>> def func():

X = 'NI!'

>>> func()

>>> print(X)
```

Ans iNeuron is the output of above code . In this case 2 variables with the name X are defined. The variable X = 'iNeuron' is global variable and X = 'NI!' is local variable. The scope of local variable ends when the function call ends. Hence printing X prints the value of global variable X having value iNeuron.

3. What does this code print, and why?

```
>>> X = 'iNeuron'

>>> def func():
```

X = 'NI'

print(X)

>>> func()

>>> print(X)

Ans

NI

iNeuron

is printed by above code.When function call is made the value of local variable X which is Ni is printed and the scope oi local variable ends with function call. Again when print statememnt is executed the value of global variable X is whihc is iNeuron is printed.

4. What output does this code produce? Why?

>>> X = 'iNeuron'

>>> def func():

global X

X = 'NI'

>>> func()

>>> print(X)

Ans NI is the output of above code. In the function func the value of global variable X is changed from iNeuron to NI.When call to function is made the value of gloabla variable X is modified, hence printing X prints the modified value of X.

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 NI

'iNeuron'

is output of above code.When func() is called, it defines a local variable X with the value 'NI' and calls the nested() function.Inside nested(), the value of the local X variable from the func() function ('NI') is printed.After that, when print(X) is called in the global scope, the global variable X with the value 'iNeuron' is printed.

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 Spam is output of above code.The nonlocal keyword allows the nested() function to modify the variable X in the nearest enclosing scope, which is the func() function's scope.