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

**Answer :**

>>> X = 'iNeuron'

>>> def func():

print(X)

>>> func()

**Output :**

**iNeuron**

Inside the func function, the value of the global variable X is accessed and printed using print(X). The output will be 'iNeuron', which is the value of X defined outside the function.

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

**Answer :**

>>> X = 'iNeuron'

>>> def func():

X = 'NI!'

>>> func()

>>> print(X)

**Output :**

**iNeuron**

Within the func function, the global keyword is used to indicate that the variable X refers to the global variable defined outside the function. The value of the global variable X is then updated to 'NI!'.

After calling the func function, the updated value of the global variable X is printed using print(X), which outputs 'NI!'.

**3. What does this code print, and why?**

**Answer :**

>>> X = 'iNeuron'

>>> def func():

X = 'NI'

print(X)

>>> func()

>>> print(X)

**Output :**

**NI**

**iNeuron**

Inside the func function, the local variable X is assigned the value 'NI' and then printed using print(X). This prints 'NI' as the output within the func function.

Outside the func function, the global variable X with the value 'iNeuron' is printed using print(X). This prints 'iNeuron' as the final output.

**4. What output does this code produce? Why?**

**Answer :**

>>> X = 'iNeuron'

>>> def func():

global X

X = 'NI'

>>> func()

>>> print(X)

After calling the func function, the value of the global variable X is updated to 'NI'. The print(X) statement outside the function then prints the updated value of X, which is 'NI'

**Output :**

**NI**

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

**Answer :**

>>> X = 'iNeuron'

>>> def func():

X = 'NI'

def nested():

print(X)

nested()

>>> func()

>>> X

**Output :**

**iNeuron**

The first print statement print(X) within the nested function prints the global X variable, which is 'iNeuron'. The second print statement print(X) after calling the func function prints the local X variable within func, which is 'NI'.

**6. How about this code: what is its output in Python 3, and explain?**

**Answer :**

def func():

X = 'NI'

def nested():

nonlocal X

X = 'Spam'

print(X)

nested()

func()

**Output :**

**Spam**

The nested function is defined within the func function, creating the necessary enclosing scope for the nonlocal keyword to work. The nonlocal keyword is then used to indicate that the variable X is defined in the outer scope of the nested function.