**PYTHON BASIC ASSIGNMENT\_22 - SUBMITTED BY SAMUEL DEVDAS**

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

Ans. Output: Ineuron

The variable ‘X’ with the string value ‘X’ is declared. Then a function named as ‘func() is

defined which prints the value of variable ’X.’ The function call ‘func()” prints the value

‘iNeuron’ as the output.>>> X = 'iNeuron'

>>> def func():

print(X)

>>> func()

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

Ans. The function prints ‘iNeuron’, as the ‘print(x)’ code block is executed at the end which takes the Global variable declared in line ‘1’ ie. X=’iNeuron’. Enven though the fuction call ‘func()’ is executed previously, it declares a local variable X=’NI!’ which wont be used outside of the code block of ‘def func()’. >>> X = 'iNeuron'

>>> def func():

X = 'NI!'

>>> func()

>>> print(X)

3. What does this code print, and why?

Ans. In line3 of code the command ‘global X’ makes the scope of the variable ‘X’ available globally ie. It will be available outside the scope of its function ‘func()’. Then the value of variable ‘X’ is assigned as ‘NI’ which gets updated outside the ‘func() code block’ using the call ‘func()’. Hence, line 4- print(X) will result in the output ‘NI’, and line 6- print(X) will result in output ‘iNeuron’.

>>> X = 'iNeuron'

>>> def func():

X = 'NI'

print(X)

>>> func()

>>> print(X)

4. What output does this code produce? Why?

Ans. Here, in line 3, within func() the variable ‘X’ has been made global and value ‘NI’ has been assigned to it. As, a result calling function call ‘func()’ in line 5 will execute X=’NI’ and print it as the output value ‘NI’.

>>> X = 'iNeuron'

>>> def func():

global X

X = 'NI'

>>> func()

>>> print(X)

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

Ans. We see that in this code block, line 3:

def nested():

print(X)

nested()

Here, if func() is called nested function ‘nested()’ runs using the local variable X=’NI’, prints ‘NI’ and runs the function call again nested() which becomes an infinite loop which prints ‘NI’ forever. This seems to get ignored by the compiler when the whole code is run together and only the value of Global variable ‘X=iNeuron’ gets printed as the output at last.

>>> X = 'iNeuron'

>>> def func():

X = 'NI'

def nested():

print(X)

nested()

>>> func()

>>> X

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

Ans. Here it seems that after function ‘func()’ gets called and after execution it gets stuck into an infinite loop within the ‘nested()’ function code block and hence gives no output finally.

>>> def func():

X = 'NI'

def nested():

nonlocal X

X = 'Spam'

nested()

print(X)

>>> func()