

Advance Python Programming & Application

Pass by value / pass by Reference

Global and local scopes

Class Tasks

28-March-2024 (Thursday)

1.

Write a Python function that takes an integer as input and doubles its value inside the function. Call this function with an integer variable as an argument and print the original and updated values of the variable before and after the function call.

2.

Write a Python function that takes a list as input and appends an element to the list inside the function. Call this function with a list variable as an argument and print the original and updated values of the list before and after the function call.

3.

Write a Python program that demonstrates the difference between pass by value and pass by reference. This could involve creating a function that modifies a variable passed by value and another function that modifies a variable passed by reference. Call both functions with appropriate variables and observe the changes.

4.

Write a Python program that declares a variable outside a function and another variable inside the function with the same name. Inside the function, modify the variable. Print both variables before and after the function call to understand how variable scope affects pass by value and pass by reference.

5.

Write a Python program that declares a variable in the global scope and another variable with the same name inside a function. Attempt to access both variables from within the function.

Print the values of both variables before and after the function call to observe the scope differences.

6.

Define a function ``outer_function`` that declares a variable ``x`` and another function ``inner_function`` inside it. The inner function should access the variable ``x`` from the enclosing scope and print its value. Call ``outer_function`` and then call ``inner_function`` to observe how the inner function can access variables from the enclosing scope.

7.

Write a Python program that defines a function ``change_global_variable`` which attempts to modify a global variable ``count``. Declare the ``count`` variable outside the function. Inside the function, use the ``global`` keyword to modify the value of ``count``. Print the value of ``count`` before and after calling the function to see the effect of the ``global`` keyword.

8. Nonlocal Keyword (Nested Functions):

Define a function ``outer_function`` that declares a variable ``x`` and defines another function ``inner_function`` inside it. The inner function should modify the variable ``x`` from the enclosing scope using the ``nonlocal`` keyword. Print the value of ``x`` before and after calling the inner function to observe the effect of the ``nonlocal`` keyword.