**Assigment 1**

Question 1:

Ans: The function returns None by default

If a function doesn't explicitly return a value using return, it will automatically return None.

**Example:**

def greet():

print("Hello!")

result = greet()

print(result)

Output:

Hello!

None

**Question 2: Can a function call another function?**

**answer**: Yes, a function can call another function in Python — this is completely normal and very common!

**✅ Example 1: Simple Function Calling Another**

def greet():

print("Hello!")

def welcome():

greet() # Calling greet inside welcome

print("Welcome to Python!")

welcome()

Output:

Hello!

Welcome to Python!

**Question 3: How can a function take another function as a parameter?**

**Answer:**You can pass a function as a parameter to another function, just like you pass numbers, strings, or lists.

**Basic Example**

def greet():

return "Hello!"

def call\_function(func):

message = func() # Call the passed-in function

print(message)

call\_function(greet)

**Output:**

Hello!

**Question4: Explain the concept of scope local and global variable?**

**Answer:**Two Main Types of Scope

**1. Local Scope**

A variable declared inside a function is local to that function.

It cannot be accessed outside that function.

**Example:**

def greet():

name = "Alia" # Local variable

print("Hello", name)

greet()

print(name) This will cause an error (name is local)

---

**2. Global Scope:**A variable declared outside all functions is global.

It can be used inside and outside of functions.

**Example**

name = "Alia" # Global variable

def greet():

print("Hello", name)

greet()

print(name) # ✅ Works fine

**Question 5: what is different between return and print?**

**Answer**: return vs print — The Difference

**return print**

**Purpose** Gives a value back to the caller Displays a message to the screen

**Used for** Logic and data processing Debugging or user output

**Output visible** Not visible unless you print() it Immediately shows in the console

**Ends function**? ✅ Yes — ends the function immediately ❌ No — function continues running

**Reusability** ✅ You can use the returned value later ❌ You can’t use printed text in logic

---

**Example** 1: return vs print

def say\_hello():

return "Hello"

msg = say\_hello() # Returns the value

print(msg) # Prints the value

**Output:**

Hello

**Question: Will calling the same function multiple times always give the same results?**

**Answer**: Calling the same function multiple times may or may not give the same result — it depends on what the function does and whether it uses any changing input or external factors.

---

**When function calls give the same result**:

If the function: uses only fixed inputs (no randomness),doesn’t depend on global variables or external data, doesn’t involve time or user input,Then calling it multiple times will give the same result.

**Example:**

def add(a, b):

return a + b

print(add(3, 4)) # ➜ 7

print(add(3, 4)) # ➜ 7 (always same)

When function calls give different results:

If the function :uses random numbers, reads user input, checks the current time or date,

modifies or depends on global or external state, Then it can return different results each time.

**Example** 1: Using random

import random

def roll\_dice():

return random.randint(1, 6)

print(roll\_dice()) # Different each time

print(roll\_dice())

Example 2: Using global variable

count = 0

def increment():

global count

count += 1

return count

print(increment()) # 1

print(increment()) # 2 (changes every call)

---

**Summary:**

Case Same Result?

Pure functions (same input → output) ✅ Yes

Random, time, user input involved ❌ No

Changes global or external values ❌ No