Name : Huzaifa Syed

Roll No. F24-611

Program : BSSE

Semester/section : 2nd C

Subject : OOP

Session : Fall 2024

Submitted To : Jamal Abdul Ahad

Department of Computer science

Abbottabad University of Science & Technology

Part 1: Introduction to Functions :

A function is a block of reusable code that performs a specific task. Functions help in organizing code and improving reusability.

Defining a Function :

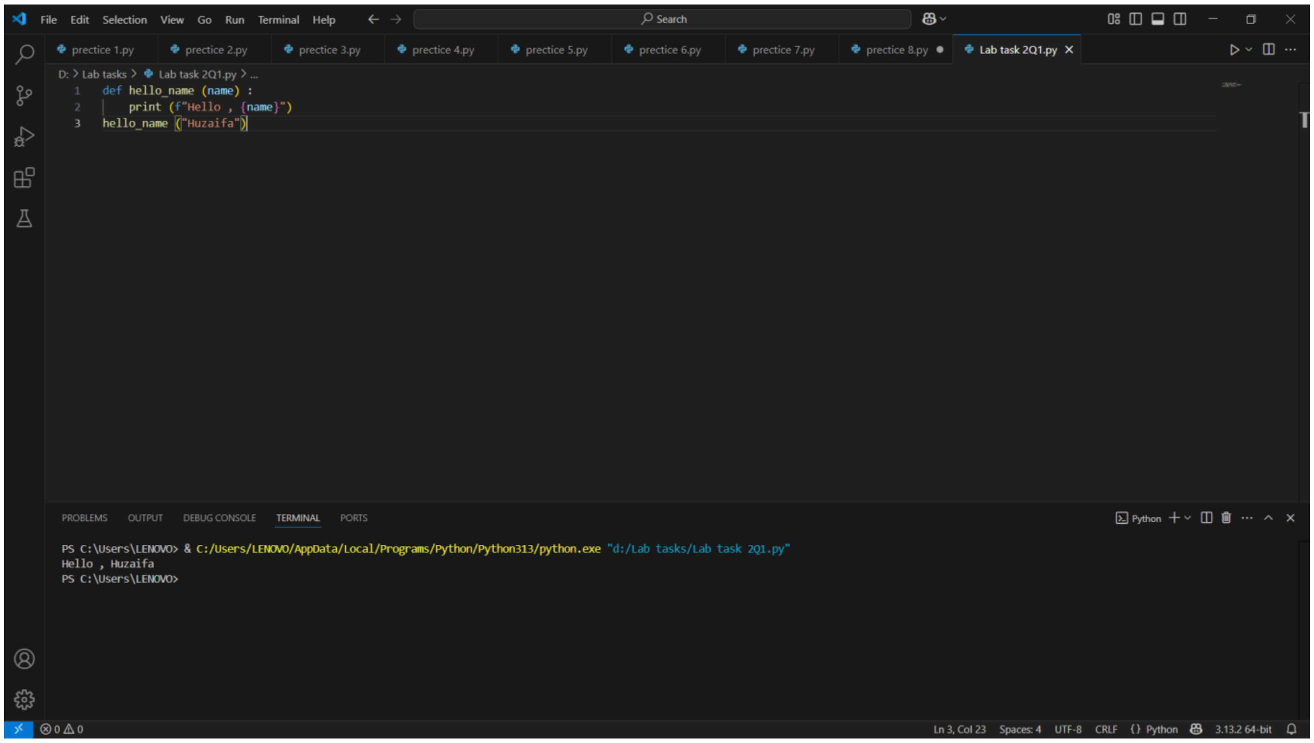
A function is defined using the “**def**” keyword.

```
# Example :
 def greet():
       print("Hello, World!")
greet() # Calling the function
```

Question No. 1 :

Write a function hello_name that takes a name as an argument and prints "Hello, !".

Code :



```
def hello_name (name) :
    print (f"Hello , {name}")
hello_name ("Huzaifa")
```

```
PS C:\Users\LENOVO> & C:/Users/LENOVO/AppData/Local/Programs/Python/Python313/python.exe "d:/Lab tasks/Lab task 2Q1.py"
Hello , Huzaifa
PS C:\Users\LENOVO>
```

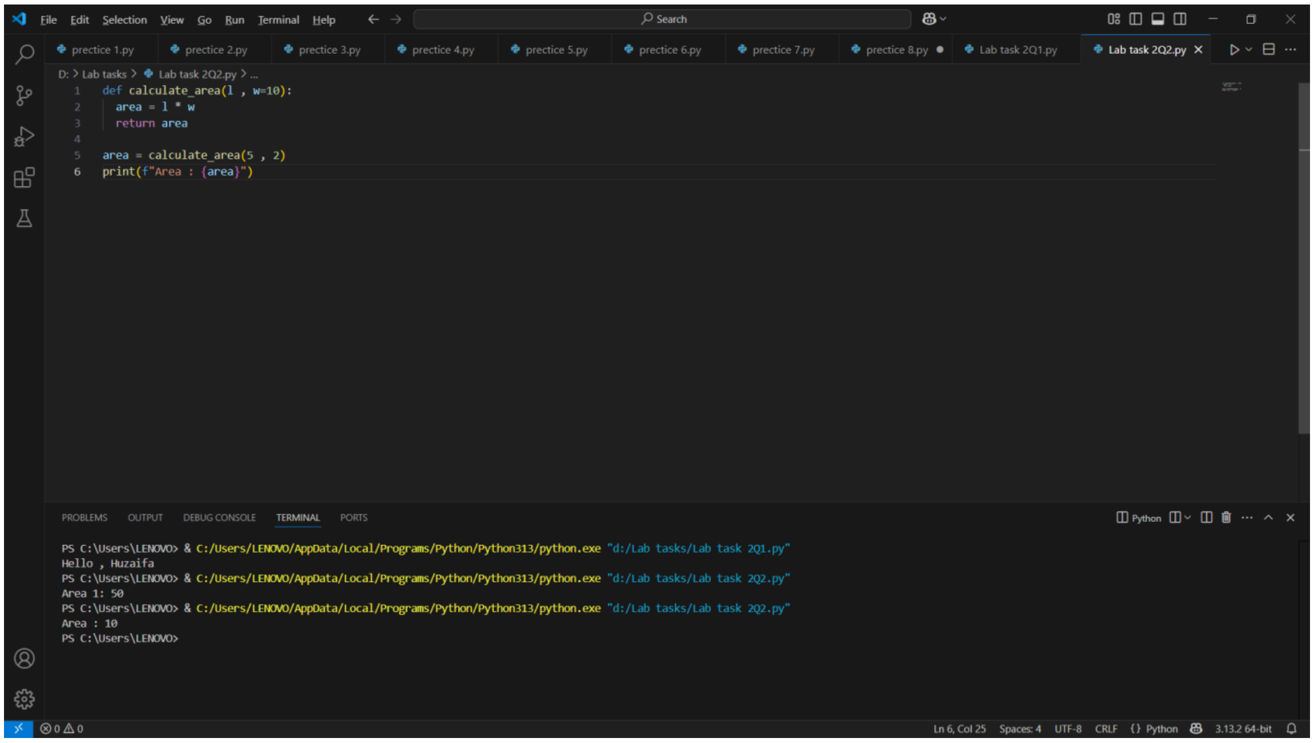
Part 2: Function Arguments :

Functions can have parameters that receive values when called.

Question No. 2 :

Write a function calculate_area that takes length and width as arguments and returns the area of a rectangle. The width should have a default value of 10.

Code :



```python
def calculate_area(l , w=10):
    area = l * w
    return area

area = calculate_area(5 , 2)
print(f"Area : {area}")
```

```
PS C:\Users\LENOVO> & C:/Users/LENOVO/AppData/Local/Programs/Python/Python313/python.exe "d:/Lab tasks/Lab task 2Q1.py"
Hello , Huzaifa
PS C:\Users\LENOVO> & C:/Users/LENOVO/AppData/Local/Programs/Python/Python313/python.exe "d:/Lab tasks/Lab task 2Q2.py"
Area 1: 50
PS C:\Users\LENOVO> & C:/Users/LENOVO/AppData/Local/Programs/Python/Python313/python.exe "d:/Lab tasks/Lab task 2Q2.py"
Area : 10
PS C:\Users\LENOVO>
```

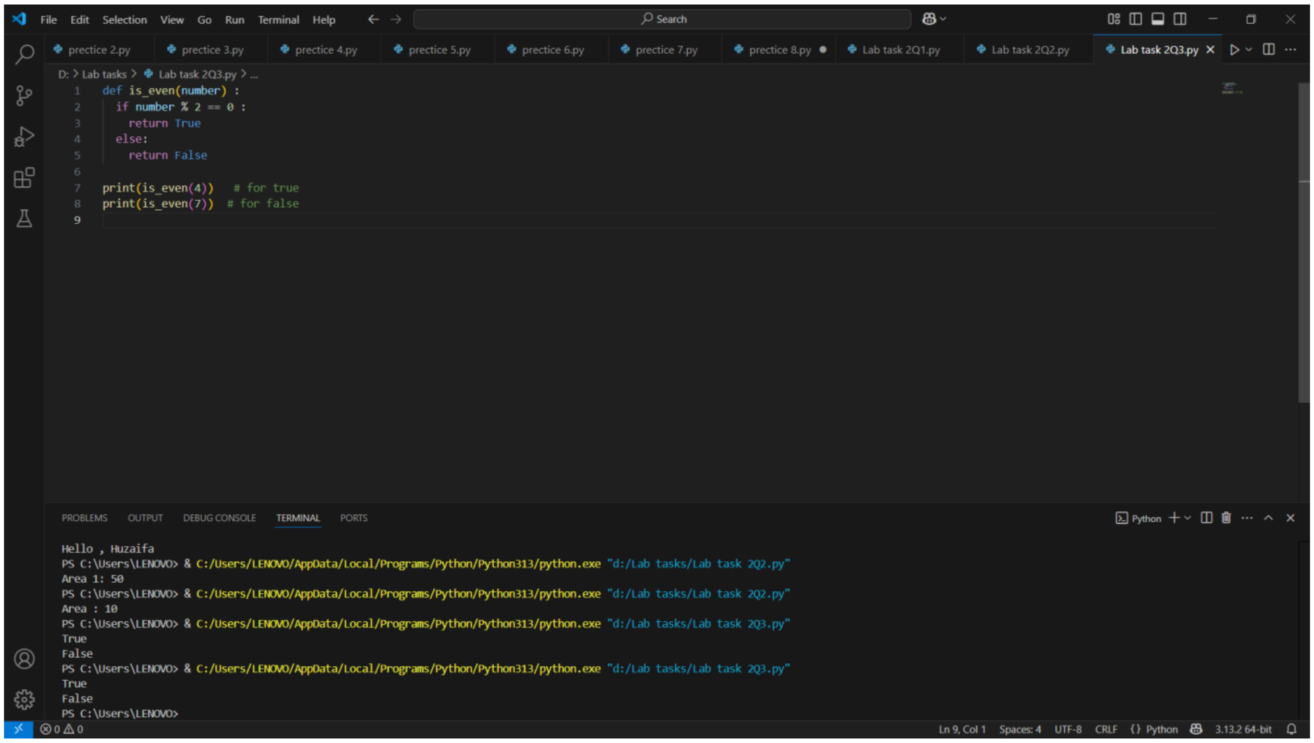
Part 3: Return Values :

A function can return a value using the “**return**” keyword.

Question No. 3 :

Write a function is_even that returns True if a given number is even and False otherwise.

Code :



```python
def is_even(number) :
  if number % 2 == 0 :
    return True
  else:
    return False

print(is_even(4))   # for true
print(is_even(7))  # for false
```

```
Hello , Huzaifa
PS C:\Users\LENOVO> & C:/Users/LENOVO/AppData/Local/Programs/Python/Python313/python.exe "d:/Lab tasks/Lab task 2Q2.py"
Area 1: 50
PS C:\Users\LENOVO> & C:/Users/LENOVO/AppData/Local/Programs/Python/Python313/python.exe "d:/Lab tasks/Lab task 2Q2.py"
Area : 10
PS C:\Users\LENOVO> & C:/Users/LENOVO/AppData/Local/Programs/Python/Python313/python.exe "d:/Lab tasks/Lab task 2Q3.py"
True
False
PS C:\Users\LENOVO> & C:/Users/LENOVO/AppData/Local/Programs/Python/Python313/python.exe "d:/Lab tasks/Lab task 2Q3.py"
True
False
PS C:\Users\LENOVO>
```

Part 4: Variable Scope :

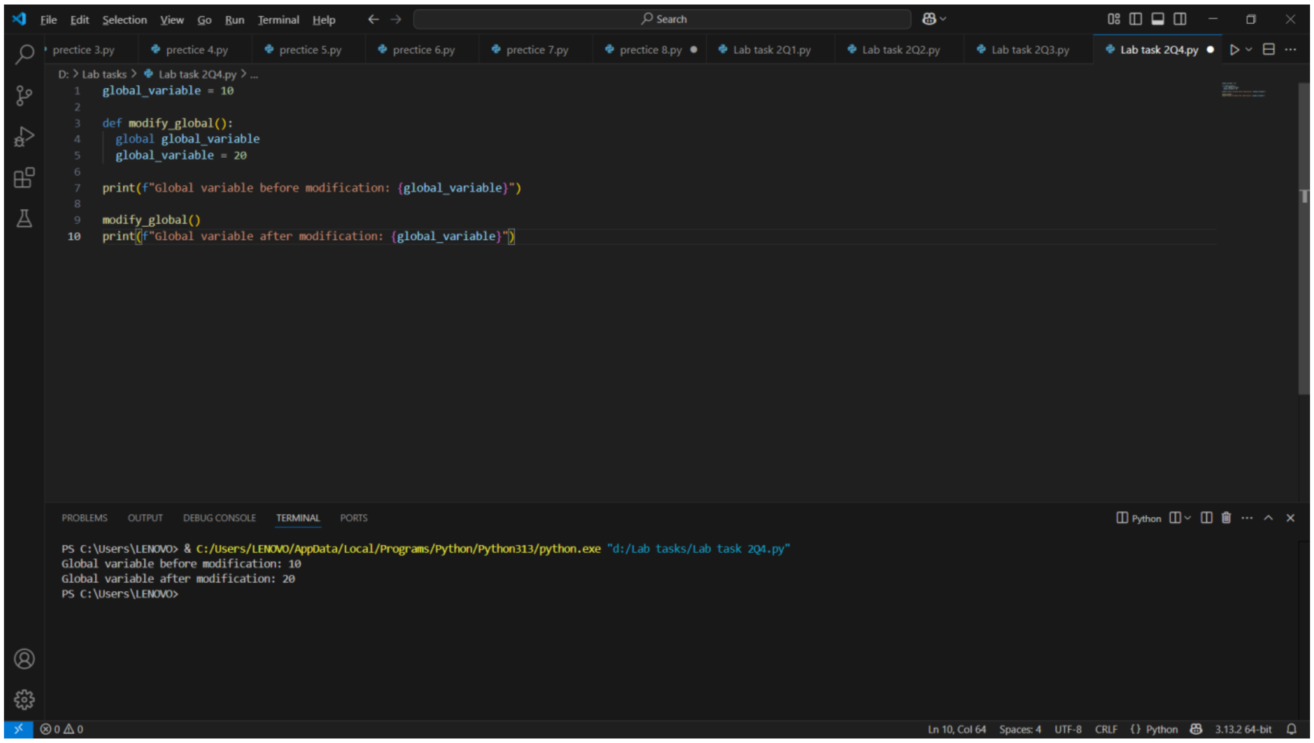
Local and Global Variables …

Variables defined inside a function are local, while those outside are global.

Question No. 4 :

Create a function that modifies a global variable inside a function using the global keyword.

Code :



```python
global_variable = 10

def modify_global():
    global global_variable
    global_variable = 20

print(f"Global variable before modification: {global_variable}")

modify_global()
print(f"Global variable after modification: {global_variable}")
```

```
PS C:\Users\LENOVO> & C:/Users/LENOVO/AppData/Local/Programs/Python/Python313/python.exe "d:/Lab tasks/Lab task 2Q4.py"
Global variable before modification: 10
Global variable after modification: 20
PS C:\Users\LENOVO>
```

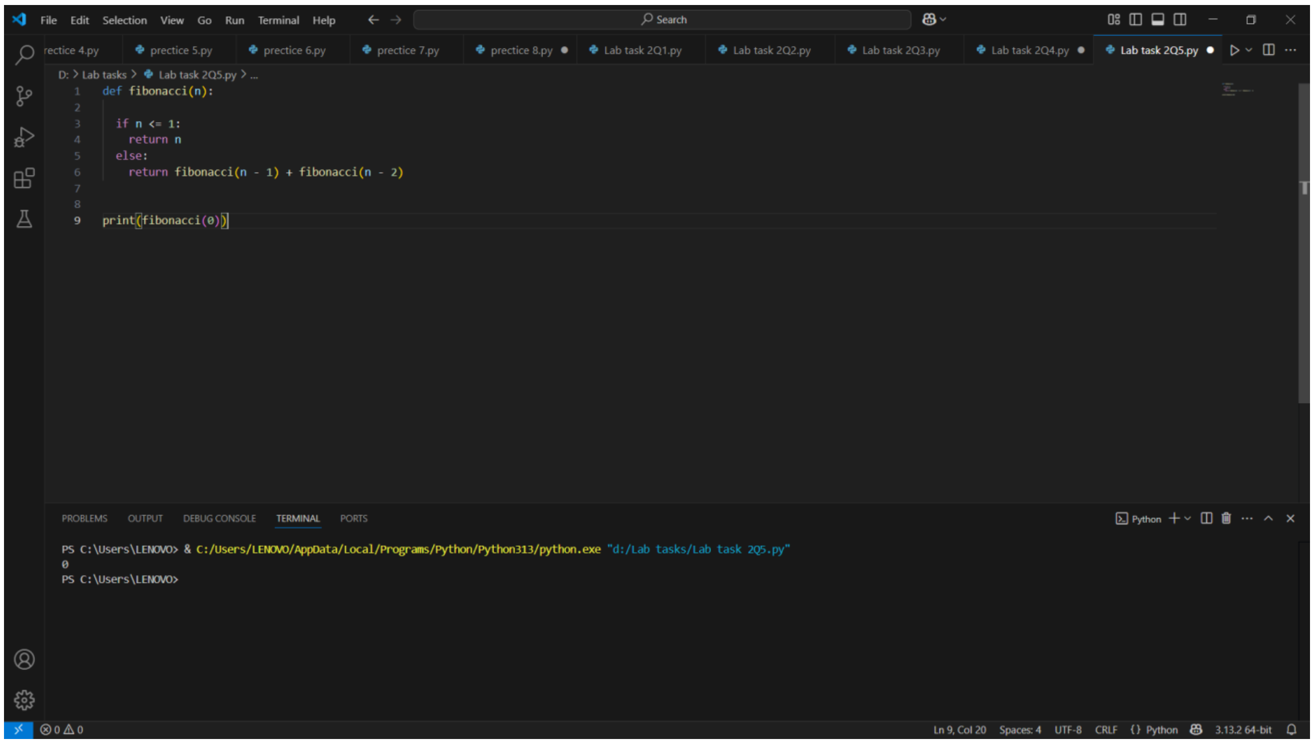
Part 5: Recursion

A function can call itself, known as recursion.

Question No. 5

Write a recursive function “fibonacci(n)” that returns the nth Fibonacci number.

Code :



```python
def fibonacci(n):

  if n <= 1:
    return n
  else:
    return fibonacci(n - 1) + fibonacci(n - 2)


print(fibonacci(0))
```

```
PS C:\Users\LENOVO> & C:/Users/LENOVO/AppData/Local/Programs/Python/Python313/python.exe "d:/Lab tasks/Lab task 2Q5.py"
0
PS C:\Users\LENOVO>
```

Part 6: Lambda Functions

Lambda functions are anonymous functions in Python.

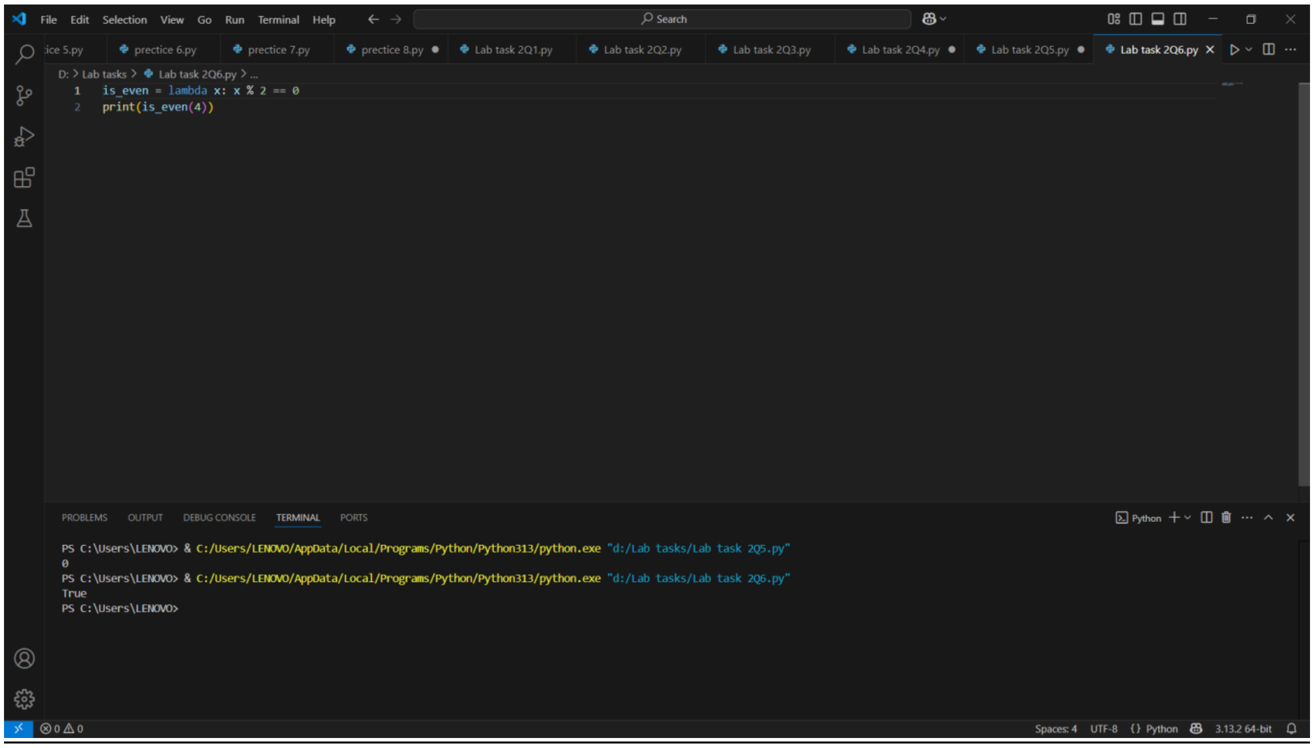
square = lambda x: x * x

print(square(5))

Question No. 6

Write a lambda function to check if a number is even.

Code :



```python
is_even = lambda x: x % 2 == 0
print(is_even(4))
```

```
PS C:\Users\LENOVO> & C:/Users/LENOVO/AppData/Local/Programs/Python/Python313/python.exe "d:/Lab tasks/Lab task 2Q5.py"
0
PS C:\Users\LENOVO> & C:/Users/LENOVO/AppData/Local/Programs/Python/Python313/python.exe "d:/Lab tasks/Lab task 2Q6.py"
True
PS C:\Users\LENOVO>
```

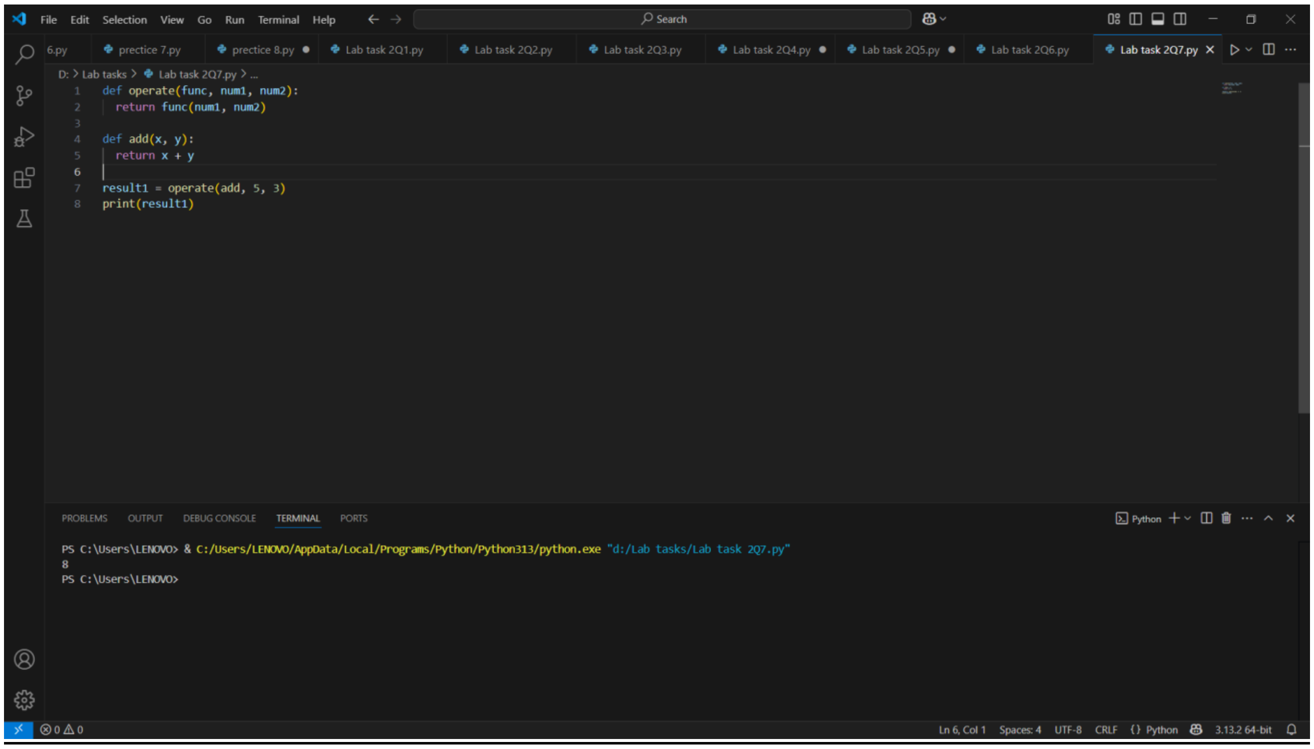
Part 7: Function as Arguments

Functions can be passed as arguments to other functions.

Question No. 7

Write a function operate that takes another function as an argument and applies it to two numbers.

Code :



```python
def operate(func, num1, num2):
    return func(num1, num2)

def add(x, y):
    return x + y

result1 = operate(add, 5, 3)
print(result1)
```

```
PS C:\Users\LENOVO> & C:/Users/LENOVO/AppData/Local/Programs/Python/Python313/python.exe "d:/Lab tasks/Lab task 2Q7.py"
8
PS C:\Users\LENOVO>
```

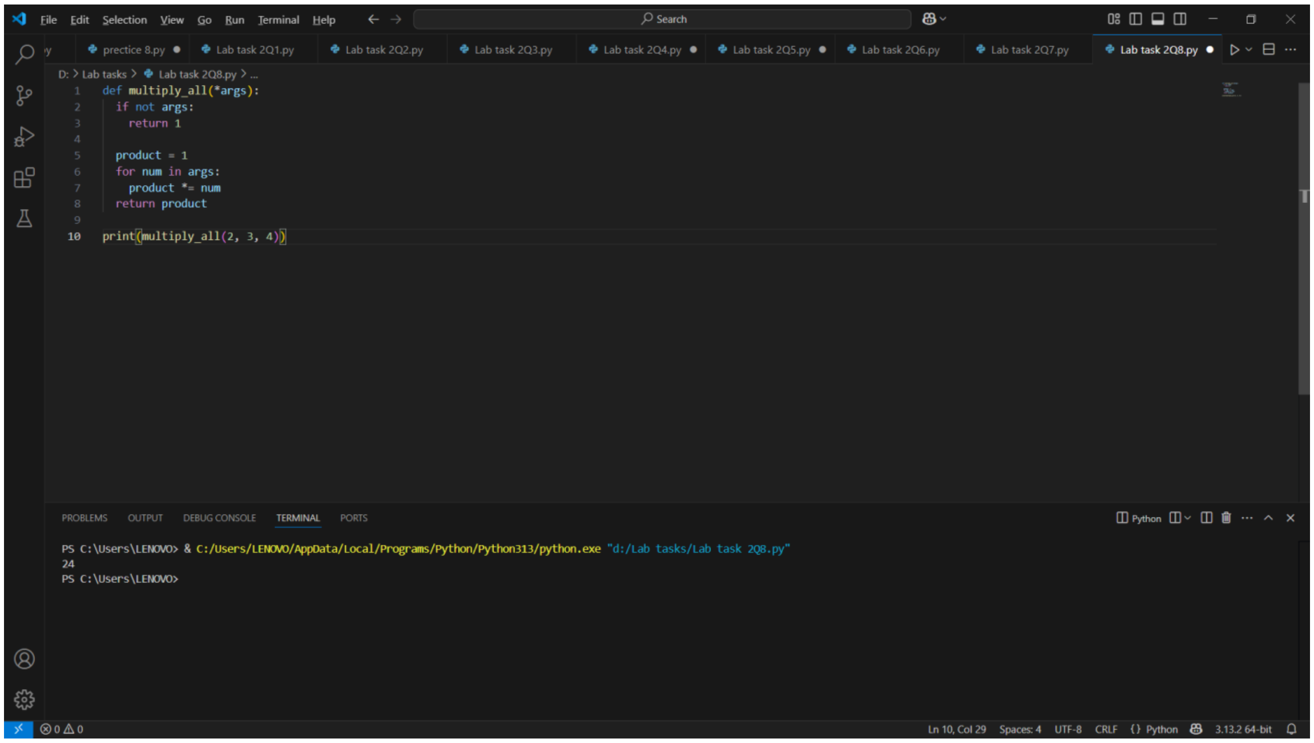
Part 8: *args and **kwargs :

Using *args for Variable-Length Arguments.

Question No. 8

Write a function multiply_all that accepts multiple arguments using *args and returns their product.

Code :



```python
def multiply_all(*args):
    if not args:
        return 1

    product = 1
    for num in args:
        product *= num
    return product

print(multiply_all(2, 3, 4))
```

```
PS C:\Users\LENOVO> & C:/Users/LENOVO/AppData/Local/Programs/Python/Python313/python.exe "d:/Lab tasks/Lab task 2Q8.py"
24
PS C:\Users\LENOVO>
```

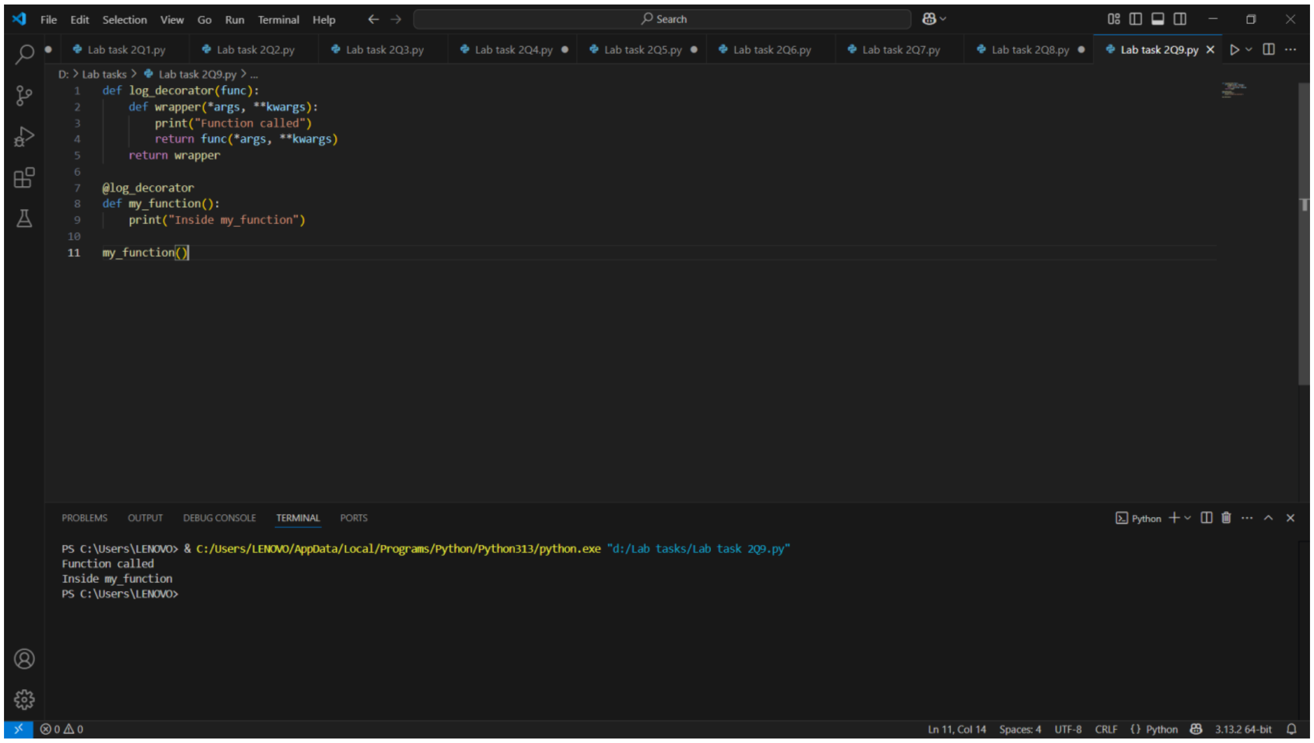
Part 9: Function Decorators

Decorators modify the behavior of a function

Question No. 9

Write a decorator log_decorator that prints "Function called" before executing the function.

Code : “””took help from ai.”””



```python
def log_decorator(func):
    def wrapper(*args, **kwargs):
        print("Function called")
        return func(*args, **kwargs)
    return wrapper

@log_decorator
def my_function():
    print("Inside my_function")

my_function()
```

```
PS C:\Users\LENOVO> & C:/Users/LENOVO/AppData/Local/Programs/Python/Python313/python.exe "d:/Lab tasks/Lab task 2Q9.py"
Function called
Inside my_function
PS C:\Users\LENOVO>
```