Name: Musfira Ahmed

Intern ID: TN/IN01/PY/002

Email ID : ahmedmusfira3@gmail.com

Internship Domain : Python Development

Task Week : 05

Instructor Name : Mr.Hassan Ali

Task 1 :

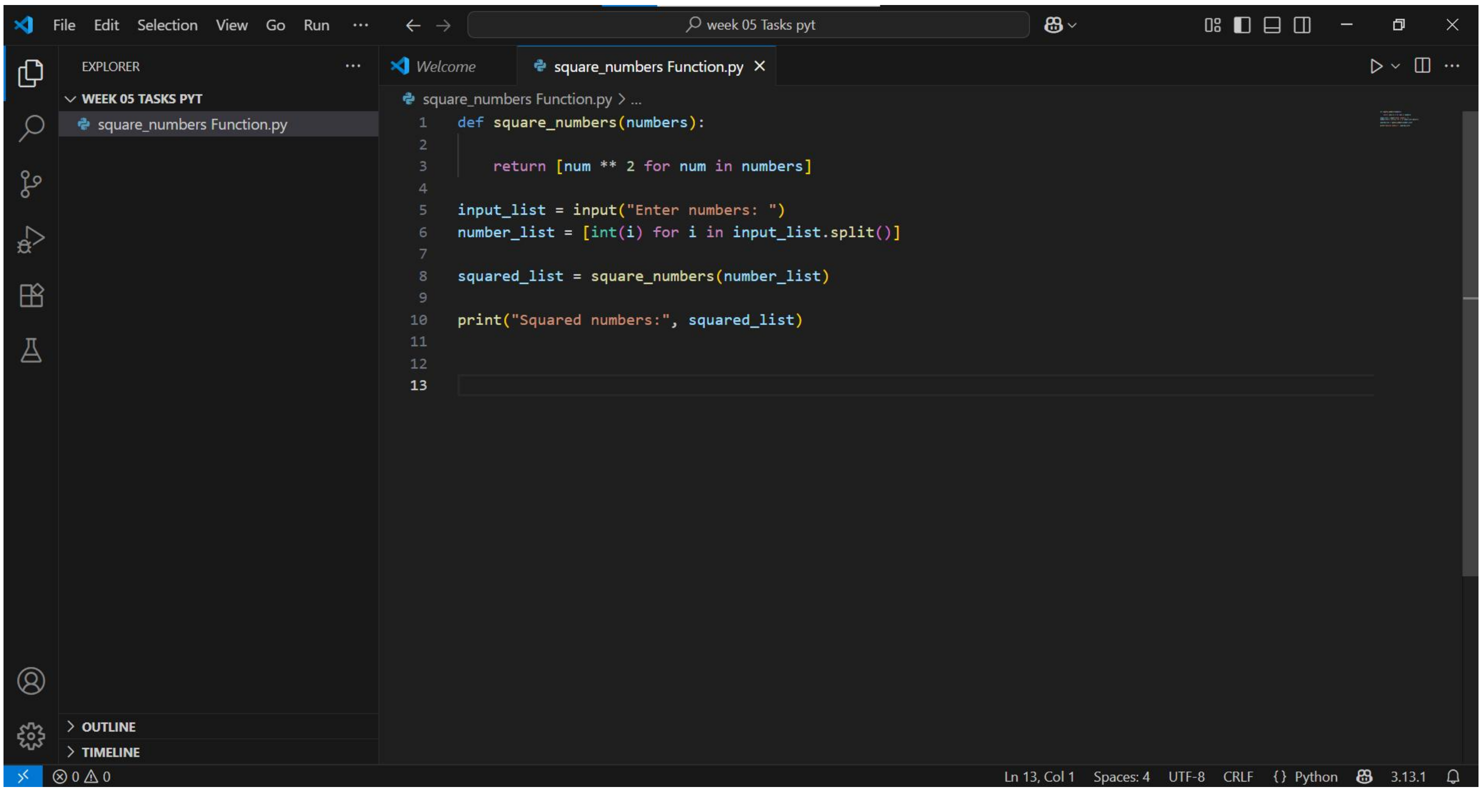
Create a function `square_numbers` that takes a list of numbers and returns a list of their squares.

Solution :

What I Did (Step by Step):

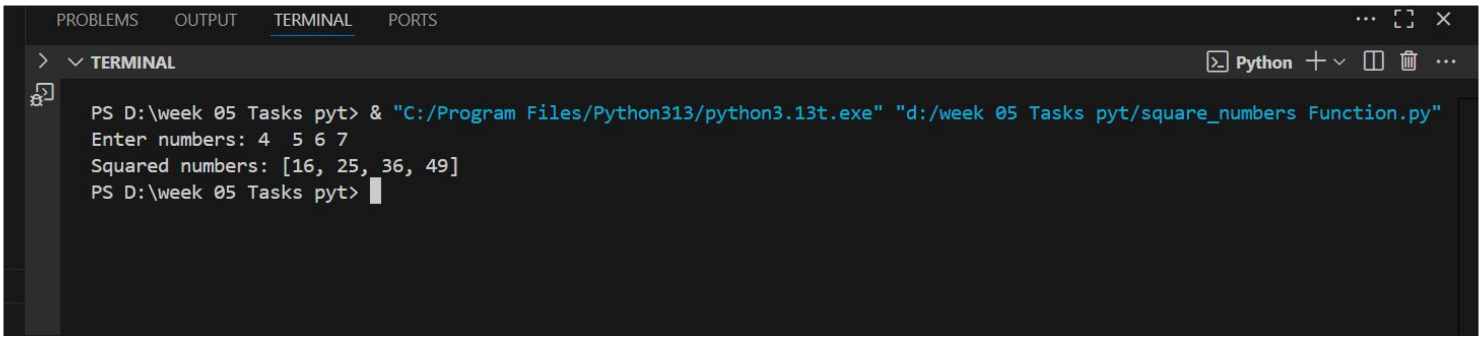
- Defined a function called square_numbers(numbers).
- Took space-separated input from the user and converted it to a list.
- Called the function with the user's list as input.
- Stored the result in a variable squared_list.
- Printed the squared numbers clearly.

Code Screenshot


File Edit Selection View Go Run
week 05 Tasks pyt
EXPLORER
WEEK 05 TASKS PYT
square_numbers Function.py
Welcome
OUTLINE
TIMELINE
Ln 13, Col 1 Spaces: 4 UTF-8 CRLF {} Python 3.13.1

```python
def square_numbers(numbers):

    return [num ** 2 for num in numbers]

input_list = input("Enter numbers: ")
number_list = [int(i) for i in input_list.split()]

squared_list = square_numbers(number_list)

print("Squared numbers:", squared_list)
```

Output Screenshot



```
PS D:\week 05 Tasks pyt> & "C:/Program Files/Python313/python3.13t.exe" "d:/week 05 Tasks pyt/square_numbers Function.py"
Enter numbers: 4  5 6 7
Squared numbers: [16, 25, 36, 49]
PS D:\week 05 Tasks pyt>
```

Learning and Challenges:

1. Learned how to write reusable functions in Python.
2. Practiced converting string input into a list of integers.
3. Faced a small challenge with split() and int() conversion.
4. Fixed it using list comprehension to handle all numbers easily.
5. Improved understanding of how to work with lists and functions together.

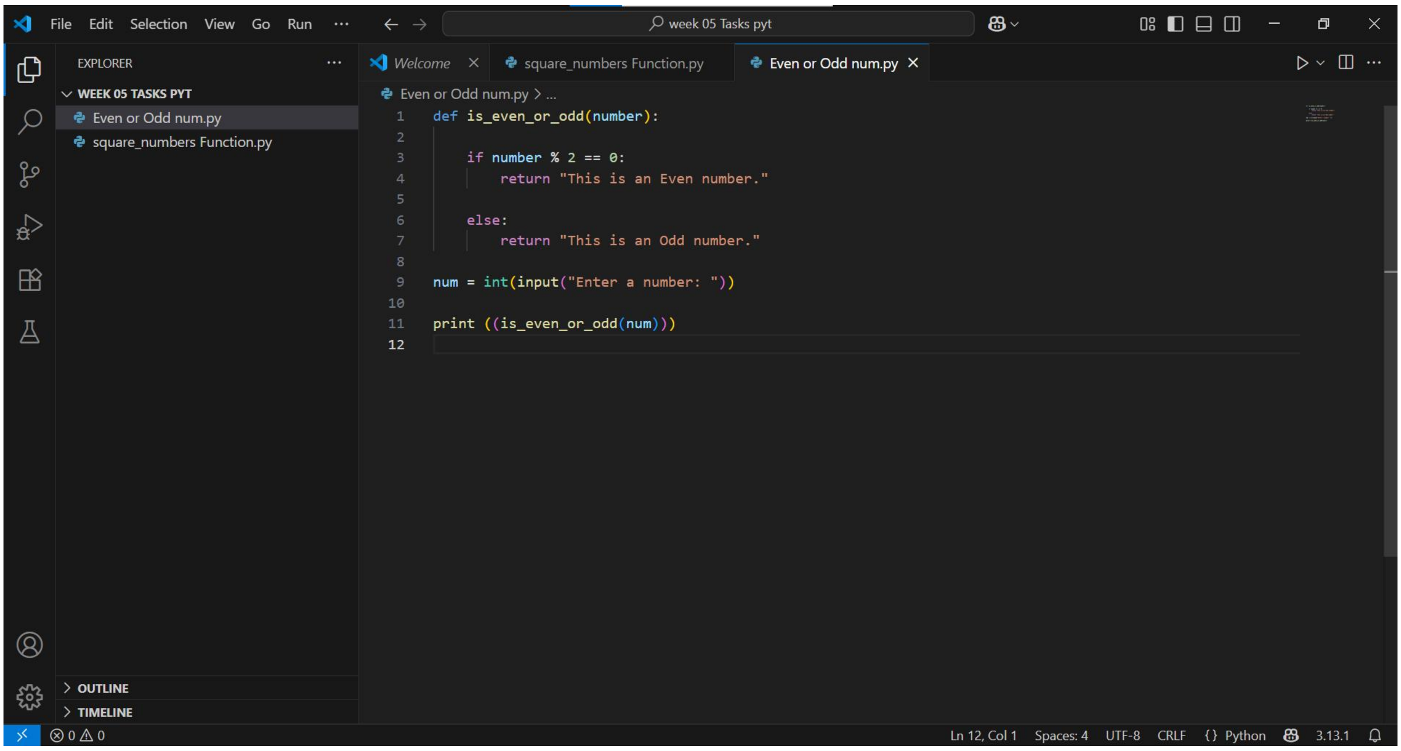
Task 02:

Create a function `is_even_or_odd` that takes a number and returns whether it is even or odd.

What I Did (Step by Step):

- Created a function called is_even_or_odd(number).
- Used the modulus % operator to check if the number is divisible by 2.
- Returned "Even" or "Odd" based on the result.
- Took input from the user using input().
- Printed whether the number is even or odd.

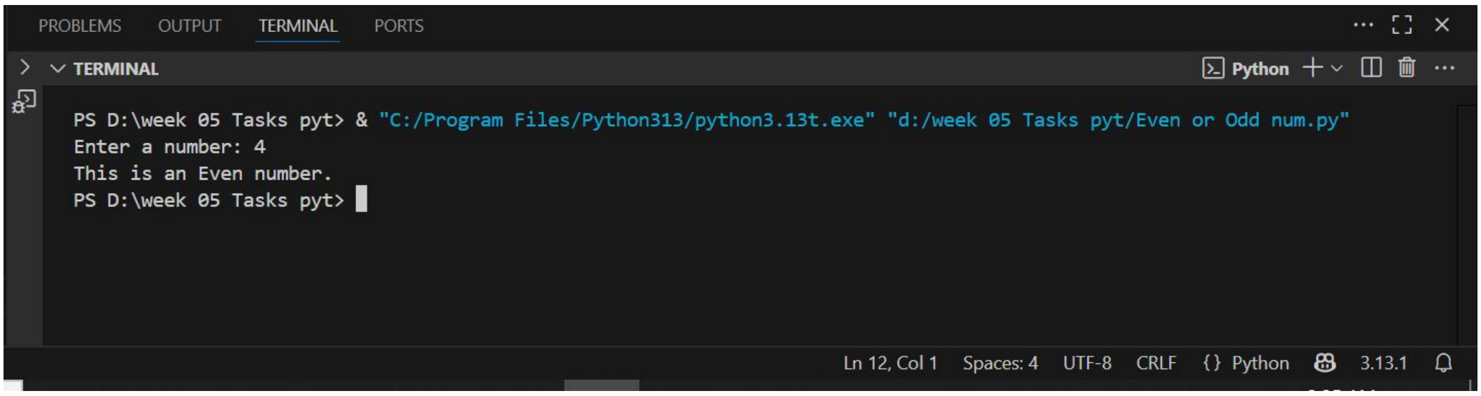
Code Screenshots



File Edit Selection View Go Run
week 05 Tasks pyt
EXPLORER
WEEK 05 TASKS PYT
Even or Odd num.py
square_numbers Function.py
Welcome
OUTLINE
TIMELINE
Ln 12, Col 1 Spaces: 4 UTF-8 CRLF {} Python 3.13.1

```python
def is_even_or_odd(number):

    if number % 2 == 0:
        return "This is an Even number."

    else:
        return "This is an Odd number."

num = int(input("Enter a number: "))

print ((is_even_or_odd(num)))

```

Output Screenshot



PROBLEMS OUTPUT TERMINAL PORTS
TERMINAL
Python
Ln 12, Col 1 Spaces: 4 UTF-8 CRLF {} Python 3.13.1

```
PS D:\week 05 Tasks pyt> & "C:/Program Files/Python313/python3.13t.exe" "d:/week 05 Tasks pyt/Even or Odd num.py"
Enter a number: 4
This is an Even number.
PS D:\week 05 Tasks pyt>
```

Learning and Challenges:

1) Learned how to check even/odd using %.
2) Practiced writing and calling a function with one input.
3) Faced no issues, logic was easy and clear.
4) Understood how return values work inside a function.
5) Improved my conditional thinking and function writing skills.

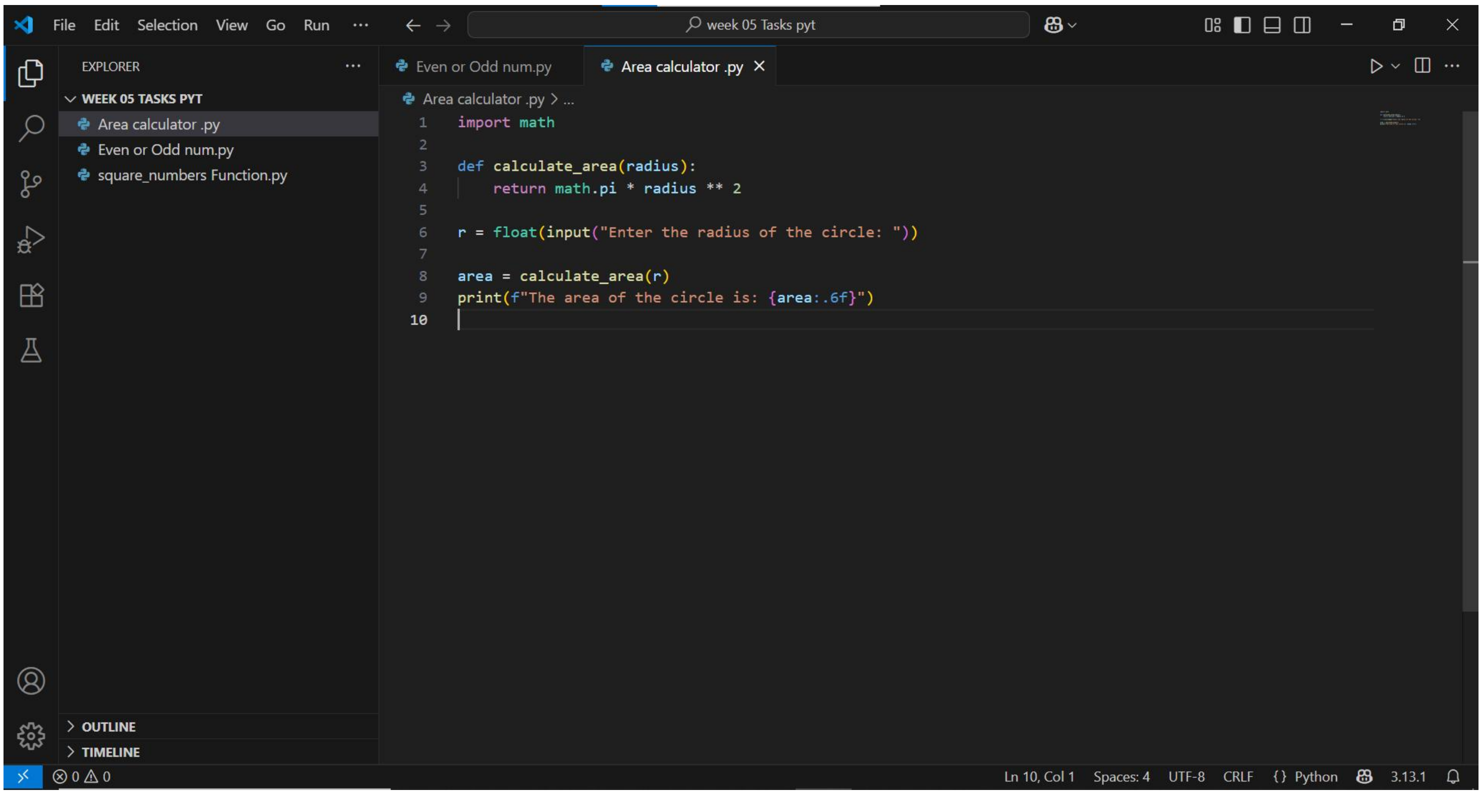
<u>Task 03:</u>

Write a function `calculate_area` that takes radius and returns area of a circle.

What I Did (Step by Step):

- Imported the math module to use math.pi.
- Created a function called calculate_area(radius).
- Used the formula π × $radius^2$ to calculate area.
- Took radius input from the user.
- Called the function and printed the result.

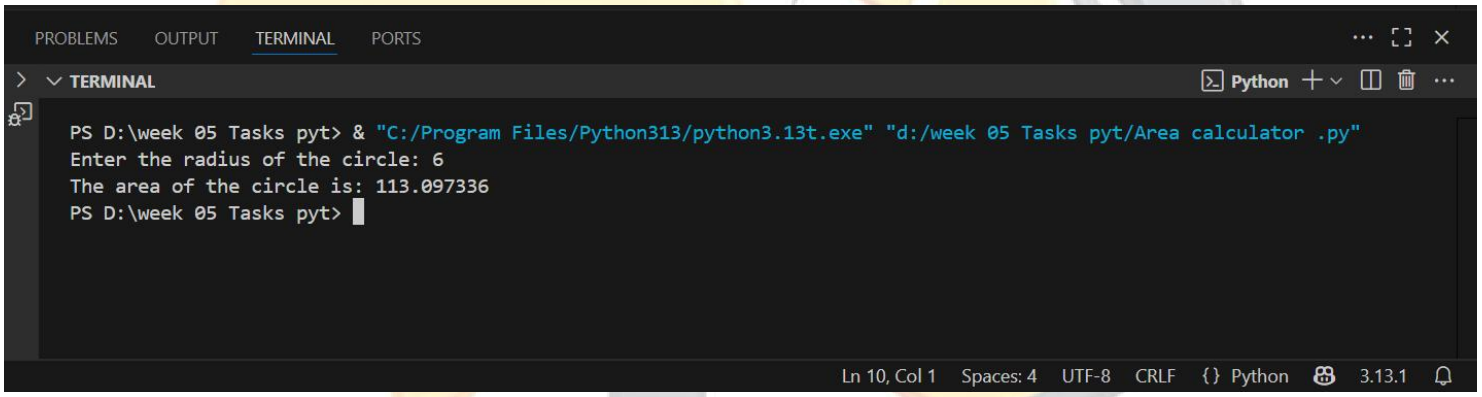
Code Screenshots



File Edit Selection View Go Run
week 05 Tasks pyt
EXPLORER
WEEK 05 TASKS PYT
Area calculator .py
Even or Odd num.py
square_numbers Function.py
OUTLINE
TIMELINE
Ln 10, Col 1 Spaces: 4 UTF-8 CRLF {} Python 3.13.1

```python
import math

def calculate_area(radius):
    return math.pi * radius ** 2

r = float(input("Enter the radius of the circle: "))

area = calculate_area(r)
print(f"The area of the circle is: {area:.6f}")

```

Output Screenshot



PROBLEMS OUTPUT TERMINAL PORTS
TERMINAL
Python
Ln 10, Col 1 Spaces: 4 UTF-8 CRLF {} Python 3.13.1

```
PS D:\week 05 Tasks pyt> & "C:/Program Files/Python313/python3.13t.exe" "d:/week 05 Tasks pyt/Area calculator .py"
Enter the radius of the circle: 6
The area of the circle is: 113.097336
PS D:\week 05 Tasks pyt> 
```

Learnings and Challenges:

1) Learned how to use the math module in Python.
2) Practiced writing functions with mathematical formulas.
3) Faced no issues, just used correct input and float type.
4) Understood how return values work in real-world formulas.
5) Improved confidence in writing math-based functions.

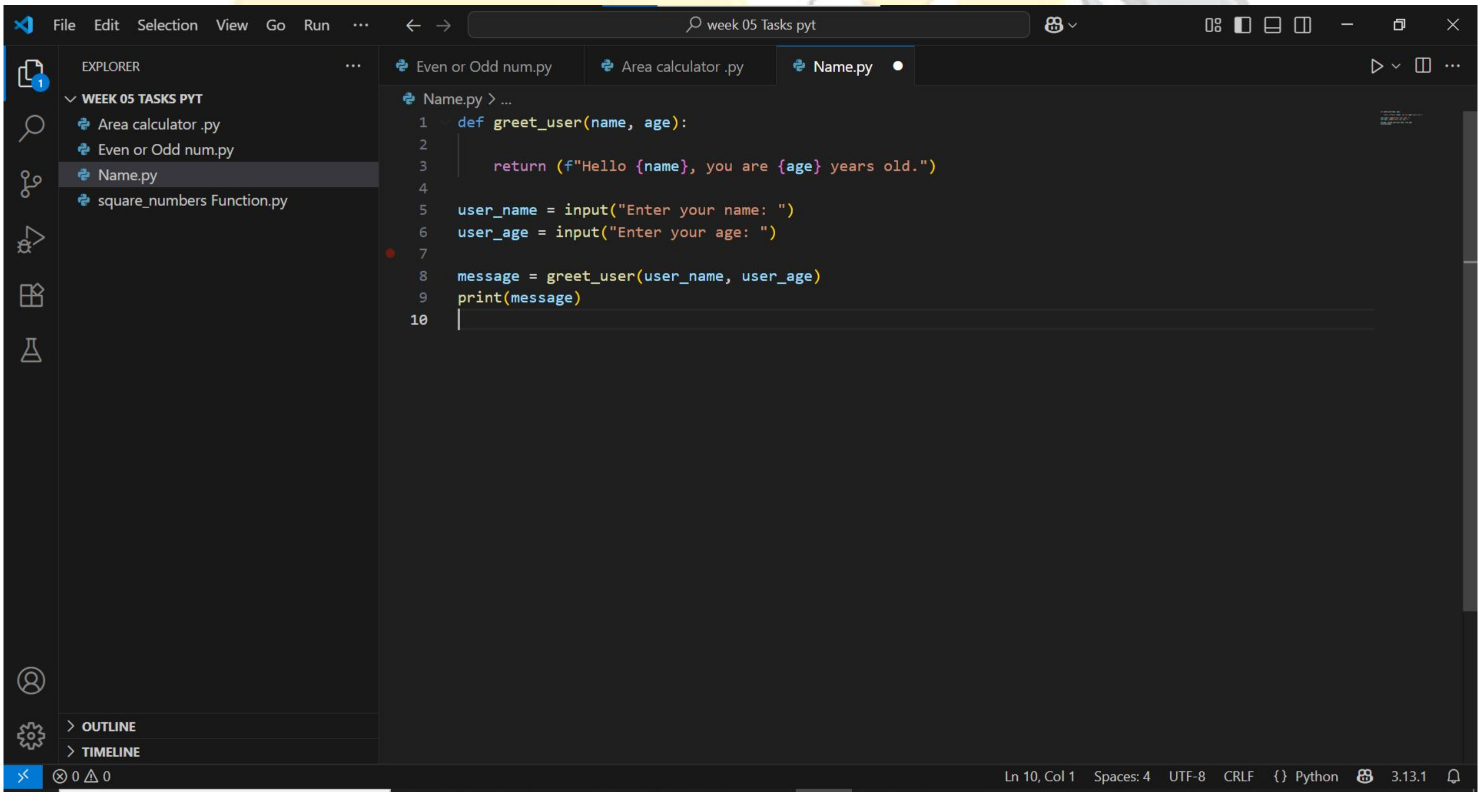
Task 04:

Write a function `greet_user(name, age)` that returns a greeting like: 'Hello Ali, you are 20 years old.'

What I Did (Step by Step):

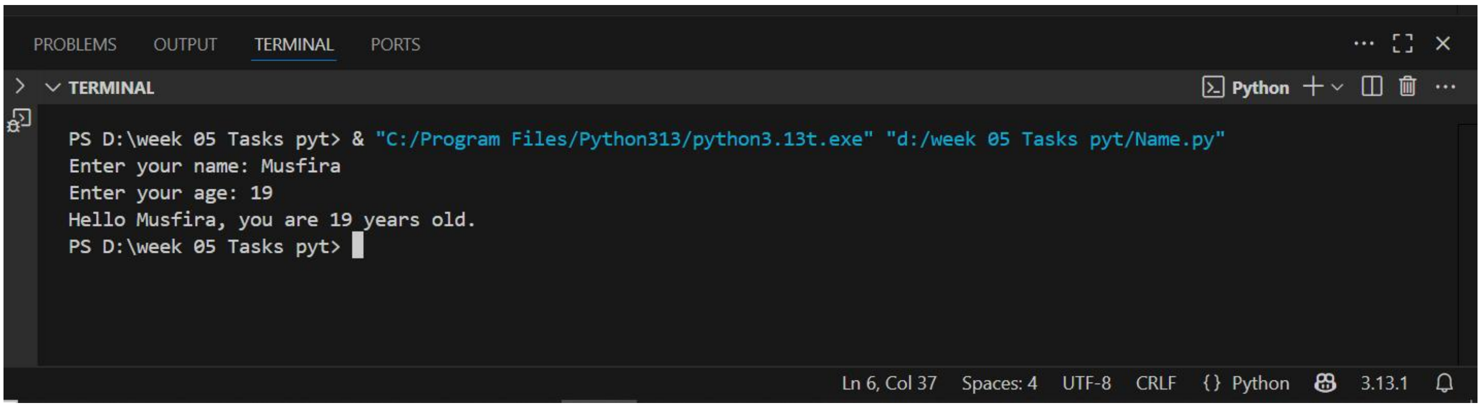
- Defined a function greet_user(name, age) with two parameters.
- Used an f-string to format the greeting message.
- Took name and age as input from the user.
- Called the function with those inputs.
- Printed the personalized message.

Code Screenshots


File Edit Selection View Go Run
week 05 Tasks pyt
EXPLORER
WEEK 05 TASKS PYT
Area calculator .py
Even or Odd num.py
Name.py
square_numbers Function.py
OUTLINE
TIMELINE
Ln 10, Col 1 Spaces: 4 UTF-8 CRLF {} Python 3.13.1

```python
def greet_user(name, age):

    return (f"Hello {name}, you are {age} years old.")

user_name = input("Enter your name: ")
user_age = input("Enter your age: ")

message = greet_user(user_name, user_age)
print(message)

```

Output Screenshots


PROBLEMS OUTPUT TERMINAL PORTS
TERMINAL
Python
Ln 6, Col 37 Spaces: 4 UTF-8 CRLF {} Python 3.13.1

```
PS D:\week 05 Tasks pyt> & "C:/Program Files/Python313/python3.13t.exe" "d:/week 05 Tasks pyt/Name.py"
Enter your name: Musfira
Enter your age: 19
Hello Musfira, you are 19 years old.
PS D:\week 05 Tasks pyt>
```

Learning and Challenges

1. Practiced using f-strings for clean formatting.
2. Faced no issues logic was clear and direct.
3. Understood how to return and use strings from a function.
4. Improved basic input/output handling and formatting.
5. Learned how to use multiple arguments in a function.

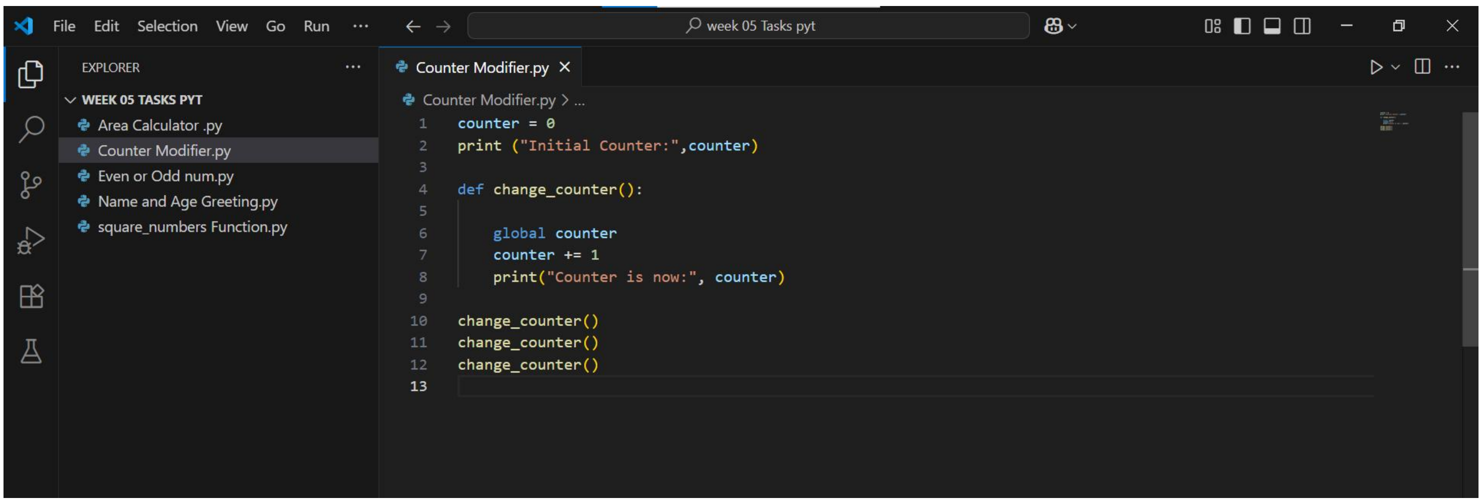
<u>Task 05:</u>

Create a function `change_counter()` that modifies a global counter variable.

What I Did (Step by Step):

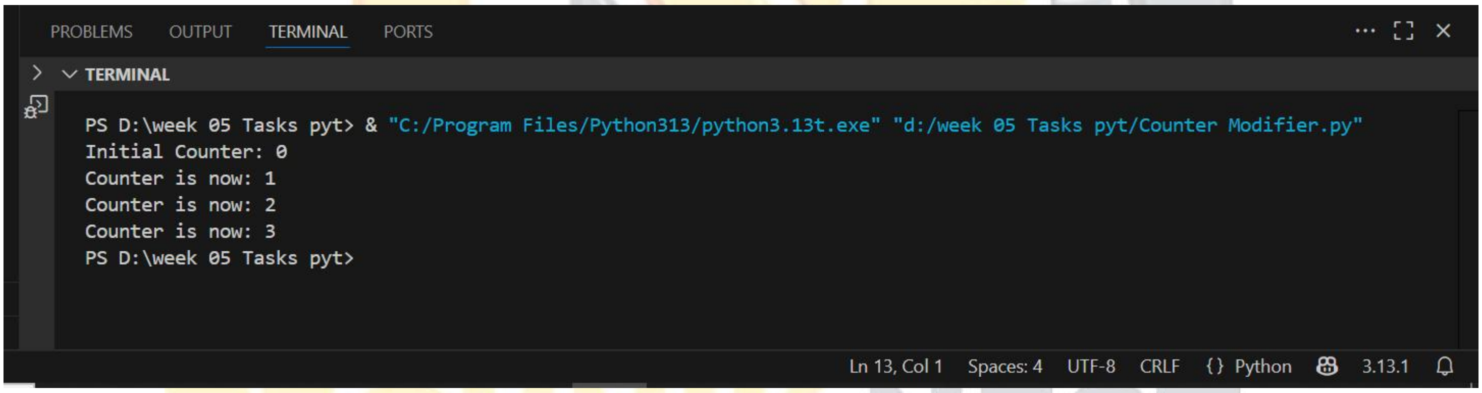
- Declared a global variable counter with initial value 0.
- Created a function change_counter() that uses global keyword.
- Increased the value of counter by 1 inside the function.
- Called the function multiple times to show the change.
- Printed the updated counter value after each call.

Code Screenshots



```
counter = 0
print ("Initial Counter:",counter)

def change_counter():

    global counter
    counter += 1
    print("Counter is now:", counter)

change_counter()
change_counter()
change_counter()
```

Output Screenshots



```
PS D:\week 05 Tasks pyt> & "C:/Program Files/Python313/python3.13t.exe" "d:/week 05 Tasks pyt/Counter Modifier.py"
Initial Counter: 0
Counter is now: 1
Counter is now: 2
Counter is now: 3
PS D:\week 05 Tasks pyt>
```

PROBLEMS OUTPUT TERMINAL PORTS
TERMINAL
Ln 13, Col 1 Spaces: 4 UTF-8 CRLF {} Python 3.13.1

Learning and Challenges

1. Learned how to use and modify global variables in a function.
2. Practiced using the global keyword correctly.
3. Faced no issues syntax is simple and clear.
4. Understood the difference between local and global scope.
5. Gained better control over how variables behave across code.

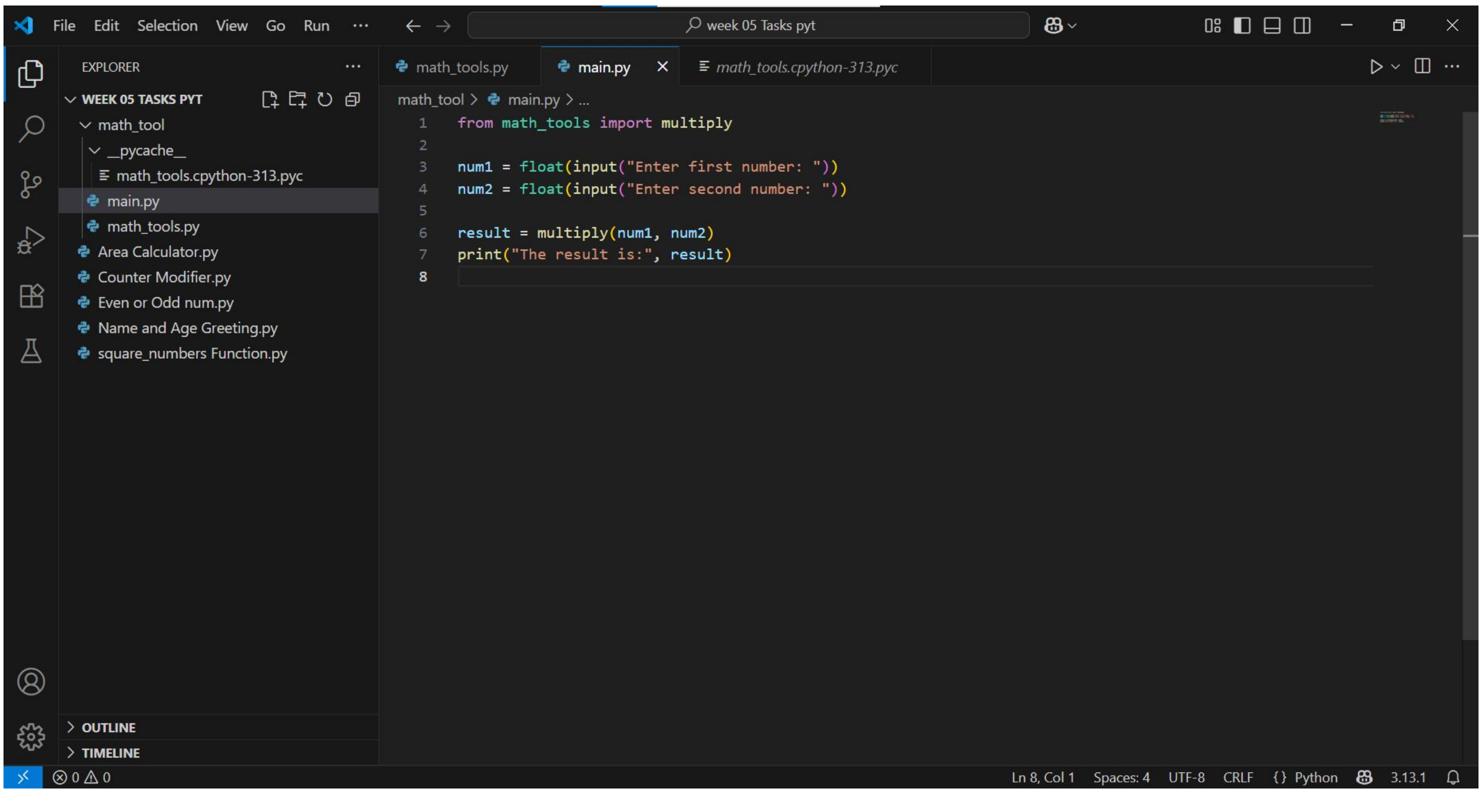
Task 06:

Create a module named `math_tools.py` with a function `multiply(x, y)` and use it in another script.

What I Did (Step by Step):

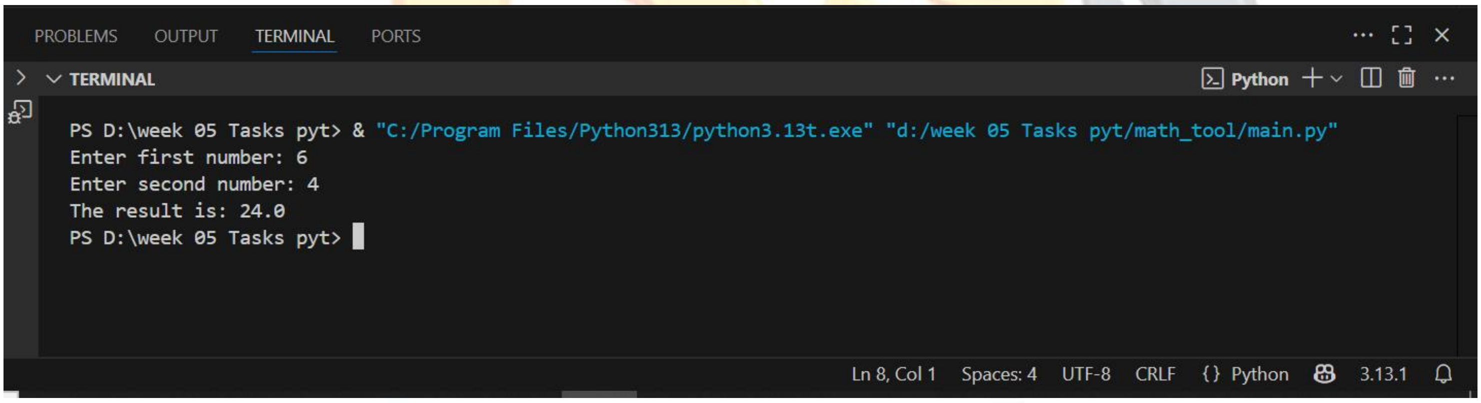
- Created a module file math_tools.py with a function multiply(x, y).
- Wrote a second file main.py to import and use that function.
- Took two numbers from user input using input().
- Called the function and stored the result.
- Printed the final result clearly.

Code Screenshots



```python
from math_tools import multiply

num1 = float(input("Enter first number: "))
num2 = float(input("Enter second number: "))

result = multiply(num1, num2)
print("The result is:", result)
```

Output Screenshots



```
PS D:\week 05 Tasks pyt> & "C:/Program Files/Python313/python3.13t.exe" "d:/week 05 Tasks pyt/math_tool/main.py"
Enter first number: 6
Enter second number: 4
The result is: 24.0
PS D:\week 05 Tasks pyt>
```

Ln 8, Col 1   Spaces: 4   UTF-8   CRLF   {} Python   3.13.1

Learning and Challenges

1. Learned how to build and import custom modules in Python.
2. Practiced separating code into reusable files.
3. Faced an import error when files weren’t in the same folder.
4. Solved it by keeping both files together.
5. Understood the use of modular code for cleaner projects.