SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE			**DEPARTMENT OF COMPUTER SCIENCE ENGINEERING**
Program Name: B. Tech		**Assignment Type: Lab**	**Academic Year:**2025-2026
CourseCode	23CS002PC304	**Course Title**	AI Assisted Coding
Year/Sem	III/II	**Regulation**	R23
Date and Day of Assignment	**Week1 – Monday**	**Batch**	23CSBTB47B
Name	Yashaswini Mittapalli	**Hall Ticket No**	2303A54049
Assignment Number:3.1			

QUESTION 1: Zero-Shot Prompting (Palindrome Number Program)

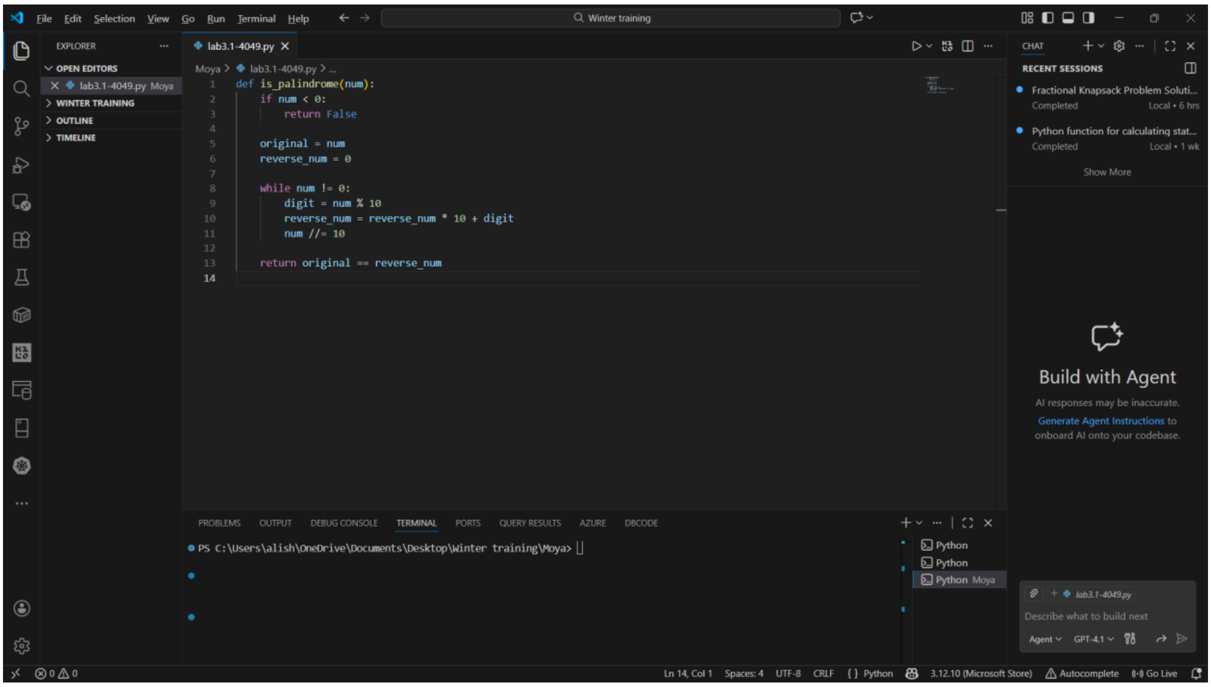
Problem Statement

Create a zero-shot prompt to generate a Python function that determines whether a given number is a palindrome.

Prompt

Write a Python function to check whether an integer is a palindrome using only mathematical operations (without converting the number into a string). The function should return True if the number is a palindrome; otherwise, return False.

Generated Code Screenshot


File Edit Selection View Go Run Terminal Help
Winter training
EXPLORER
OPEN EDITORS
lab3.1-4049.py Moya
WINTER TRAINING
OUTLINE
TIMELINE
lab3.1-4049.py
Moya > lab3.1-4049.py > ...
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULTS AZURE DBCODE
Python
Python
Python Moya
CHAT
RECENT SESSIONS
Fractional Knapsack Problem Soluti...
Completed Local • 6 hrs
Python function for calculating stat...
Completed Local • 1 wk
Show More
Build with Agent
AI responses may be inaccurate.
Generate Agent Instructions to onboard AI onto your codebase.
lab3.1-4049.py
Describe what to build next
Agent GPT-4.1
Ln 14, Col 1 Spaces: 4 UTF-8 CRLF {} Python 3.12.10 (Microsoft Store) Autocomplete Go Live

```python
def is_palindrome(num):
    if num < 0:
        return False

    original = num
    reverse_num = 0

    while num != 0:
        digit = num % 10
        reverse_num = reverse_num * 10 + digit
        num //= 10

    return original == reverse_num
```

```
PS C:\Users\alish\OneDrive\Documents\Desktop\Winter training\Moya>
```

Input and Output

Input	Output	Reason
121	1	The reversed value matches the original
-121	0	Negative numbers are not considered palindromes.
10	0	The reversed number (1) does not match input.
12321	1	Reversing the digits gives the same num

Explanation

The function immediately rejects negative values. It then stores the original number and constructs its reverse by extracting digits one by one using arithmetic operations. After reversing, the function compares the result with the original value to determine whether it is a palindrome.

QUESTION 2: One-Shot Prompting (Factorial Computation)

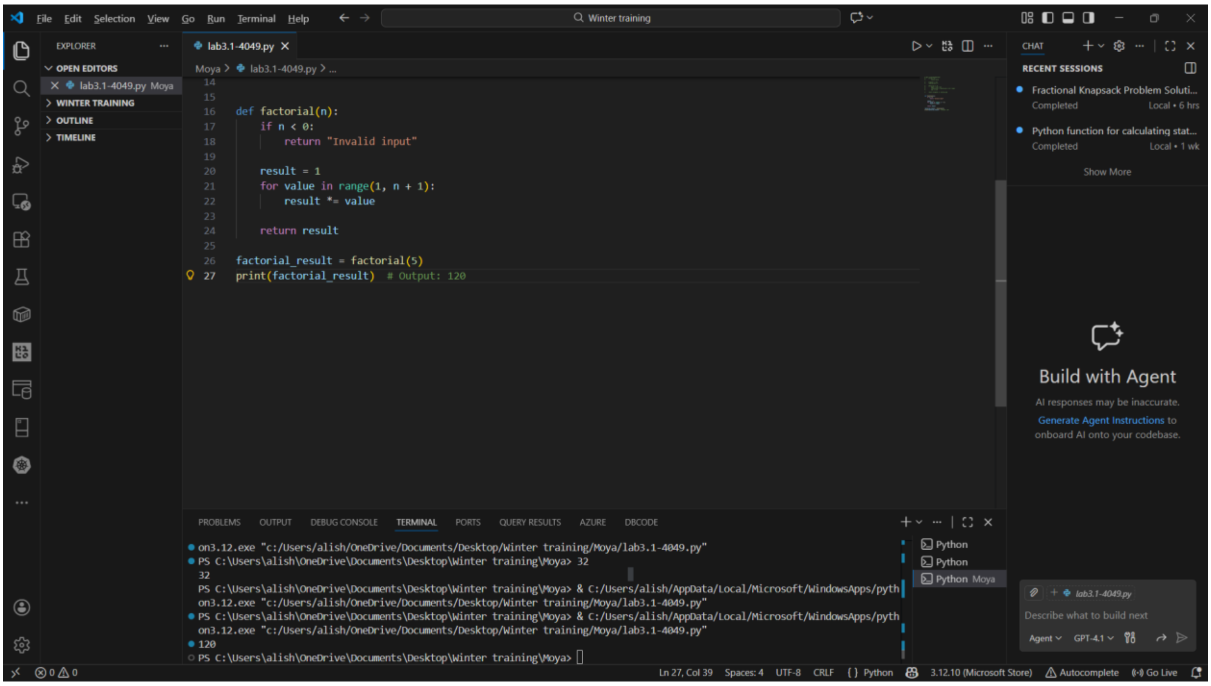
Problem Statement

Provide a one-shot prompt with a single example to generate a Python function that calculates factorial values.

Prompt

Write a Python function that calculates the factorial of a given integer. The function should correctly compute factorials for non-negative values and handle invalid inputs such as negative numbers.

Generated Code Screenshot



```python


def factorial(n):
    if n < 0:
        return "Invalid input"

    result = 1
    for value in range(1, n + 1):
        result *= value

    return result

factorial_result = factorial(5)
print(factorial_result)  # Output: 120
```

```
on3.12.exe "c:/Users/alish/OneDrive/Documents/Desktop/Winter training/Moya/lab3.1-4049.py"
PS C:\Users\alish\OneDrive\Documents\Desktop\Winter training\Moya> 32
32
PS C:\Users\alish\OneDrive\Documents\Desktop\Winter training\Moya> & C:/Users/alish/AppData/Local/Microsoft/WindowsApps/pyth
on3.12.exe "c:/Users/alish/OneDrive/Documents/Desktop/Winter training/Moya/lab3.1-4049.py"
PS C:\Users\alish\OneDrive\Documents\Desktop\Winter training\Moya> & C:/Users/alish/AppData/Local/Microsoft/WindowsApps/pyth
on3.12.exe "c:/Users/alish/OneDrive/Documents/Desktop/Winter training/Moya/lab3.1-4049.py"
120
PS C:\Users\alish\OneDrive\Documents\Desktop\Winter training\Moya> 
```

Input and Output

Input	Output	Reason
5	120	Factorial is the product of all integers from 1 to
0	1	By definition, factorial of 0 is 1.

-3	Invalid input	Factorial is undefined for negative numbers.
7	5040	Computed as 7 × 6 × 5 × 4 × 3 × 2 × 1.

Explanation

Factorial represents the multiplication of all positive integers up to a given number. The function accounts for the special case of 0 and properly handles invalid negative inputs.

QUESTION 3: Few-Shot Prompting (Armstrong Number Check)

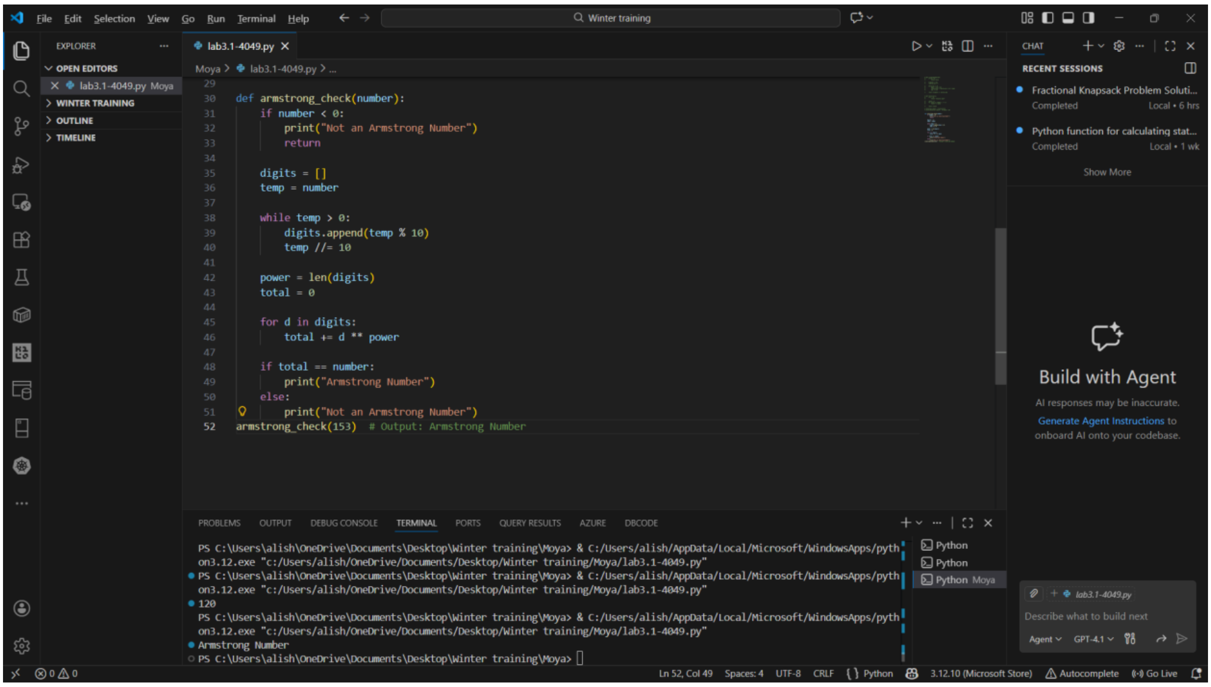
Problem Statement

Design a few-shot prompt using multiple examples to guide the AI in identifying Armstrong numbers.

Prompt

Write a Python function that verifies whether a given integer is an Armstrong number. The program should display “Armstrong Number” if the condition is satisfied; otherwise, display “Not an Armstrong Number”.

Generated Code Screenshot



```python

def armstrong_check(number):
    if number < 0:
        print("Not an Armstrong Number")
        return

    digits = []
    temp = number

    while temp > 0:
        digits.append(temp % 10)
        temp //= 10

    power = len(digits)
    total = 0

    for d in digits:
        total += d ** power

    if total == number:
        print("Armstrong Number")
    else:
        print("Not an Armstrong Number")
armstrong_check(153)  # Output: Armstrong Number
```

```
PS C:\Users\alish\OneDrive\Documents\Desktop\Winter training\Moya> & C:/Users/alish/AppData/Local/Microsoft/WindowsApps/pyth
on3.12.exe "c:/Users/alish/OneDrive/Documents/Desktop/Winter training/Moya/lab3.1-4049.py"
PS C:\Users\alish\OneDrive\Documents\Desktop\Winter training\Moya> & C:/Users/alish/AppData/Local/Microsoft/WindowsApps/pyth
on3.12.exe "c:/Users/alish/OneDrive/Documents/Desktop/Winter training/Moya/lab3.1-4049.py"
120
PS C:\Users\alish\OneDrive\Documents\Desktop\Winter training\Moya> & C:/Users/alish/AppData/Local/Microsoft/WindowsApps/pyth
on3.12.exe "c:/Users/alish/OneDrive/Documents/Desktop/Winter training/Moya/lab3.1-4049.py"
Armstrong Number
PS C:\Users\alish\OneDrive\Documents\Desktop\Winter training\Moya> 
```

Input and Output

Input	Output	Reason
153	Armstrong Number	Sum of cubes of digits equals the number.
123	Not an Armstrong Number	Sum of cubes does not match the input.
9474	Armstrong Number	Sum of fourth powers of digits equals the number.
-5	Not an Armstrong Number	Negative numbers are excluded.
370	Armstrong Number	Digit powers sum to the original number.

Explanation

An Armstrong number is equal to the sum of its digits raised to the power of the total number of digits. The program calculates this sum and compares it with the original number to determine the result.

QUESTION 4: Context-Managed Prompting (Prime, Composite, or Neither)

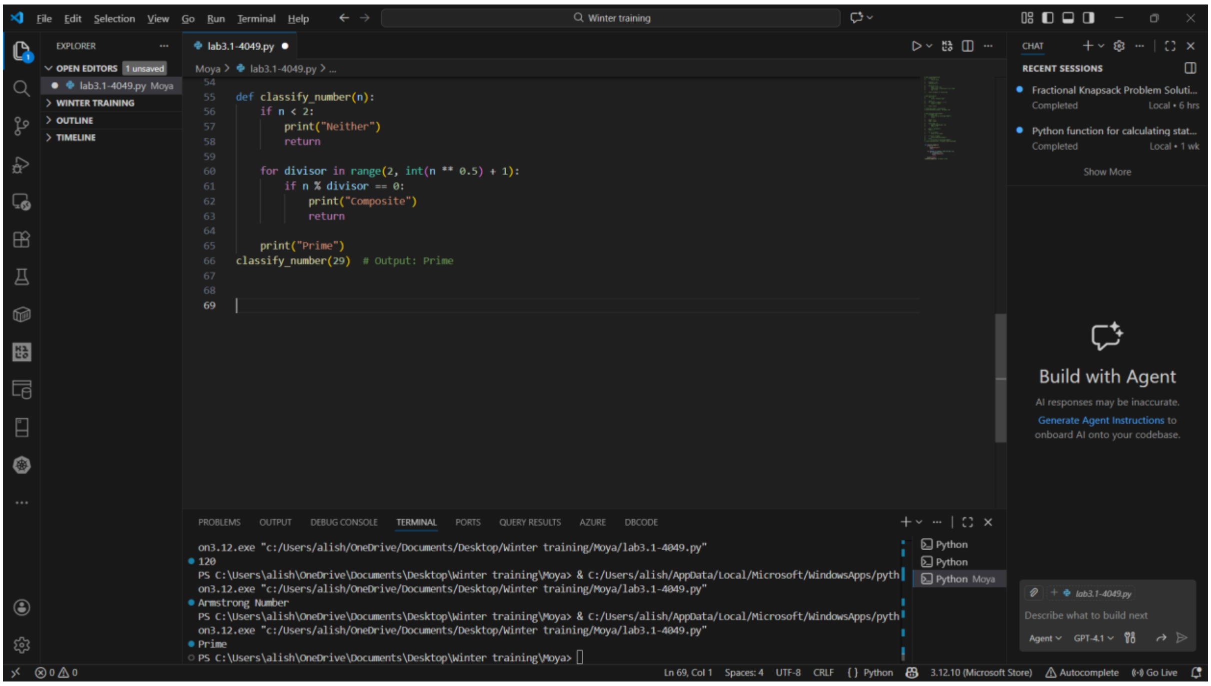
Problem Statement

Create a context-managed prompt with constraints to generate an optimized number classification program.

Prompt

Write an optimized Python program that classifies a given integer as Prime, Composite, or Neither. The program must validate inputs, check divisibility only up to the square root of the number, and produce a single output. Also, briefly compare this approach with a basic divisibility method.

Generated Code Screenshot



Winter training
lab3.1-4049.py
Moya > lab3.1-4049.py > ...
EXPLORER
OPEN EDITORS 1 unsaved
WINTER TRAINING
OUTLINE
TIMELINE
CHAT
RECENT SESSIONS
Fractional Knapsack Problem Soluti...
Python function for calculating stat...
Build with Agent
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULTS AZURE DBCODE
Ln 69, Col 1 Spaces: 4 UTF-8 CRLF { } Python

```python

def classify_number(n):
    if n < 2:
        print("Neither")
        return

    for divisor in range(2, int(n ** 0.5) + 1):
        if n % divisor == 0:
            print("Composite")
            return

    print("Prime")
classify_number(29)  # Output: Prime


```

```
on3.12.exe "c:/Users/alish/OneDrive/Documents/Desktop/Winter training/Moya/lab3.1-4049.py"
120
PS C:\Users\alish\OneDrive\Documents\Desktop\Winter training\Moya> & C:/Users/alish/AppData/Local/Microsoft/WindowsApps/pyth
on3.12.exe "c:/Users/alish/OneDrive/Documents/Desktop/Winter training/Moya/lab3.1-4049.py"
Armstrong Number
PS C:\Users\alish\OneDrive\Documents\Desktop\Winter training\Moya> & C:/Users/alish/AppData/Local/Microsoft/WindowsApps/pyth
on3.12.exe "c:/Users/alish/OneDrive/Documents/Desktop/Winter training/Moya/lab3.1-4049.py"
Prime
PS C:\Users\alish\OneDrive\Documents\Desktop\Winter training\Moya> 
```

Input and Output

Input	Output	Reason
11	Prime	No divisors found within √11.
15	Composite	Divisible by 3.
1	Neither	Numbers below 2 are neither prime nor composite.
-5	Neither	Negative numbers are invalid for classification.
2	Prime	Only divisible by 1 and itself.

Explanation

The program first handles invalid and boundary values. For valid numbers, it checks divisibility only up to √n, which significantly reduces computation compared to checking all values up to n.

QUESTION 5: Zero-Shot Prompting (Perfect Number Check)

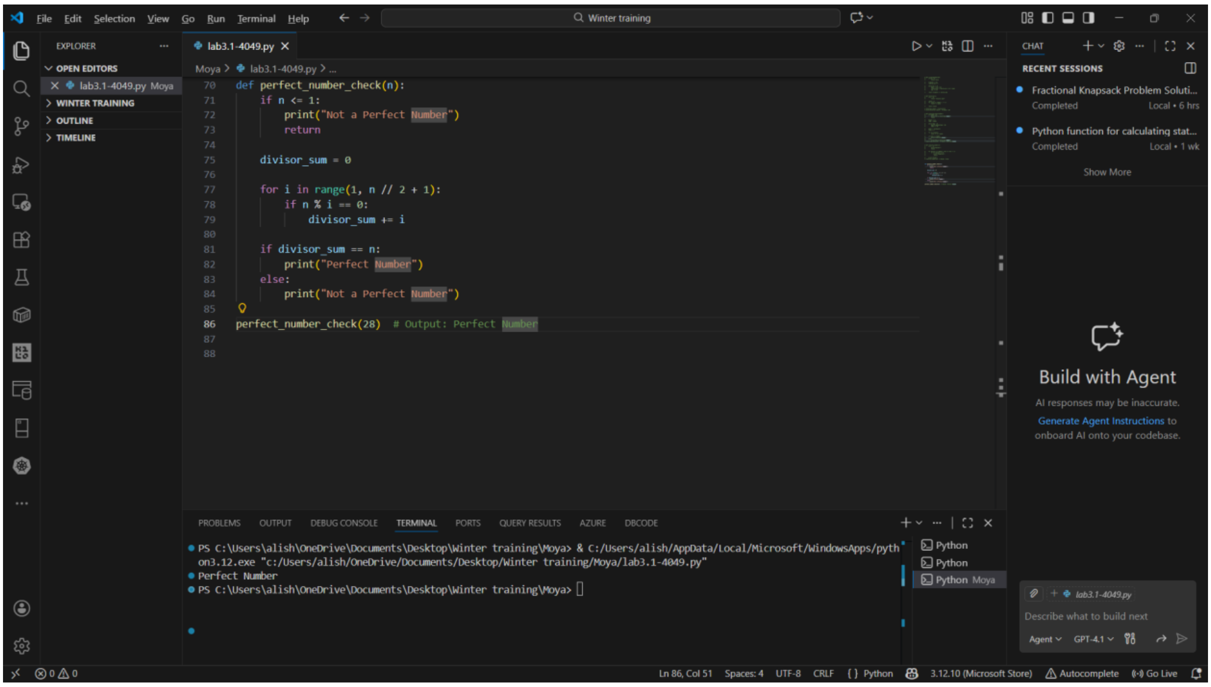
Problem Statement

Generate a zero-shot prompt to determine whether a number is a perfect number.

Prompt

Write a Python function that checks whether a given integer is a Perfect Number. The function should handle negative numbers, 0, and 1 correctly, use an efficient divisor-checking method, and print appropriate results.

Generated Code Screenshot



```python
def perfect_number_check(n):
    if n <= 1:
        print("Not a Perfect Number")
        return

    divisor_sum = 0

    for i in range(1, n // 2 + 1):
        if n % i == 0:
            divisor_sum += i

    if divisor_sum == n:
        print("Perfect Number")
    else:
        print("Not a Perfect Number")

perfect_number_check(28)  # Output: Perfect Number
```

```
PS C:\Users\alish\OneDrive\Documents\Desktop\Winter training\Moya> & C:/Users/alish/AppData/Local/Microsoft/WindowsApps/pyth
on3.12.exe "c:/Users/alish/OneDrive/Documents/Desktop/Winter training/Moya/lab3.1-4049.py"
Perfect Number
PS C:\Users\alish\OneDrive\Documents\Desktop\Winter training\Moya>
```

Input and Output

Input	Output	Reason
6	Perfect Number	Sum of proper divisors equals 6.
28	Perfect Number	Sum of divisors equals the number.
12	Not a Perfect Number	Divisor sum exceeds the input.
1	Not a Perfect Number	No proper divisors exist.
-5	Not a Perfect Number	Negative values are invalid.

Explanation

The program efficiently finds divisor pairs up to √n and calculates their sum. If the sum matches the original number, it is identified as a perfect number.

QUESTION 6: Few-Shot Prompting (Even or Odd with Validation)

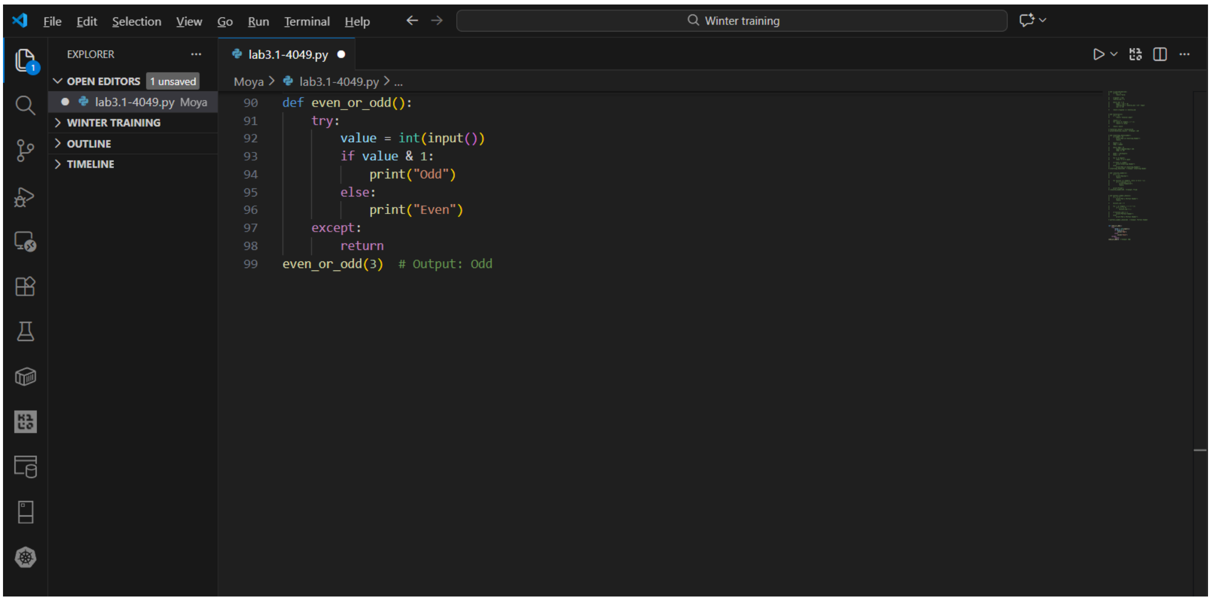
Problem Statement

Use few-shot prompting to generate a program that classifies numbers as even or odd with proper validation.

Prompt

Write a Python program that checks whether a number is Even or Odd. The program should validate input, correctly process negative numbers, and print only the classification result.

Generated Code Screenshot



```python
def even_or_odd():
    try:
        value = int(input())
        if value & 1:
            print("Odd")
        else:
            print("Even")
    except:
        return
even_or_odd(3)  # Output: Odd
```

Input and Output

Input	Output	Reason
8	Even	Divisible by 2.
15	Odd	Not divisible by 2.
0	Even	Zero is an even number.
-4	Even	Negative even numbers remai
-7	Odd	Negative odd numbers remain

Explanation

The program ensures valid integer input and applies the modulo operation to determine parity. Based on the remainder, it prints either Even or Odd.