Ai Assisted coding

Lab Assignment 10.3

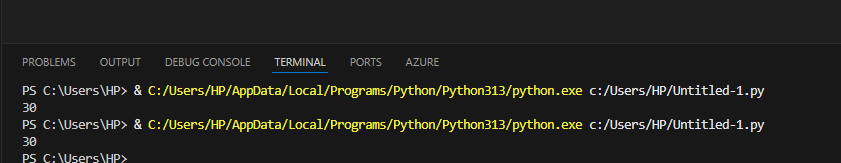
Roll no :2403A510A9

Batch:05

# Task 1 – Syntax and Error Detection

## Prompt: I have a Python script with syntax, indentation, and variable errors. Please identify and fix them # buggy\_code\_task1.py def add\_numbers(a, b) result = a + b return reslt print(add\_numbers(10 20))

## Code: Output:



## Observation:

- Added missing ':' after function definition  
- Fixed typo 'reslt' → 'result'  
- Added missing comma in function call  
- Corrected indentation

# Task 2

**Prompt:**

I have a Python script that finds duplicate numbers in a list, but the logic is inefficient because it uses nested loops. Please optimize the code so that it still produces the correct result but runs more efficiently.

## Code:

## Output:



## Observation:

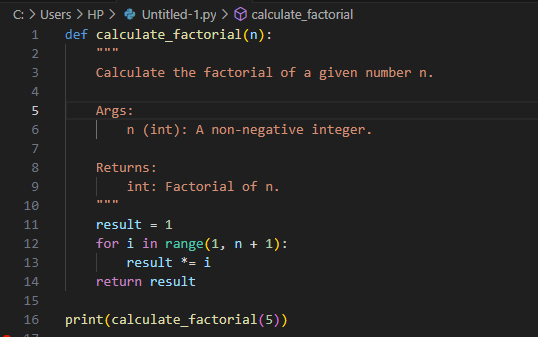
- Replaced nested loops with sets to achieve O(n) time complexity  
- Avoids redundant checks and improves readability

# Task 3 – Code Refactoring for Readability

## Prompt:

I have a Python script that calculates the factorial of a number, but the code is messy and not PEP 8–compliant. Please refactor it into a clean, well-structured version with: • Proper indentation and formatting. • A meaningful function name (calculate\_factorial). • Clear variable naming. • A docstring explaining the function

## Code:



## Output:



## Observation:

- Renamed function to 'calculate\_factorial'  
- Added descriptive docstring  
- Improved variable naming  
- Applied PEP 8 formatting

# Task 4 – Security and Error Handling Enhancement

## Prompt:

I have a Python script that calculates the factorial of a number, but the code is messy and not PEP 8–compliant. Please refactor it into a clean, well-structured version with: • Proper indentation and formatting. • A meaningful function name (calculate\_factorial). • Clear variable naming. • A docstring explaining the function

## Code:

## Output:



## Observation:

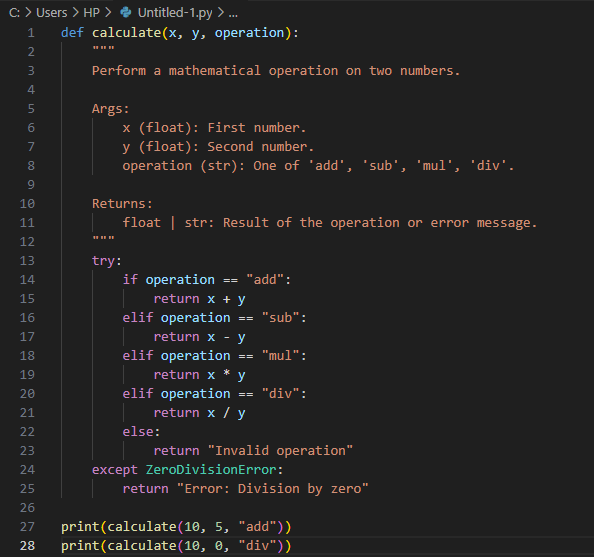
- Prevented SQL injection using parameterized queries  
- Added try-except-finally for safe database handling  
- Added input validation for security

# Task 5 – Automated Code Review Report Generation

## Prompt:

You are provided with a simple calculator function that performs addition, subtraction, multiplication, and division.   
However, the code is messy and lacks good practices. Your task is to conduct an automated-style code review and prepare a report that highlights the following issues:  
- Missing docstrings and documentation.  
- Inconsistent formatting (indentation and inline statements).  
- Lack of error handling (e.g., division by zero).  
- Use of non-descriptive names for the function and parameters.  
- Overall readability and PEP 8 compliance issues.  
After writing the review report, refactor the code to address these issues and demonstrate an improved, clean version of the calculator function.

## Code:



## Output:



## Observation:

- Added descriptive docstring  
- Renamed function and variables for clarity  
- Implemented division by zero error handling  
- Applied consistent PEP 8 formatting