Ai Assisted coding

Lab Assignment 10.3

Roll no :2403A510A3

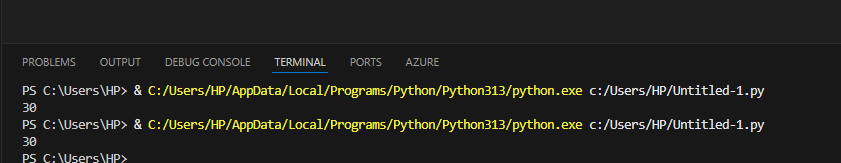
Batch:06

# Task 1 – Syntax and Error Detection

## Prompt:

You are given a Python script with several syntax, indentation, and variable name errors.   
Your task is to carefully inspect the code, identify all the mistakes, and correct them so that the program runs successfully and produces the correct output. Specifically:  
- Ensure the function definition has the proper syntax (including a colon).  
- Correct any misspelled variable names inside the function.  
- Fix incorrect function calls (e.g., missing commas between arguments).  
- Maintain proper indentation throughout the code.  
After fixing the issues, run the corrected program to verify it produces the expected result.

## Code: Output:



## Observation:

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

# Task 2 – Logical and Performance Issue Review

## Prompt:

You are given a Python program that finds duplicates in a list.   
However, the current implementation uses nested loops and checks each element against every other element, which makes the program inefficient (O(n²) complexity).   
Your task is to:  
- Identify the inefficiency in the current implementation.  
- Rewrite the function in a more optimized way using sets or dictionaries to reduce time complexity to O(n).  
- Ensure the program still produces the correct result.  
- Provide an explanation of why the new solution is better in terms of performance and readability.

## Code:

## Output:



## Observation:

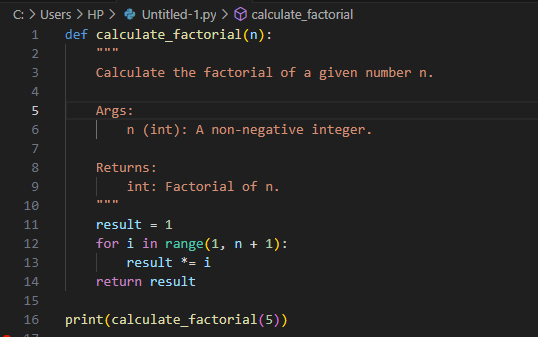
- Replaced nested loops with sets to achieve O(n) time complexity  
- Avoids redundant checks and improves readability

# Task 3 – Code Refactoring for Readability

## Prompt:

The given Python script calculates the factorial of a number but is poorly written and not PEP 8 compliant.   
Your task is to refactor the code by:  
- Renaming the function and variables to meaningful, descriptive names.  
- Adding a proper docstring that explains the function’s purpose, parameters, and return value.  
- Formatting the code according to PEP 8 guidelines (indentation, spacing, line length).  
- Ensuring the logic remains correct and the program outputs the correct factorial value.  
After refactoring, run the program and confirm that it produces the expected result.

## 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:

The given Python script connects to a SQLite database and retrieves user data by constructing an SQL query using string concatenation.   
This makes the program vulnerable to SQL injection attacks. Your task is to:  
- Modify the query to use parameterized SQL queries (? placeholders) instead of string concatenation.  
- Add exception handling (try-except-finally) to safely catch database errors and ensure the connection is properly closed.  
- Implement input validation to prevent invalid user inputs (e.g., ensure the entered ID is an integer).  
- Maintain the same functionality but ensure the program is both secure and reliable.

## Code:

## Output:



## Observation:

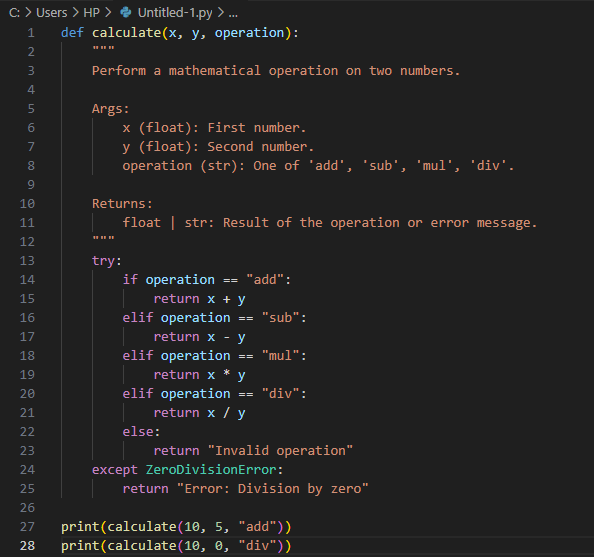
- Prevented SQL injection using parameterized queries  
- Added try-except-finally for safe database handling  
- Added input validation for security

# 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