

SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE		DEPARTMENT OF COMPUTER SCIENCE ENGINEERING	
Program Name: B. Tech		Assignment Type: Lab	Academic Year:2025-2026
Course Coordinator Name		Venkataramana Veeramsetty	
Instructor(s) Name		Dr. V. Venkataramana (Co-ordinator)	
		Dr. T. Sampath Kumar	
		Dr. Pramoda Patro	
		Dr. Brij Kishor Tiwari	
		Dr.J.Ravichander	
		Dr. Mohammand Ali Shaik	
		Dr. Anirodh Kumar	
		Mr. S.Naresh Kumar	
		Dr. RAJESH VELPULA	
		Mr. Kundhan Kumar	
		Ms. Ch.Rajitha	
		Mr. M Prakash	
		Mr. B.Raju	
		Intern 1 (Dharma teja)	
		Intern 2 (Sai Prasad)	
		Intern 3 (Sowmya)	
		NS_2 ( Mounika)	
Course Code	24CS002PC215	Course Title	AI Assisted Coding
Year/Sem	II/I	Regulation	R24
Date and Day of Assignment	Week5 - Monday	Time(s)	
Duration	2 Hours	Applicable to Batches	
AssignmentNumber: 9.1(Present assignment number)/24(Total number of assignments)			
Q.No.	Question		Expected Time to complete
1	<b>Lab 9 – Code Review and Quality: Using AI to improve code quality and readability</b>  <b>Lab Objectives</b> <ul style="list-style-type: none"> <li>• Inline comments</li> <li>• Docstrings</li> <li>• Auto-documentation tools</li> <li>• AI-assisted summarization</li> </ul>		Week5 - Monday

**Task Description #1 (AI-Assisted Bug Detection)**

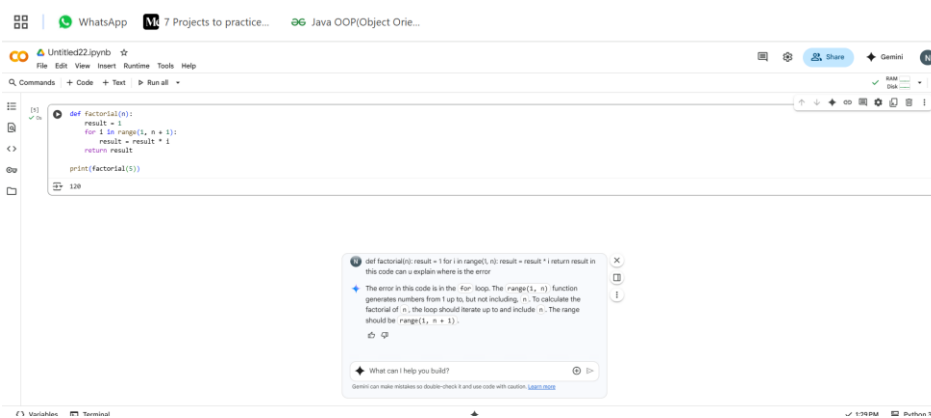
**Scenario:** A junior developer wrote the following Python function to calculate factorials:

```
def factorial(n):
    result = 1
    for i in range(1, n):
        result = result * i
    return result
```

PROMPT1:The above code is not giving the code

PROMPT2:Can u explain why it is error

CODE & OUTPUT:



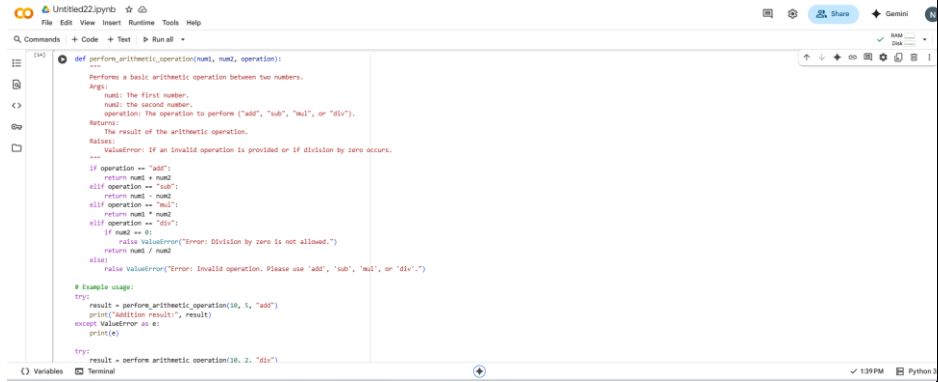
**EXPLANATION FROM AI:**

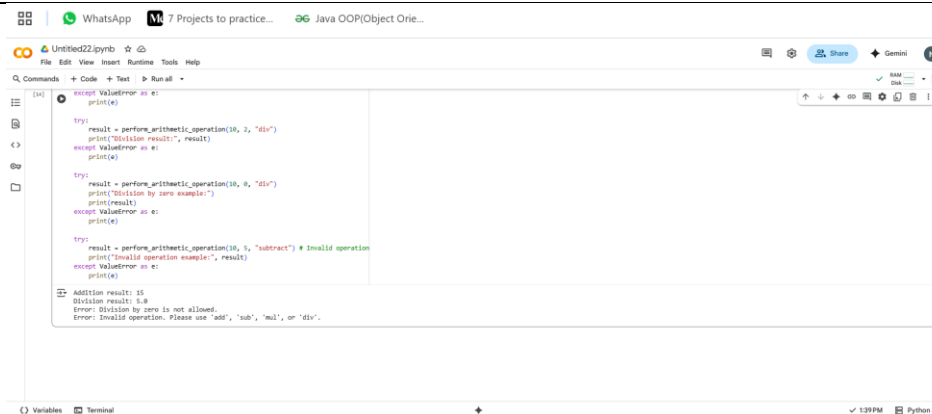
The error in this code is in the for loop. The range(1, n) function generates numbers from 1 up to, but not including, n. To calculate the factorial of n, the loop should iterate up to and include n. The range should be range(1, n + 1).

**OBSERVATION:**

From the above code before AI Correction , I observed that in the for loop if we write 5 in the input then the number of iterations are 4 (1,2,3,4). One more iteration is going to be done we will get an wrong output.And also there is no print function to print the output. After Giving the code the iteration are 5 and also it modifies the code and given the correct output with the understandable explanation.

- Run the code and test it with factorial(5)

	<p>(expected output = 120).</p> <ul style="list-style-type: none"><li>• Use AI (prompting) to review this code and identify the bug.</li><li>• Ask AI to suggest corrections and rewrite the code.</li><li>• Compare AI's corrected code with your own fix.</li></ul>	
	<p><b>Task Description #2</b> (Improving Readability &amp; Documentation)</p> <p><b>Scenario:</b> The following code works but is poorly written:</p> <pre>def calc(a,b,c):     if c=="add":         return a+b     elif c=="sub":         return a-b     elif c=="mul":         return a*b     elif c=="div":         return a/b</pre> <p>PROMPT1: From the above code rewrite the code with clear functions ,variable names,proper docstring and exception handling for division by zero.</p> <p>PROMPT2:Its not giving an output can u tell why it not giving?</p> <p><b>CODE &amp; OUTPUT:</b></p> 	



```
def perform_arithmetic_operation(num1, num2, operation):
    """Perform arithmetic operation on two numbers.

    Args:
        num1 (int): The first number.
        num2 (int): The second number.
        operation (str): The operation to perform ('add', 'sub', 'mul', or 'div').

    Returns:
        int: The result of the operation.

    Raises:
        ValueError: If the operation is invalid or the second number is zero for division.
    """
    try:
        result = perform_arithmetic_operation(num1, num2, operation)
        print("Division result:", result)
    except ValueError as e:
        print(e)

    try:
        result = perform_arithmetic_operation(num1, 0, "div")
        print("Division by zero example:")
        print(result)
    except ValueError as e:
        print(e)

    try:
        result = perform_arithmetic_operation(num1, 1, "subtract") # Invalid operation
        print("Invalid operation example:", result)
    except ValueError as e:
        print(e)

# Addition result: 11
# Division result: 5.0
# Error: Division by zero is not allowed.
# Error: Invalid operation. Please use 'add', 'sub', 'mul', or 'div'.
```

## EXPLANATION FROM AI:

### 1. Readability and Naming:

- The function name is changed from the generic `calc` to a more descriptive `perform_arithmetic_operation`, which clearly indicates its purpose.
- Variable names `a`, `b`, and `c` are replaced with more meaningful names `num1`, `num2`, and `operation`, making the code easier to understand.

### 2. Documentation (Docstring):

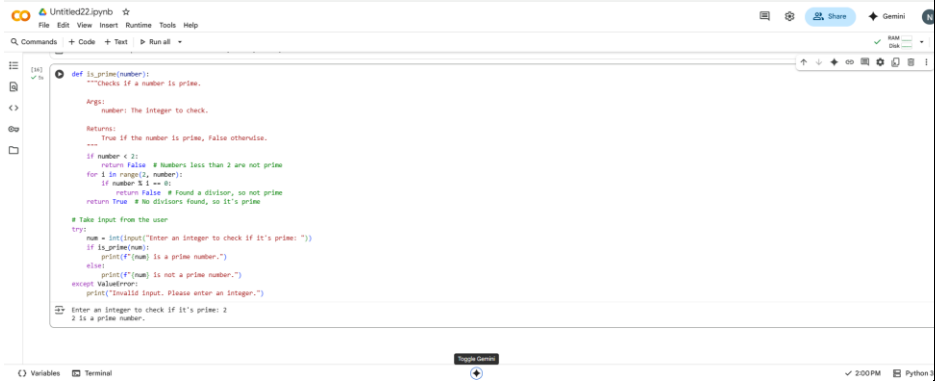
- The improved version includes a comprehensive docstring that explains what the function does, its arguments (Args), what it returns (Returns), and the potential errors it might raise (Raises). This makes it much easier for other developers (or your future self) to understand how to use the function.

### 3. Exception Handling for Division by Zero:

- The improved function explicitly checks if the second number (`num2`) is zero when the operation is `"div"`.
- Instead of potentially causing a `ZeroDivisionError` and crashing the program, it raises a `ValueError` with a clear error message. This allows the calling code to handle the error gracefully using a `try...except` block.

## OBSERVATION:

From the above code what is observed is function name is changed from `calc` to `perform_arithmetic_operation`. If every thing goes smooth then it returns the value. Otherwise it raises the error. Before giving code to AI given variables as `a, b` but after the variable names are `num1, num2`. if `num2` is zero then it raises an error because anything divides by zero is undefined/ not possible.

	<p>.Use AI to review this code for readability, naming, and documentation issues.</p> <ul style="list-style-type: none"><li>● Prompt AI to rewrite the code with:<ul style="list-style-type: none"><li>○ Clear function &amp; variable names.</li><li>○ Proper docstrings.</li><li>○ Exception handling for division by zero.</li></ul></li><li>● Compare the before-and-after versions to evaluate AI's contribution.</li></ul>	
	<p><b>Task Description #3</b> (Enforcing Coding Standards)</p> <p><b>Scenario:</b> A team project requires following PEP8 style guide. One developer submits:</p> <pre>def Checkprime(n):     for i in range(2,n):         if n%i==0:             return False     return True</pre> <p>PROMPT1:From the above code use code code quality review for PEP8 compliances and return with proper naming,indentation,spacing.</p> <p>PROMPT2:Explain the code why it is error?</p> <p>PROMPT3:Take inputs after the execution.</p> <p>CODE &amp;OUTPUT:</p>  <p><b>EXPLANATION FROM AI:</b></p> <ol style="list-style-type: none"><li>1. <b>Naming:</b> The function name Checkprime uses CamelCase, which is not the standard in Python. PEP 8 recommends using snake_case for function and variable names (e.g., check_prime or is_prime).</li><li>2. <b>Indentation and Spacing:</b> The indentation is inconsistent, and there's a lack of spacing around operators and after commas,</li></ol>	

	<p>which makes the code harder to read. PEP 8 suggests using 4 spaces for indentation and adding spaces around operators (+, -, *, /, ==, etc.).</p> <ol style="list-style-type: none"><li>3. <b>Missing Docstring:</b> The function lacks a docstring. A docstring is crucial for explaining what the function does, its arguments, and what it returns.</li><li>4. <b>Efficiency (Optional but good practice):</b> For checking primality, you only need to check for divisibility up to the square root of n. Checking up to n is less efficient for larger numbers. Also, handling numbers less than 2 as non-prime is a good practice.</li></ol> <p><b>OBSERVATION:</b></p> <p>From the above code ,I observed that it changed the function name before it is CheckPrime after its uses the Check_prime because it is one of the Standard condition PEP8 means enforcing coding standard which is nothing there is some rules of the code like indentation,naming and the proper spacing.After giving code to AI it modified the code and giving the correct code.</p> <ul style="list-style-type: none"><li>• Run this code and verify correctness.</li><li>• Use AI to perform a code quality review for PEP8 compliance.</li><li>• Prompt AI to return a refactored version with proper indentation, spacing, and naming conventions.</li><li>• Discuss how automated AI review can save time in large-scale projects.</li></ul>	

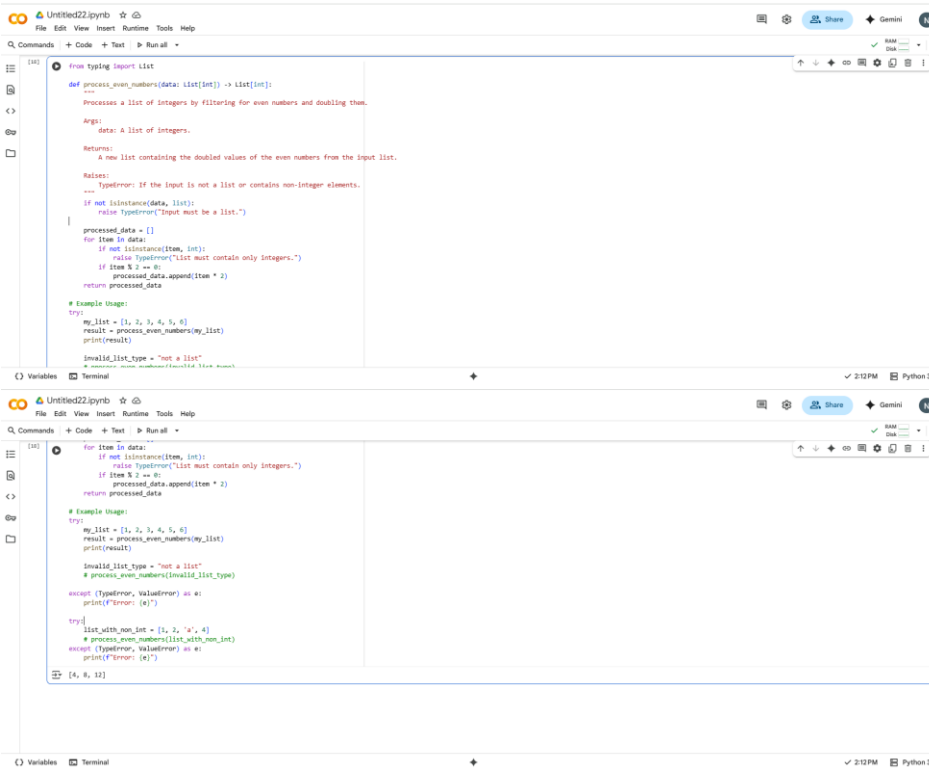
**Task Description #4 (AI as a Code Reviewer in Real Projects)**

**Scenario:** You are part of a GitHub project. A teammate submits this pull request:

```
def processData(d):  
    return [x*2 for x in d if x%2==0]
```

**PROMPT:** From the above code focus on the naming conventions,input validations and adding type hints.

**CODE & OUTPUT:**



**OBSERVATION:**

From the above code I observed that it takes the inputs like set of list in numbers. And check whether any number in the list have the even number or not. If any number is even then it doubles the value and check all numbers in a list. And doubles the all even numbers. Then it gives the output.

- Review this function manually for readability, reusability, and edge cases.
- Use AI to generate a code review comment, focusing on:
  - Naming conventions.
  - Input validation (e.g., what if d is not a list?).
  - Adding type hints.
- Modify the function based on AI’s suggestions.

	<ul style="list-style-type: none"> <li>• Write a short reflection: <i>Would you trust AI as a standalone reviewer, or only as a support tool? Why?</i>  <i>MY ANS:I go with the Only as a support tool because if I completely depend on the AI then I won't get the Knowledge. So I will use AI but I Will with a not depend on AI. I work with my Own knowledge in any code .If I get any error (or) if iam unable to correct the error then I will ask the AI to get mistake in my code.So that, I will more knowledge from myside and and I can get new thing from AI also.</i></li> </ul>	
--	--	--