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	**Lab 9 – Code Review and Quality: Using AI to improve code quality and readability** **Lab Objectives** • Inline comments • Docstrings • Auto-documentation tools • AI-assisted summarization	Week5 - Monday

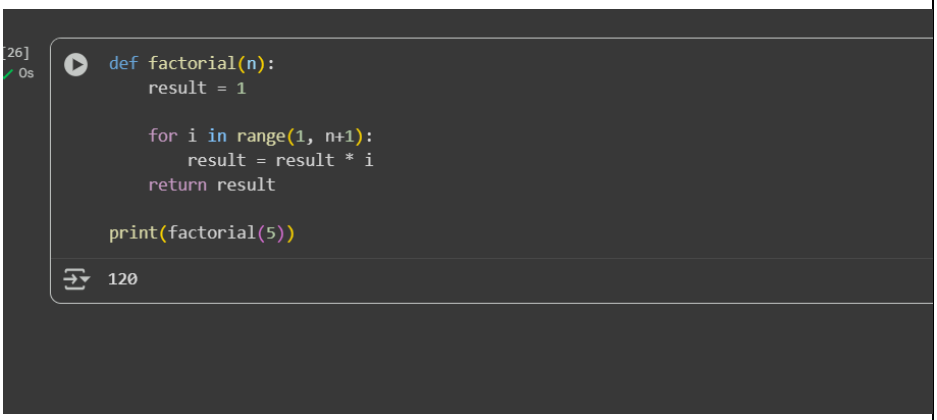
Task Description #1 (AI-Assisted Bug Detection)

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
```

- Run the code and test it with factorial(5)
  (expected output = 120).
- Use AI (prompting) to review this code and identify the bug.
- Ask AI to suggest corrections and rewrite the code.
- Compare AI's corrected code with your own fix.

ANSWERS:



BUG:

1.In range function nth term was excluded so to include nth term I have added +1 to it.

2. and to print the output we have to add extra line i.e (print(factorial(5))

Comparison:

I have some indentation error while fixing my code which is solved by ai suugestion .

Task Description #2 (Improving Readability & Documentation)

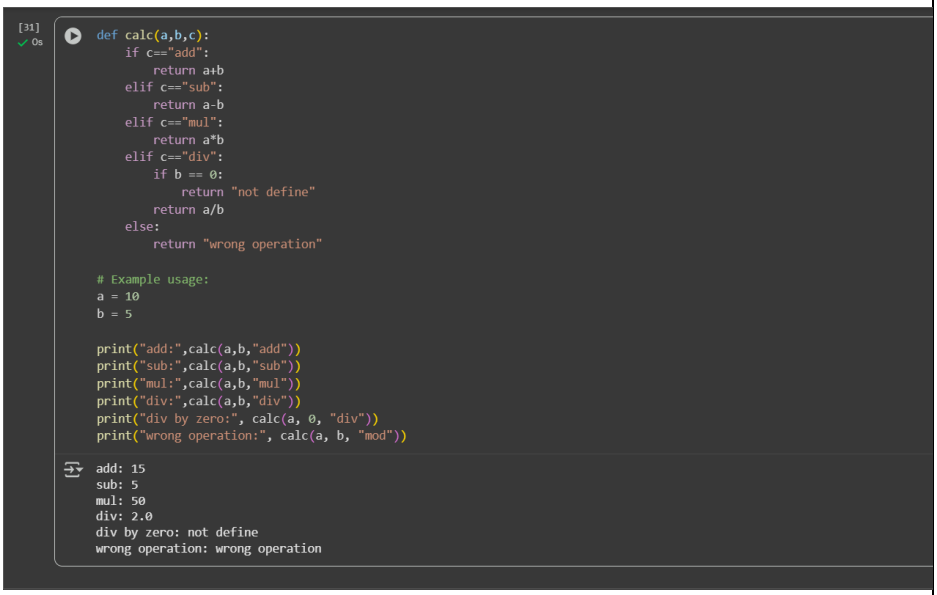
Scenario: The following code works but is poorly written:

```
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
```

- Use AI to review this code for readability, naming, and documentation issues.
- Prompt AI to rewrite the code with:
    - Clear function & variable names.
    - Proper docstrings.
    - Exception handling for division by zero.
- Compare the before-and-after versions to evaluate AI's contribution.

ANSWERS:

SS:



```
[31]
0s
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":
        if b == 0:
            return "not define"
        return a/b
    else:
        return "wrong operation"

# Example usage:
a = 10
b = 5

print("add:",calc(a,b,"add"))
print("sub:",calc(a,b,"sub"))
print("mul:",calc(a,b,"mul"))
print("div:",calc(a,b,"div"))
print("div by zero:", calc(a, 0, "div"))
print("wrong operation:", calc(a, b, "mod"))

add: 15
sub: 5
mul: 50
div: 2.0
div by zero: not define
wrong operation: wrong operation
```

BUG:

1. **Division by zero not handled**
2. **No case for invalid operations**

Comparison:

code fixes by me and the AI was same .

Task Description #3 (Enforcing Coding Standards)
Scenario: A team project requires following PEP8 style guide. One developer submits:

```
def Checkprime(n):
 for i in range(2,n):
  if n%i==0:
   return False
 return True
```

- Run this code and verify correctness.
- Use AI to perform a code quality review for PEP8 compliance.
- Prompt AI to return a refactored version with proper indentation, spacing, and naming conventions.
- Discuss how automated AI review can save time in large-scale projects.

ANSWERS:

```
def is_prime(n):
    for i in range(2, n):
        if n % i == 0:
            return False
    return True

print(is_prime(2))
print(is_prime(10))
print(is_prime(7))
```

```
True
False
True
```

OBSERVATION:

. Changes name to is_prime.
. Proper **indentation** and **spacing**.
. More **efficient** prime check (up to √n).

DISCUSSION:

. Consistency
.Error Prevention
. Scalability

.**Productivity**.

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]
```

- 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.
- Write a short reflection: Would you trust AI as a standalone reviewer, or only as a support tool? Why?

ANSWERS:

```python
from typing import List, Union

def process_data(data: List[Union[int, float]]) -> List[Union[int, float]]:
    """
    Process a list of numbers, doubling only the even ones.
    Works with integers and float values like 2.0, 4.0, etc.
    """
    if not isinstance(data, list):
        raise TypeError("Input must be a list.")

    processed_data = []
    for x in data:
        if isinstance(x, (int, float)) and x % 2 == 0:
            processed_data.append(x * 2)
    return processed_data



print(process_data([1, 2, 3, 4, 5, 6]))
print(process_data([2.0, 3.5, 4.0, 5.2]))
print(process_data([10, 15, 20.0, 25]))
print(process_data([]))
```

```
[4, 8, 12]
[4.0, 8.0]
[20, 40.0]
[]
```

MANUAL OBSERVATION:
1.Function name should be process_data.
2.Missing input validation

	3.No type hints. 4.No docstring. **Reflection:** • AI review is **useful for style, validation, and documentation**. • However, in critical projects, AI should be used as a **support tool** alongside human reviewers for logic and security checks.	