

SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE		DEPARTMENT OF COMPUTER SCIENCE ENGINEERING	
ProgramName: B. Tech		Assignment Type: Lab	AcademicYear: 2025-2026
CourseCoordinatorName		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)			
CourseCode	24CS002PC215	CourseTitle	AI Assisted Coding
Year/Sem	II/I	Regulation	R24
Date and Day of Assignment	Week1 - Tuesday	Time(s)	
Duration	2 Hours	Applicable to Batches	24CSBTB01 To 24CSBTB39
AssignmentNumber: 1.2 (Present assignment number) / 24 (Total number of assignments)			
Q.No.	Question	Expected Time to complete	
1	Lab 1: Environment Setup – GitHub Copilot and VS Code Integration Lab Objectives: <ul style="list-style-type: none"> To install and configure GitHub Copilot in Visual Studio Code. To explore AI-assisted code generation using GitHub Copilot. 	Week1 - wednesday	

- To analyze the accuracy and effectiveness of Copilot's code suggestions.
- To understand prompt-based programming using comments and code context

Lab Outcomes (LOs):

After completing this lab, students will be able to:

- Set up GitHub Copilot in VS Code successfully.
- Use inline comments and context to generate code with Copilot.
- Evaluate AI-generated code for correctness and readability.
- Compare code suggestions based on different prompts and programming styles.

Task Description#1

- Write a comment: # Function to check if a string is a valid palindrome (ignoring spaces and case) and allow Copilot to complete it.

Expected Output#1

- A function that correctly returns True for phrases like "A man a plan a canal Panama"
- Prompt: write a python code to check if function is a string and that was a valid palindrome or not with input value

```

lab program 1.py × Extension: Python Debugger
lab program 1.py > ...
1  def is_string_palindrome(value):
2      if not isinstance(value, str):
3          return "Input is not a string."
4      cleaned = ''.join(c.lower() for c in value if c.isalnum())
5      if cleaned == cleaned[::-1]:
6          return "Valid palindrome."
7      else:
8          return "Not a palindrome."
9
10 # Example usage:
11 user_input = input("Enter a value: ")
12 result = is_string_palindrome(user_input)
13 print(result)
14
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Enter a value: a man a plan a canal panama
Valid palindrome.
PS C:\Users\DEEKSHA\OneDrive\Desktop\AI.py> & C:/Users/DEEKSHA/AppData/Local/Programs/Microsoft VS Code/bin/python /lab program 1.py"
Enter a value: a man a plan a canal panama no
Not a palindrome.
PS C:\Users\DEEKSHA\OneDrive\Desktop\AI.py> & C:/Users/DEEKSHA/AppData/Local/Programs/Microsoft VS Code/bin/python /lab program 1.py"

```

Task Description#2

- Generate a Python function that returns the Fibonacci sequence up to n terms. Prompt with only a function header and docstring

Expected Output#2

- AI completes the function logic using loop or recursion with accurate output
- PROMT : write a python code that function return the fibonacci sequence up to n terms and use loop with input value

```
def fibonacci_sequence(n):  
    sequence = []  
    a, b = 0, 1  
    for _ in range(n):  
        sequence.append(a)  
        a, b = b, a + b  
    return sequence  
  
# Get input from user  
n_terms = int(input("Enter the number of terms: "))  
fib_seq = fibonacci_sequence(n_terms)  
print("Fibonacci sequence up to", n_terms, "terms:", fib_seq)
```

- Enter the number of terms: 10
- Fibonacci sequence up to 10 terms: [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]

Task Description#3

- Write a comment like # Function to reverse a string and use Copilot to generate the function.

Expected Output#3

- Auto-completed reverse function
- PROMT: write the python code for a function to reverse a string function

```
def reverse_string(s):  
    """  
    Reverses the given string.  
  
    Args:  
        s (str): The string to reverse.  
  
    Returns:  
        str: The reversed string.  
    """  
    return s[::-1]  
  
if __name__ == "__main__":  
    input_str = "hello"  
    print("Original:", input_str)  
    print("Reversed:", reverse_string(input_str))
```

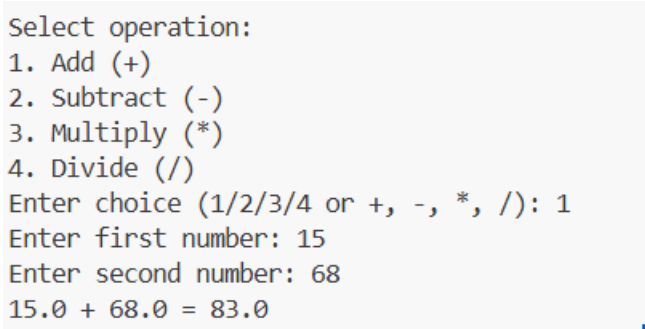
- Original: hello
Reversed: olleh

Task Description#4

- Generate a program that simulates a basic calculator (add, subtract, multiply, divide).
Write the comment: # Simple calculator with 4 operations and let AI complete it.

Expected Output#4

- Fully working calculator with input/output and operator selection logic
- PROMT: write a python code that was a basic calculator that do

	<p>(add,subtract,multiply,divide) with operator logics and also with input value</p> <pre> def add(x, y): return x + y def subtract(x, y): return x - y def multiply(x, y): return x * y def divide(x, y): if y == 0: return "Error! Division by zero." return x / y print("Select operation:") print("1. Add (+)") print("2. Subtract (-)") print("3. Multiply (*)") print("4. Divide (/)") choice = input("Enter choice (1/2/3/4 or +, -, *, /): ") num1 = float(input("Enter first number: ")) num2 = float(input("Enter second number: ")) if choice in ('1', '+'): print(f"{num1} + {num2} = {add(num1, num2)}") elif choice in ('2', '-'): print(f"{num1} - {num2} = {subtract(num1, num2)}") elif choice in ('3', '*'): print(f"{num1} * {num2} = {multiply(num1, num2)}") elif choice in ('4', '/'): print(f"{num1} / {num2} = {divide(num1, num2)}") else: print("Invalid input") </pre> <ul style="list-style-type: none">  <pre> Select operation: 1. Add (+) 2. Subtract (-) 3. Multiply (*) 4. Divide (/) Enter choice (1/2/3/4 or +, -, *, /): 1 Enter first number: 15 Enter second number: 68 15.0 + 68.0 = 83.0 </pre> <p>Task Description#5</p> <ul style="list-style-type: none"> Use a comment to instruct AI to write a function that reads a file and returns the number of lines.. <p>Expected Output#5</p> <ul style="list-style-type: none"> Functional implementation using open() or with open() and readlines() PROMT: write python code that function reads a file and return the number 	
--	--	--

of lines with input value

```
with open('example.txt', 'w') as f:
    f.write('HELLO!!\n')
    f.write("THIS SUBJECT WAS AIAC.\n")
    f.write("BY USING PYTHON CODE.\n")

def count_lines_with_value(filename, value):
    count = 0
    with open(filename, 'r') as file:
        for line in file:
            if value in line:
                count += 1
    return count

try:
    result = count_lines_with_value('example.txt', 'HELLO')
    print(result)
except FileNotFoundError:
    print("The file 'example.txt' was not found.")
```

1

Note: Report should be submitted a word document for all tasks in a single document with prompts, comments & code explanation, and output and if required, screenshots

Evaluation Criteria:

Criteria	Max Marks
Task #1	0.5
Task #2	0.5
Task #3	0.5
Task #4	0.5
Task #5	0.5
Total	2.5 Marks