

AI Assisted Coding-ASSIGNMENT-1

Task 1:

Write a Function to check if a string is a valid palindrome (ignoring spaces and case).

The screenshot shows the PyCharm IDE interface. The left sidebar displays a project structure under 'EXPLORER' with several sub-directories like '1_lab', '5th_lab', '6th_lab', and files such as 'task(1).py', 'task(2).py', etc. The main editor window contains Python code for a palindrome checker:

```
task(1).py
1 #Function to check if a string is palindrome or not
2 def is_palindrome(s):
3     # Convert the string to lowercase and remove spaces
4     s = s.replace(" ", "").lower()
5     # Check if the string is equal to its reverse
6     return s == s[::-1]
7 # Test the function
8 test_string = input("Enter a string to check if it's a palindrome: ")
9 if is_palindrome(test_string):
10    print(f'{test_string} is a palindrome.')
11 else:
12    print(f'{test_string} is not a palindrome.')

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
```

The 'TERMINAL' tab at the bottom shows the command line output of running the script:

```
PS C:\Users\haru\OneDrive\Desktop\AI CODING> & 'c:\Program Files\Python311\python.exe' 'c:\Users\haru\.vscode\extensions\ms-python.debugpy-2025.10.0-win32-x64\bundle\libs\debugpy\launcher' '62243' '--' 'c:\Users\haru\OneDrive\Desktop\AI CODING\1_lab\task(1).py'
Enter a string to check if it's a palindrome: mom
"mom" is a palindrome.

PS C:\Users\haru\OneDrive\Desktop\AI CODING>
```

NAME: B.Nishant

HT.NO: 2403A52089

BATCH: AIB04

Task 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

The screenshot shows the PyCharm IDE interface. In the top navigation bar, the file 'task(2).py' is selected. The code editor displays the following Python function definition:

```
1_lab > task2.py > fibonacci_sequence
1   def fibonacci_sequence(n):
2       """
3           Generate the Fibonacci sequence up to n terms.
4       """
5       Args:
6           n (int): The number of terms in the Fibonacci sequence to generate.
7       Returns:
8           list: A list containing the Fibonacci sequence up to n terms.
9       """
10      if n <= 0:
11          return []
12      elif n == 1:
13          return [0]
14      elif n == 2:
15          return [0, 1]
16      sequence = [0, 1]
17      for i in range(2, n):
18          sequence.append(sequence[-1] + sequence[-2])
19      return sequence
20
21  # Test the function
22 num_terms = int(input("Enter the number of terms for the Fibonacci sequence: "))
23 print(f"Fibonacci sequence up to {num_terms} terms: {fibonacci_sequence(num_terms)}")
```

The bottom terminal window shows the command-line output of running the script:

```
PS C:\Users\tharu\OneDrive\Desktop\AI CODING> ^
PS C:\Users\tharu\OneDrive\Desktop\AI CODING> c;; cd 'c:\Users\tharu\OneDrive\Desktop\AI CODING'; & 'c:\Program Files\Python311\python.exe' 'c:\Users\tharu\vscode\extensions\ms-python.python.debugpy-2025.10.0-win32-x64\bundled\libs\debugpy\launcher' '56763' '--' 'c:\Users\tharu\OneDrive\Desktop\AI CODING\1_lab\task(2).py'
Enter the number of terms for the Fibonacci sequence: 5
Fibonacci sequence up to 5 terms: [0, 1, 1, 2, 3]
```

NAME: B.Nishant

HT.NO: 2403A52089

BATCH: AIB04

Task 3:

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

Expected Output#3

- Auto-completed reverse function

The screenshot shows the Visual Studio Code interface. The code editor displays a Python file named task3.py with the following content:

```
# Function to reverse a string
def reverse_string(s):
    """
    Reverse the given string.

    Args:
        s (str): The string to reverse.

    Returns:
        str: The reversed string.
    """
    return s[::-1]
# Test the function
test_string = input("Enter a string to reverse: ")
print(f"Reversed string: {reverse_string(test_string)}")
```

The terminal below shows the command line environment with the following session:

```
PS C:\Users\tharu\OneDrive\Desktop\AI CODING> ^C
PS C:\Users\tharu\OneDrive\Desktop\AI CODING>
PS C:\Users\tharu\OneDrive\Desktop\AI CODING> c:; cd 'c:\users\tharu\OneDrive\Desktop\AI CODING'; & 'c:\Program Files\Python311\python.exe' 'c:\Users\tharu\vscodeextensions\ms-python.debugpy-2025.10.0-win32-x64\bundled\libs\debugpy\launcher' '61182' '--' 'c:\Users\tharu\OneDrive\Desktop\AI CODING\1_lab\task3.py'
Enter a string to reverse: varshitha
Reversed string: ahtihsrav
PS C:\Users\tharu\OneDrive\Desktop\AI CODING>
```

NAME: B.Nishant

HT.NO: 2403A52089

BATCH: AIB04

Task 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

The screenshot shows the PyCharm IDE interface. The left sidebar displays a project structure with several sub-directories like 1st_lab, 2nd_lab, 3rd_lab, etc., each containing files such as task1.py, task2.py, task3.py, and task4.py. The main editor window shows the code for task4.py, which defines four operations: add, subtract, multiply, and divide. It includes a user interaction loop where it prompts the user to select an operation (1 for Add, 2 for Subtract, 3 for Multiply, 4 for Divide) and then takes two floating-point numbers as input to perform the selected operation. The code is well-formatted with comments and proper indentation.

```
# Simple calculator with 4 operations
def add(a, b):
    return a + b
def subtract(a, b):
    return a - b
def multiply(a, b):
    return a * b
def divide(a, b):
    if b != 0:
        return a / b
    else:
        return "Error: Division by zero is not allowed."
if __name__ == "__main__":
    print("Select operation:")
    print("1. Add")
    print("2. Subtract")
    print("3. Multiply")
    print("4. Divide")
    choice = input("Enter choice (1/2/3/4): ")
    if choice in ['1', '2', '3', '4']:
        num1 = float(input("Enter first number: "))
        num2 = float(input("Enter second number: "))
        if choice == '1':
            print(f"The result is: {add(num1, num2)}")
        elif choice == '2':
            print(f"The result is: {subtract(num1, num2)}")
        elif choice == '3':
            print(f"The result is: {multiply(num1, num2)}")
        elif choice == '4':
            print(f"The result is: {divide(num1, num2)}")
    else:
        print("Invalid input. Please select a valid operation.")
```

This screenshot shows the PyCharm IDE during the execution of task4.py. The terminal tab at the bottom is active, displaying the output of the program. The user has run the script, and the terminal shows the following interaction:

```
Select operation:
1. Add
2. Subtract
3. Multiply
4. Divide
Enter choice (1/2/3/4): 1
Enter first number: 10
Enter second number: 56
The result is: 66.0
```

The terminal also shows the current working directory as C:\Users\tharu\OneDrive\Desktop\AI CODING.

NAME: B.Nishant

HT.NO: 2403A52089

BATCH: AIB04

Task 5:

- Use a comment to instruct AI to write a function that reads a file and returns the number of lines.

Expected Output#5

- Functional implementation using open() or with open() and readlines()

The screenshot shows the Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure under "AI CODING". The "task5.py" file is selected.
- Code Editor:** Displays the Python code for "task5.py". The code defines a function `count_lines_in_file` that reads a file and returns the number of lines. It includes comments explaining the function's purpose, arguments, and return type. It uses a context manager `with` to handle file opening and `readlines()` to read all lines.
- Terminal:** Shows the command-line output of running the script. The user enters "task5.py" and the script prints "The result is: 66.0".
- Status Bar:** Shows the current file is "task5.py", line 22, column 49, with 4 spaces, encoding as UTF-8, CRLF, Python 3.11.5, and Go Live status.

NAME: B.Nishant

HT.NO: 2403A52089

BATCH: AIB04