AI Assisted Coding

Assignment - 1

B.Tejaskumar || 2303A51489|| Batch:- 8

Task 1: AI-Generated Logic Without Modularizaton (Fibonacci Sequence

Without Functons) **Code:**

```python
# Q1. Generate a python program to perform fibonacci series
# take user inputs
# Do not use functions

n = int(input("Enter the number of terms in Fibonacci series: "))

a, b = 0, 1
count = 0

if n <= 0:
    print("Please enter a positive integer.")

elif n == 1:
    print("Fibonacci series up to", n, ":")
    print(a)

elif n == 2:
    print("Fibonacci series up to", n, ":")
    print(a)
    print(b)

else:
    print("Fibonacci series:")
```

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

PS C:\Users\porika manikanta\OneDrive\Desktop\ai assistant> & "C:/Program Files/Python311/python.exe"
"c:/Users/porika manikanta/OneDrive/Desktop/ai assistant/lab 1.3.py"
Enter the number of terms in Fibonacci series: 10
Fibonacci series:
0 1 1 2 3 5 8 13 21 34
PS C:\Users\porika manikanta\OneDrive\Desktop\ai assistant>
```

Task 2: AI Code Optmizaton & Cleanup (Improving Efciency) **Code**:

```python
# Q2. Optimize this Fibonacci code
# Simplify logic and variable usage

n = int(input("Enter the number of terms in Fibonacci series: "))

a, b = 0, 1

for i in range(n):
    print(a, end=' ')
    a, b = b, a + b
```

```
 C:\Users\porika manikanta\OneDrive\Desktop\ai assistant> & "C:/Program Files/Python311/python.
c:/Users/porika manikanta/OneDrive/Desktop/ai assistant/lab 1.3.py"
ter the number of terms in Fibonacci series: 10
bonacci series:
1 1 2 3 5 8 13 21 34
 C:\Users\porika manikanta\OneDrive\Desktop\ai assistant> & "C:/Program Files/Python311/python.
c:/Users/porika manikanta/OneDrive/Desktop/ai assistant/lab 1.3.py"
ter the number of terms in Fibonacci series: 5
1 1 2 3
 C:\Users\porika manikanta\OneDrive\Desktop\ai assistant>
```

Task 3: Modular Design Using AI Assistance (Fibonacci Using Functions) **Code**:

```python
# Q3. Write python function to generate Fibonacci series up to n terms
# Use meaningful comments

def fibonacci_series(n):
    """
    Generate Fibonacci series up to n terms.

    Parameters:
    n (int): The number of terms in the Fibonacci series to generate.

    Returns:
    list: A list containing the Fibonacci series up to n terms.
    """

    series = []             # Initialize an empty list to store the Fibonacci series
    a, b = 0, 1             # Starting values for the Fibonacci series

    for i in range(n):      # Loop n times to generate n terms
        series.append(a)      # Append the current value of 'a' to the series
        a, b = b, a + b       # Update to the next Fibonacci numbers

    return series           # Return the generated Fibonacci series
```

OBLEMS   OUTPUT   DEBUG CONSOLE   **TERMINAL**   PORTS   Python + ⌄ ⟦

```
C:\Users\porika manikanta\OneDrive\Desktop\ai assistant> & "C:/Program Files/Python311/python.exe" "c
anikanta/OneDrive/Desktop/ai assistant/lab 1.3.py"
ter the number of terms in Fibonacci series: 10
bonacci series up to 10 terms:
, 1, 1, 2, 3, 5, 8, 13, 21, 34]
 C:\Users\porika manikanta\OneDrive\Desktop\ai assistant> |
```

Task 4: Comparative Analysis – Procedural vs Modular Fibonacci Code **Code:**

```python
84    # --------------- Modular approach ----------------
85    def fibonacci_modular(n):
86        a, b = 0, 1
87        series = []
88
89        for i in range(n):
90            series.append(a)
91            a, b = b, a + b
92
93        return series
94
95
96    n_modular = int(input("Enter the number of terms in Fibonacci series (Modular): "))
97    result_modular = fibonacci_modular(n_modular)
98
99    print("Fibonacci series (Modular):")
100   print(result_modular)
101
102
103
104
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   **TERMINAL**   PORTS   ▶ Python + ∨ ⫛

```
PS C:\Users\porika manikanta\OneDrive\Desktop\ai assistant> & "C:/Program Files/Python311/python.exe"
 manikanta/OneDrive/Desktop/ai assistant/lab 1.3.py"
Enter the number of terms in Fibonacci series (Procedural): 10
Fibonacci series (Procedural):
0 1 1 2 3 5 8 13 21 34
Enter the number of terms in Fibonacci series (Modular): 10
Fibonacci series (Modular):
[0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
PS C:\Users\porika manikanta\OneDrive\Desktop\ai assistant> ▌
```

Task 5: AI-Generated Iterative vs Recursive Fibonacci Approaches (Different

Algorithmic Approaches for Fibonacci Series) **Code:**

lab 1.3.py > ...

```python
101
102    # Generate python code for AI-Generated Iterative vs. Recursive Fibonacci Approaches
103    # (Different Algorithmic Approaches for Fibonacci Series)
104
105    # ---------------- Iterative approach ----------------
106    n_iter = int(input("Enter the number of terms in Fibonacci series (Iterative): "))
107
108    a, b = 0, 1
109    print("Fibonacci series (Iterative):")
110
111    for i in range(n_iter):
112        print(a, end=' ')
113        a, b = b, a + b
114
115    print()  # New line for better readability
116
117
118    # ---------------- Recursive approach ----------------
119    def fibonacci_recursive(n):
120        if n <= 0:
121            return []
122        elif n == 1:
123            return [0]
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   ⊠ Python + ∨ ⊡ 🗑

```
● PS C:\Users\porika manikanta\OneDrive\Desktop\ai assistant> & "C:/Program Files/Python311/python.exe" "c:/
  manikanta/OneDrive/Desktop/ai assistant/lab 1.3.py"
  Enter the number of terms in Fibonacci series (Iterative): 10
  Fibonacci series (Iterative):
  0 1 1 2 3 5 8 13 21 34
  Enter the number of terms in Fibonacci series (Recursive): 10
  Fibonacci series (Recursive):
  [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
○ PS C:\Users\porika manikanta\OneDrive\Desktop\ai assistant> ▊
```