

AI Assisted Coding

Assignment - 1

P.Manikanta || 2303A51271 || Batch:- 8

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

Without Functions) **Code:**

```
lab 1.3.py > ...
1  # Q1. Generate a python program to perform fibonacci series
2  # take user inputs
3  # Do not use functions
4
5 n = int(input("Enter the number of terms in Fibonacci series: "))
6
7 a, b = 0, 1
8 count = 0
9
10 if n <= 0:
11     print("Please enter a positive integer.")
12
13 elif n == 1:
14     print("Fibonacci series up to", n, ":")
15     print(a)
16
17 elif n == 2:
18     print("Fibonacci series up to", n, ":")
19     print(a)
20     print(b)
21
22 else:
23     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 Optimization & Cleanup (Improving Efficiency) **Code:**

```
8
9     # Q2. Optimize this Fibonacci code
0     # Simplify logic and variable usage
1
2     n = int(input("Enter the number of terms in Fibonacci series: "))
3
4     a, b = 0, 1
5
6     for i in range(n):
7         print(a, end=' ')
8         a, b = b, a + b
9
0
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
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"
Enter the number of terms in Fibonacci series: 10
Fibonacci 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"
Enter 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:**

```

9 # Q3. Write python function to generate Fibonacci series up to n terms
0 # Use meaningful comments
1
2 def fibonacci_series(n):
3     """
4         Generate Fibonacci series up to n terms.
5
6     Parameters:
7         n (int): The number of terms in the Fibonacci series to generate.
8
9     Returns:
0         list: A list containing the Fibonacci series up to n terms.
1     """
2
3     series = []          # Initialize an empty list to store the Fibonacci series
4     a, b = 0, 1          # Starting values for the Fibonacci series
5
6     for i in range(n):  # Loop n times to generate n terms
7         series.append(a) # Append the current value of 'a' to the series
8         a, b = b, a + b   # Update to the next Fibonacci numbers
9
0     return series        # Return the generated Fibonacci series

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Python + ▾

```

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 up to 10 terms:
[0, 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:**

```
lab 1.3.py > fibonacci_modular
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:**

```
>Welcome lab 4.3.py lab 1.3.py X
lab 1.3.py > ...
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>
```