

AI Assisted Coding

Assignment - 01

M.Sathwik || 2303A51483|| Batch:- 08

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

Without Functions) **Code:**

```
lab1.py > ...
1  # Q1. Generate a python program to perform fibonacci series
2  # take user inputs
3  # Do not use functions
4  n = int(input("Enter the number of terms in Fibonacci series: "))
5  a, b = 0, 1
6  count = 0
7  if n <= 0:
8      print("Please enter a positive integer.")
9  elif n == 1:
10     print("Fibonacci series up to", n, ":")
11     print(a)
12 elif n == 2:
13     print("Fibonacci series up to", n, ":")
14     print(a)
15     print(b)
16 else:
17     print("Fibonacci series:")
18     while count < n:
19         print(a, end=' ')
20         a, b = b, a + b
21         count += 1

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

● PS C:\Users\Sathwik\OneDrive\Desktop\AI-Lab> & C:/Users/Sathwik/AppData/Local/Microsoft/WindowsApps/python3.13.exe c:/Users/Sathwik/OneDrive/Desktop/AI-Lab/lab1.py
Enter the number of terms in Fibonacci series: 6
Fibonacci series:
0 1 1 2 3 5
○ PS C:\Users\Sathwik\OneDrive\Desktop\AI-Lab>
```

Task 2: AI Code Optimization & Cleanup (Improving Efficiency) **Code:**

```

24     # Q2.Optimize this Fibonacci code
25     # Simplify logic and variable usage
26     n = int(input("Enter the number of terms in Fibonacci series: "))
27     a, b = 0, 1
28     for _ in range(n):
29         print(a, end=' ')
```


```

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

```

33 # Q3.write python function to generate Fibonacci series up to n terms.use meaningful comments
34 def fibonacci_series(n):
35 """
36 Generate Fibonacci series up to n terms.
37
38 Parameters:
39 n (int): The number of terms in the Fibonacci series to generate.
40
41 Returns:
42 list: A list containing the Fibonacci series up to n terms.
43 """
44 series = [] # Initialize an empty list to store the Fibonacci series
45 a, b = 0, 1 # Starting values for the Fibonacci series
46 for _ in range(n):
47 series.append(a) # Append the current term to the series list
48 a, b = b, a + b # Update 'a' and 'b' to the next two Fibonacci numbers
49 return series # Return the generated Fibonacci series
50 if __name__ == "__main__":
51 n = int(input("Enter the number of terms in Fibonacci series: ")) # Take user input for the number of terms
52 result = fibonacci_series(n) # call the function to generate the Fibonacci series
53 print("Fibonacci series up to", n, "terms:")
54 print(result) # Print the resulting Fibonacci series
```


```

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

```

57  # Generate python program for comparative Analysis - Procedural vs Modular Fibonacci code
58  # Procedural approach
59  n = int(input("Enter the number of terms in Fibonacci series (Procedural): "))
60  a, b = 0, 1
61  print("Fibonacci series (Procedural):")
62  for _ in range(n):
63      print(a, end=' ')
64      a, b = b, a + b
65  print() #New line for better readability
66  # Modular approach
67  def fibonacci_series_modular(n):
68      a, b = 0, 1
69      series = []
70      for _ in range(n):
71          series.append(a)
72          a, b = b, a + b
73  return series
74 n_modular = int(input("Enter the number of terms in Fibonacci series (Modular): "))
75 result_modular = fibonacci_series_modular(n_modular)
76 print("Fibonacci series (Modular):")
77 print(result_modular) # Print the resulting Fibonacci series from the modular approach

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

○ PS C:\Users\Sathwik\OneDrive\Desktop\AI-Lab> & C:/Users/Sathwik/AppData/Local/Microsoft/WindowsApps/python3.13.exe c:/AI-Lab/lab1.py
 Enter the number of terms in Fibonacci series: 10
 Fibonacci series:
 0 1 1 2 3 5 8 13 21 34
 Enter the number of terms in Fibonacci series: 10
 0 1 1 2 3 5 8 13 21 34

Task 5: AI-Generated Iterative vs Recursive Fibonacci Approaches (Different Algorithmic Approaches for Fibonacci Series) Code:

```
80  # Generate python code for AI-Generated Iterative vs Recursive Fibonacci Approaches (Different Algorithmic Approaches for Fibonacci)
81  # Iterative approach
82  n_iter = int(input("Enter the number of terms in Fibonacci series (Iterative): "))
83  a, b = 0, 1
84  print("Fibonacci series (Iterative):")
85  for _ in range(n_iter):
86      print(a, end=" ")
87      a, b = b, a + b
88  print() # New line for better readability
89  # Recursive approach
90  def fibonacci_recursive(n):
91      if n <= 0:
92          return []
93      elif n == 1:
94          return [0]
95      elif n == 2:
96          return [0, 1]
97      else:
98          series = fibonacci_recursive(n - 1)
99          series.append(series[-1] + series[-2])
100     return series
101 n_rec = int(input("Enter the number of terms in Fibonacci series (Recursive): "))
102 result_recursive = fibonacci_recursive(n_rec)
103 print("Fibonacci series (Recursive):")
104 print(result_recursive) # Print the resulting Fibonacci series from recursive approach
105
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
hwik/AppData/Local/Microsoft/WindowsApps/python3.13.exe c:/Users/Sathwik/OneDrive/Desktop/AI-Lab/lab1.py
hwik/AppData/Local/Microsoft/WindowsApps/python3.13.exe c:/Users/Sathwik/OneDrive/Desktop/AI-Lab/lab1.py
0 1 1 2 3 5 8 13 21 34
Enter the number of terms in Fibonacci series: 10
0 1 1 2 3 5 8 13 21 34
Enter the number of terms in Fibonacci series: 
```