

ASSIGNMENT-1.3

Name: Bandana Giri

HT. No: 2303A51590

Batch: 07

Lab 1: Environment Setup – GitHub Copilot and VS Code Integration +
Understanding AI-assisted Coding Workflow

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

```
1  #Q1. Write a Python program to print Fibonacci series up to n terms
2  # Take user input
3  # Do not use functions
4  n = int(input("Enter number of terms: "))
5
6  a = 0
7  b = 1
8
9  if n <= 0:
10     print("Please enter a positive number")
11 elif n == 1:
12     print(a)
13 else:
14     print("Fibonacci sequence:")
15     print(a, b, end=" ")
16     for i in range(2, n):
17         c = a + b
18         print(c, end=" ")
19         a = b
20         b = c
```

```
Enter number of terms: 7
Fibonacci sequence:
0 1 1 2 3 5 8
```

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

```
1  # Q2.Optimize this Fibonacci code
2  # Simplify logic and variable usage
3  n = int(input("Enter number of terms: "))
4
5  a, b = 0, 1
6
7  if n <= 0:
8     print("Please enter a positive number")
9  else:
10     print("Fibonacci sequence:")
11     for _ in range(n):
12         print(a, end=" ")
13         a, b = b, a + b
```

```
Enter number of terms: 5
Fibonacci sequence:
0 1 1 2 3
```

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

```
1  # Q3.Create a function to generate Fibonacci series up to n
2  # Use meaningful comments
3  def fibonacci(n):
4      """
5      Generates Fibonacci sequence up to n terms
6      """
7      a, b = 0, 1
8      sequence = []
9
10     for _ in range(n):
11         sequence.append(a)
12         a, b = b, a + b
13
14     return sequence
15
16
17 n = int(input("Enter number of terms: "))
18 result = fibonacci(n)
19 print("Fibonacci sequence:", result)
```

```
Enter number of terms: 6
Fibonacci sequence: [0, 1, 1, 2, 3, 5]
```

Task 4: Comparative Analysis – Procedural vs Modular Fibonacci Code

```
1  #Q4 : Comparative Analysis - Procedural vs Modular Fibonacci Code
2  n = int(input("Enter number of terms: "))
3  a, b = 0, 1
4  if n <= 0:
5      print("Please enter a positive number")
6  elif n == 1:
7      print(a)
8  else:
9      print("Fibonacci series:")
10     print(a, end=" ")
11     print(b, end=" ")
12     for i in range(2, n):
13         c = a + b
14         print(c, end=" ")
15         a = b
16         b = c
```

```
Enter number of terms: 6
Fibonacci series:
0 1 1 2 3 5
```

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

```

1  #Q5.AI-Generated Iterative vs Recursive Fibonacci Approaches (Different
   Algorithmic Approaches for Fibonacci Series)
2  ✓def fibonacci_iterative(n):
3      |   a, b = 0, 1
4      ✓   for _ in range(n):
5          |   |   print(a, end=" ")
6          |   |   a, b = b, a + b
7
8
9      n = int(input("Enter number of terms: "))
10     fibonacci_iterative(n)
11  ✓def fibonacci_recursive(n):
12  ✓   |   if n <= 1:
13      |   |   return n
14      |   return fibonacci_recursive(n - 1) + fibonacci_recursive(n - 2)
15
16
17     n = int(input("Enter number of terms: "))
18  ✓for i in range(n):
19     |   print(fibonacci_recursive(i), end=" ")

```

Enter number of terms: 6

0 1 1 2 3 5 Enter number of terms: 3

0 1 1