

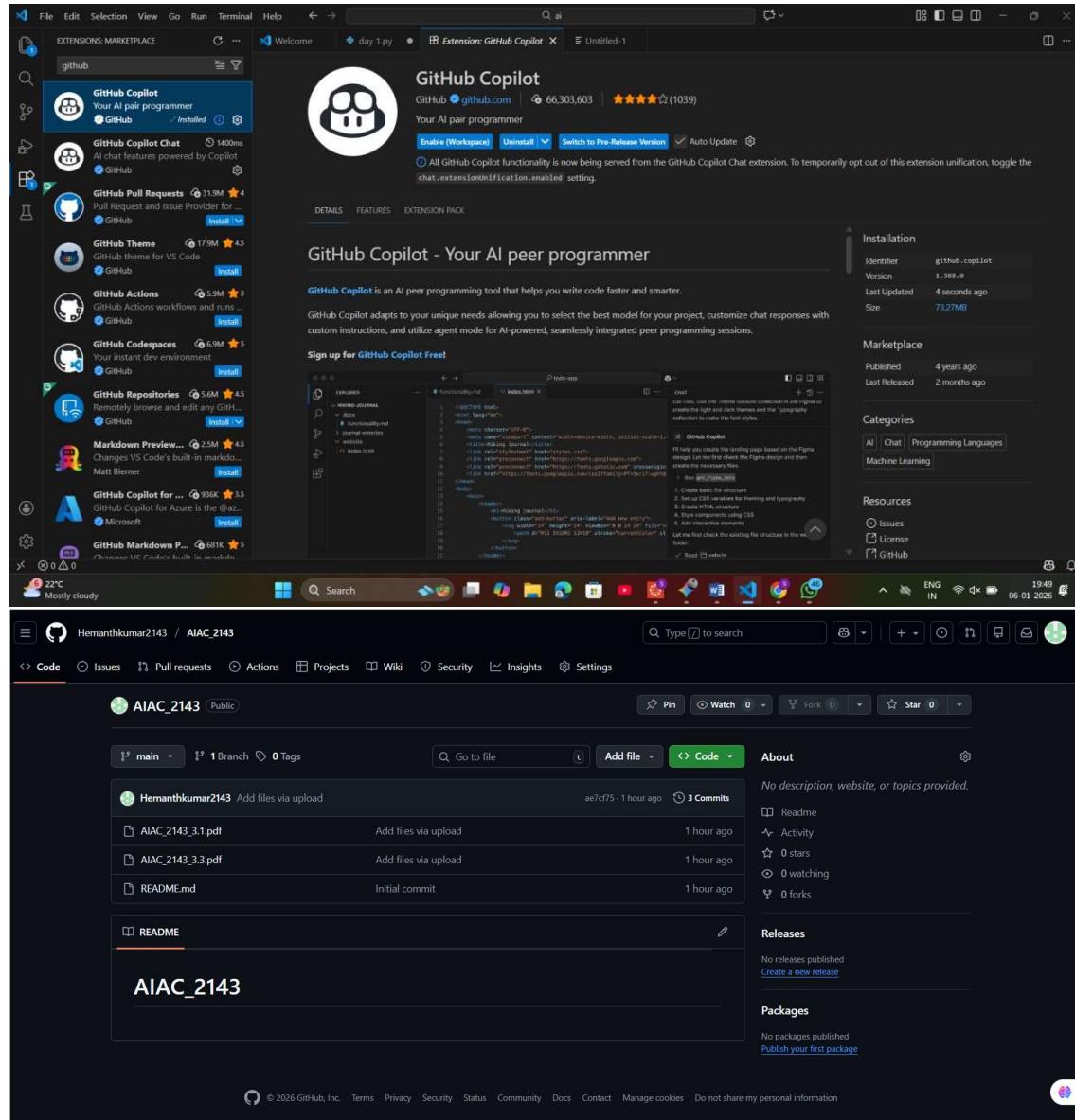
Name : P. Hemanth kumar

2303A52143

Batch - 32

AI Assisted Coding

Task 0: Environment Setup:-



Task 1: Non-Modular Logic (Factorial):-

```

C:\> Users > hp > OneDrive > Desktop > ai > task1.py > ...
1 # Task 1: Procedural Factorial Implementation
2 num = int(input("Enter a number: "))
3 factorial = 1
4
5 if num < 0:
6     print("Factorial does not exist for negative numbers")
7 elif num == 0:
8     print("The factorial of 0 is 1")
9 else:
10    temp = num
11    while temp > 0:
12        factorial *= temp
13        temp -= 1
14    print(f"The factorial of {num} is {factorial}")

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

Enter number: 5
Result: 120
PS C:\Users\hp\OneDrive\Desktop\HPC> & C:/Users/hp/AppData/Local/Microsoft/WindowsApps/python3.13.exe c:/Users/hp/OneDrive/Desktop/ai/task1.py
PS C:\Users\hp\OneDrive\Desktop\HPC> Enter a number: 5
Factorial is: 120
PS C:\Users\hp\OneDrive\Desktop\HPC>

```

LN 14, COL 52 SPACES: 4 UTF-8 CRLF Python 3.13.9 (Microsoft Store) 1549 07-01-2026

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

PS C:\Users\hp\OneDrive\Desktop\HPC> & C:/Users/hp/AppData/Local/Microsoft/WindowsApps/python3.13.exe c:/Users/hp/OneDrive/Desktop/ai/task1.py
Enter a number: 5
Factorial is: 120
PS C:\Users\hp\OneDrive\Desktop\HPC>

```

LN 9, COL 20 SPACES: 4 LITE: 8 — CP

Task 2: AI Code Optimization:-

```

C:\> Users > hp > OneDrive > Desktop > ai > task1.py > ...
1 # Task 2: Optimized Factorial
2 num = int(input("Enter a number: "))
3 factorial = 1
4
5 for i in range(1, num + 1):
6     factorial *= i
7
8 print(f"Factorial: {factorial}")

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

PS C:\Users\hp\OneDrive\Desktop\HPC> & C:/Users/hp/AppData/Local/Microsoft/WindowsApps/python3.13.exe c:/Users/hp/OneDrive/Desktop/ai/task1.py
Enter a number: 5
Factorial: 120
PS C:\Users\hp\OneDrive\Desktop\HPC>

```

LN 8, COL 33 SPACES: 4 UTF-8 CRLF Python 3.13.9 (Microsoft Store) 1535 07-01-2026

Task 3: Modular Design:-

```

1 def calculate_factorial(n):
2     """Calculates the factorial of a given number iteratively."""
3     result = 1
4     for i in range(1, n + 1):
5         result *= i
6     return result
7
8 if __name__ == "__main__":
9     user_input = int(input("Enter number: "))
10    print(f"Result: {calculate_factorial(user_input)}")

```

CHAT

CLARIFICATION ON PYTHON CODE REQUESTS

python code

Asked for clarification on Python code request

I'd be happy to help! However, your request is unclear. Could you clarify what you'd like me to do?

For example:

- View existing code: Should I look at lab1.py or task1.py?
- Write new code: What should the code do?
- Fix/debug code: What problem needs fixing?
- Run code: Which file should I execute?
- Explain code: Which code would you like me to explain?

Please let me know what you're trying to accomplish!

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\hp\OneDrive\Desktop\HPC> ~C
PS C:\Users\hp\OneDrive\Desktop\HPC>
PS C:\Users\hp\OneDrive\Desktop\HPC> cd 'C:\Users\hp\OneDrive\Desktop\HPC' & 'C:\Users\hp\AppData\Local\Microsoft\WindowsApps\python3.13.exe' 'C:\Users\hp\.vscode\extensions\ms-python.python-2025.18.0-win32-x64\unbundled\libs\debug\launcher' '65e97' ... 'C:\Users\hp\OneDrive\Desktop\ai\task1.py'
Enter number: 5
Results: 120
PS C:\Users\hp\OneDrive\Desktop\HPC>

Ln 10, Col 56 Spaces: 4 UTF-8 CRLF Agent Auto 15:57 07-01-2026

Task 4: Comparative Analysis:-

Criteria	Procedural (Task 1 & 2)	Modular (Task 3)
Logic Clarity	Linear and straightforward for very small tasks but becomes "spaghetti code" as complexity grows.	High clarity; the mathematical logic is isolated from the input/output logic.
Reusability	None. To use the logic elsewhere, the code must be manually copied and pasted.	High. The function can be imported into other Python files or called multiple times in one script.
Debugging Ease	Difficult. Errors in logic are mixed with errors in user input handling.	Simple. You can test the function with specific values (Unit Testing) to ensure the math is correct.
Project Suitability	Suitable only for small, one-off scripts or prototypes.	Essential for enterprise-level, large-scale software development.
AI Dependency Risk	High. AI might generate redundant variables or inefficient loops in long scripts.	Low. AI is highly specialized and accurate when asked to write specific, single-purpose functions.

Task 5: Iterative vs Recursive Thinking:-

The screenshot shows the Microsoft Visual Studio Code (VS Code) interface. The Explorer sidebar on the left shows a folder named 'HPC' containing a file 'lab1.py'. The main editor area displays two Python functions: 'factorial_iterative' and 'factorial_recursive'. The terminal at the bottom shows the execution of the code, entering '5' and displaying the result '120'. A Chat window on the right has a message from 'python code' asking for clarification on a Python code request. The status bar at the bottom indicates the file is 'task1.py', the language is 'Python', and the version is '3.13.9 (Microsoft Store)'.

```
C:\> Users > hp > OneDrive > Desktop > ai > task1.py > factorial_iterative
1 def factorial_iterative(n):
2     res = 1
3     for i in range(2, n + 1):
4         res *= i
5     return res
6
7 def factorial_recursive(n):
8     if n == 0 or n == 1:
9         return 1
10    return n * factorial_recursive(n - 1)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
```

Enter number: 5
Result: 120

```
PS C:\Users\hp\OneDrive\Desktop\HPC> ^C
PS C:\Users\hp\OneDrive\Desktop\HPC>
PS C:\Users\hp\OneDrive\Desktop\HPC> cd "c:\Users\hp\OneDrive\Desktop\HPC" & "c:\Users\hp\AppData\Local\Microsoft\WindowsApps\python3.13.exe" "c:\Users\hp\vscode\extensions\ms-python.python.debugpy-2025.18.0\python\bin\debug\lib\debug\launcher" "62630" -- "c:\Users\hp\OneDrive\Desktop\ai\task1.py"
```

task1.py +

Describe what to build next

Agent: Auto 15:40 ENG IN 07-01-2026