****

**AI ASSISTED CODING**

**ASSIGNMENT-10.4**

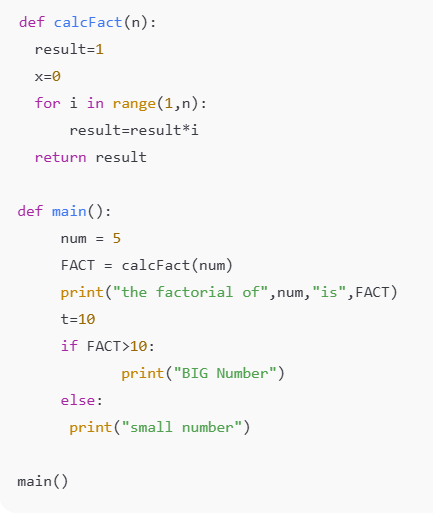
**NAME:**BATHINI SAHASRA

**ROLL NO:**2503A51L07

**BATCH:** 24BTCAICSB19

**Task Description#1 AI-Assisted Code Review (Basic Errors)**

* Write python program as shown below.
* Use an AI assistant to review and suggest corrections



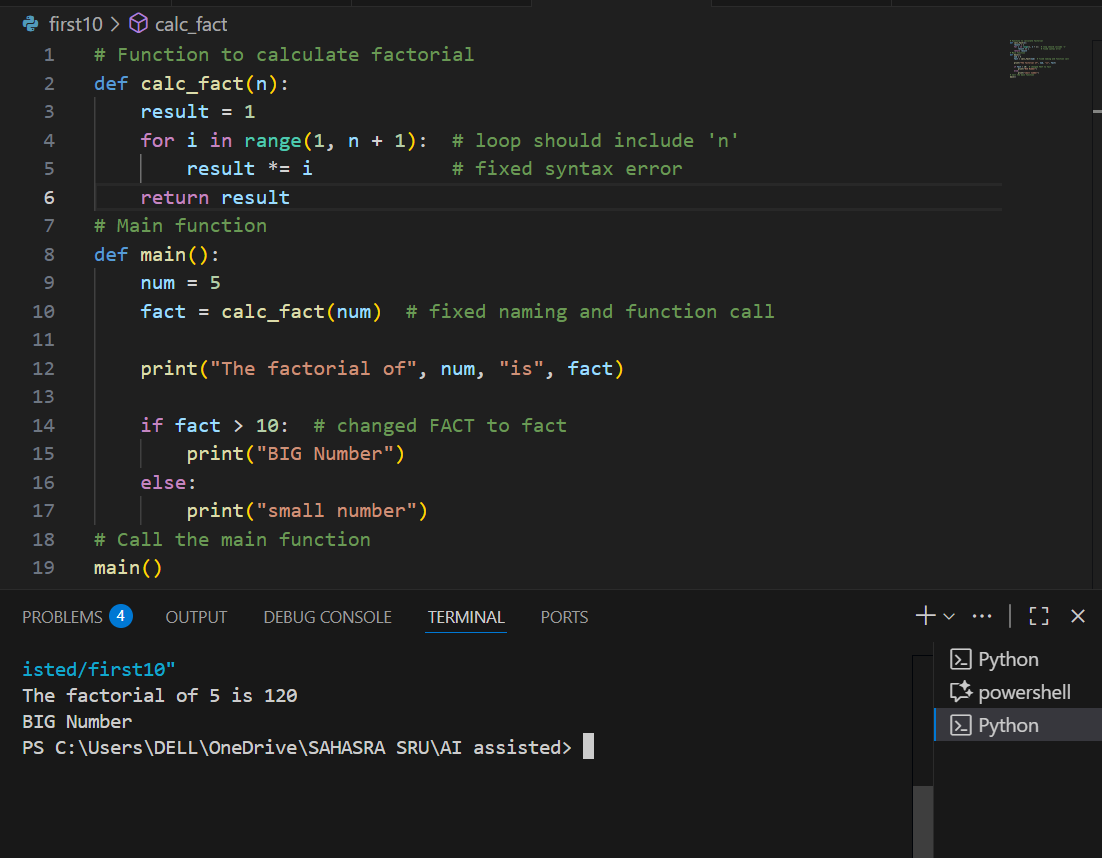
**Expected Outcome#1:** Students need to submit corrected code with comments.

**PROMPT:**Review the above code and suggest corrections.

rewrite the correct code with comments.

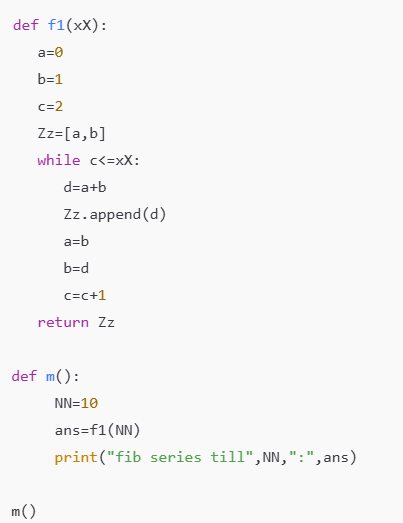
**Corrections made in the code:**

* The loop for i in range(1, n): is incorrect. It should be range(1, n + 1) to include n itself.
* Variable x = 0 in calcFact() is never used.
* In main(), num is defined, but passed to calcFact() as num, yet the function is called before num is declared as a parameter.
* The variable t = 10 is declared but unused.
* The if FACT > 10: is valid, but t being declared above has no effect.
* Use proper naming conventions (e.g., fact instead of FACT).
* Ensure consistent indentation and spacing for readability.



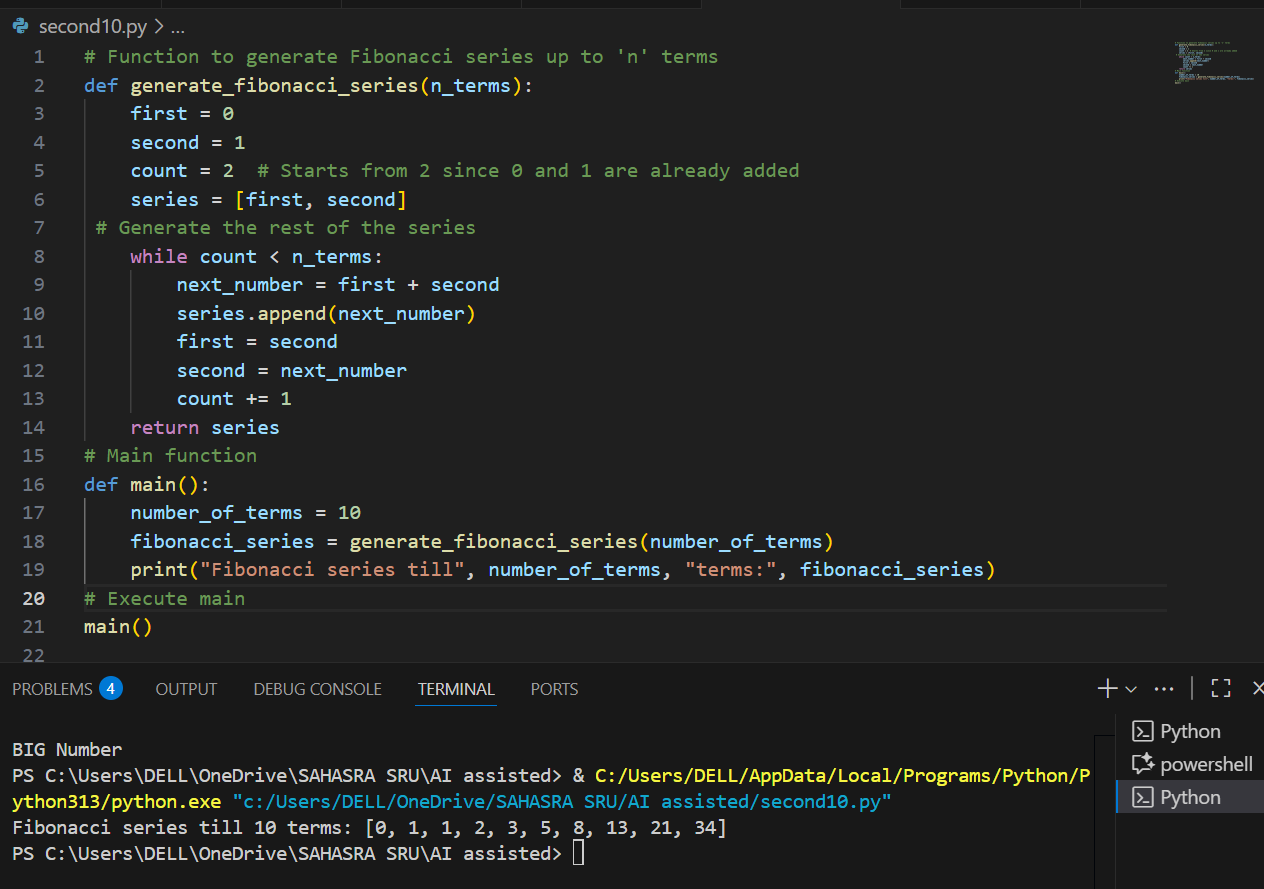
**Task Description#2 Automatic Inline Comments**

* Write the Python code for Fibonacci as shown below and execute.
* Ask AI to improve variable names, add comments, and apply PEP8 formatting (cleaned up).
* Students evaluate which suggestions improve readability most. one.



**Expected Output#2:** Clean format python code with much readability.

**PROMPT:** improve variable names, add comments, and apply PEP8 formatting (cleaned up) for the above python code

****

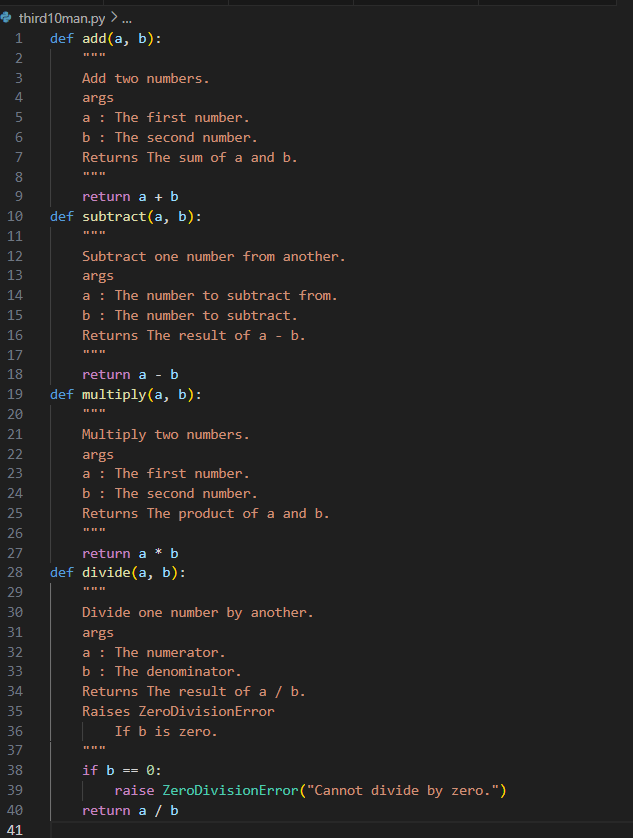
**Key Improvements:**

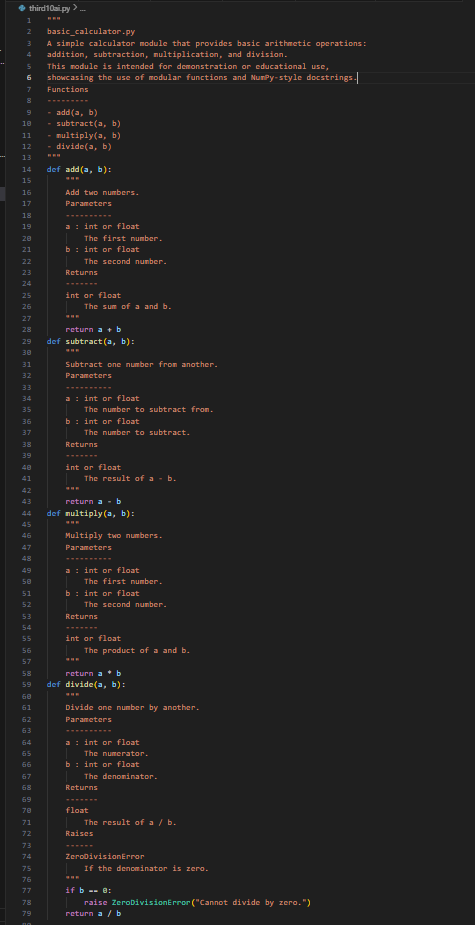
* Replacing a, b, c, Zz with first, second, count, series makes the logic self-explanatory.
* Function name generate\_fibonacci\_series tells you exactly what the function does.
* Original used while c <= xX, which actually **generates xX + 1 terms**, but your corrected code uses while count < n\_terms, which properly limits the list to **exactly n\_terms values**.
* Each logical block now has a comment — helps learners or future readers quickly understand what the code is doing.
* PEP8 style: consistent spacing, indentation, and line length.
* Your code looks professional and beginner-friendly now.

**Task Description#3**

* Write a Python script with 3–4 functions (e.g., calculator: add, subtract, multiply, divide).
* Incorporate manual **docstring** in code with NumPy Style
* Use AI assistance to generate a module-level docstring + individual function docstrings.
* Compare the AI-generated docstring with your manually written one.

MANUAL CODE:



****

OBSERVATION:

 **AI-assisted docstrings** offer:

* Professional structure
* Better readability
* Tool compatibility
* Reusability in teams, APIs, and documentation websites

 ️ **Manual comments** are a good starting point for beginners, but lack structure and formal style, which may be insufficient for large or shared codebases.

The **manual comments** provide basic understanding but lack structure, standard formatting, and detailed type information. In contrast, the **AI-assisted docstrings** follow the **NumPy documentation standard**, offering clearer structure, better readability, and compatibility with professional tools. This makes them more suitable for collaborative and scalable projects.