## K.SHIVASAI

## 2403a51386

## 24/09/25

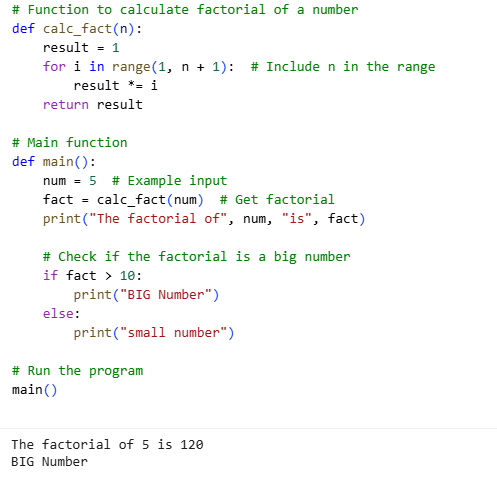
## ASSIGENMENT 10.2

## TASK #1

AI-Assisted Code Review (Basic Errors)  
• Write python program as shown below.  
• Use an AI assistant to review and suggest corrections.

PROMPT:

* Write a Python function to calculate the factorial of a non-negative integer.
* Modify the given factorial code to handle negative inputs and return an appropriate message.
* Create a Python script that calculates factorials using recursion.
* Write a function that takes a number and checks if it is a factorial of any integer.



OUTPUT:

Screenshot 2025-09-24 122015.png

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

OBSERVATION:

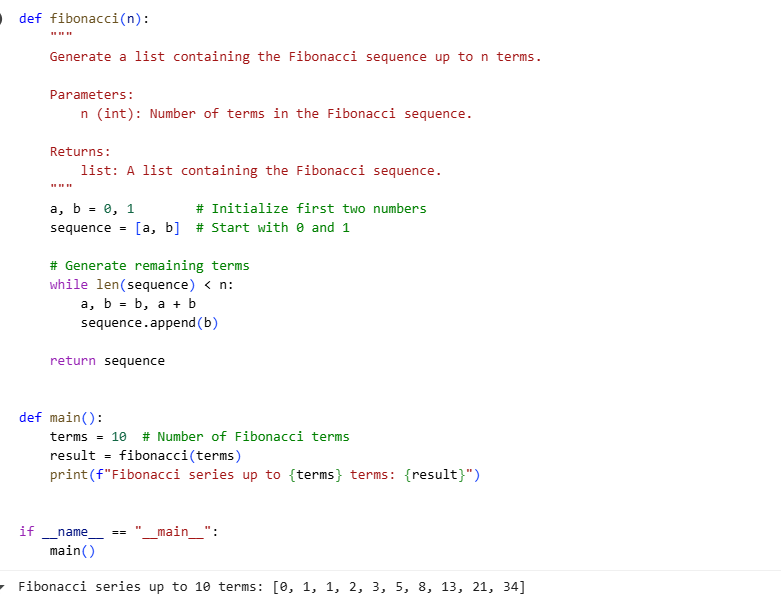
->The function calc\_fact(n) correctly computes the factorial by multiplying numbers from 1 to n.

->The function and variables (calc\_fact, result, num, fact) are descriptive and improve code readability

->The program uses a fixed input num = 5 to demonstrate functionality, which makes it easy to test.Task Description#2  
• 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.

PROMPT:

* Write a Python function that generates the Fibonacci sequence up to a specified number *n*.
* "Modify the given Fibonacci code to return the sequence as a generator instead of a list.
* Create a Python script that calculates and prints the nth Fibonacci number using a recursive approach.
* Write a function that takes a number *n* and checks if it is a Fibonacci number.



OUTPUT:

Screenshot 2025-09-24 122028.png

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

OBSERVATION:

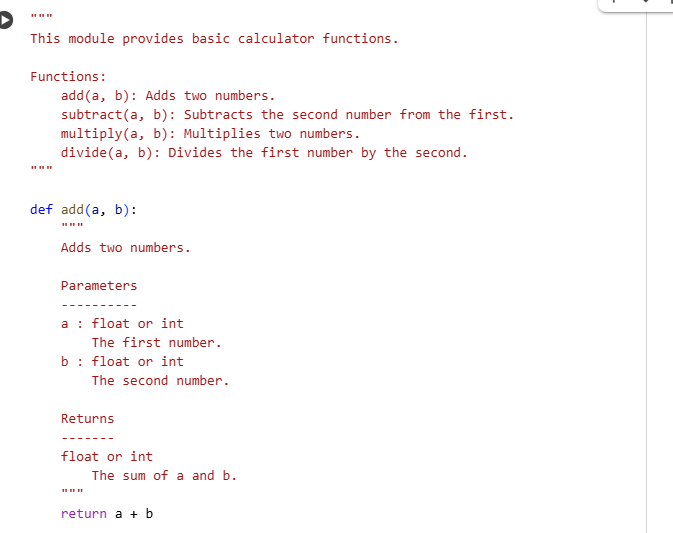
-> The code is relatively easy to read and follow, with clear variable names (a, b, sequence, n, terms, result) and comments explaining the initialization.

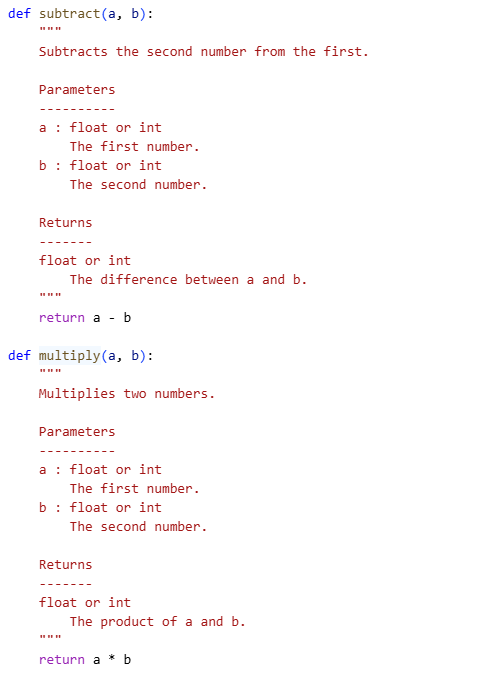
 ->The function fibonacci has a good docstring that explains its purpose, parameters, and return value, making it easy for someone else to understand how to use it.

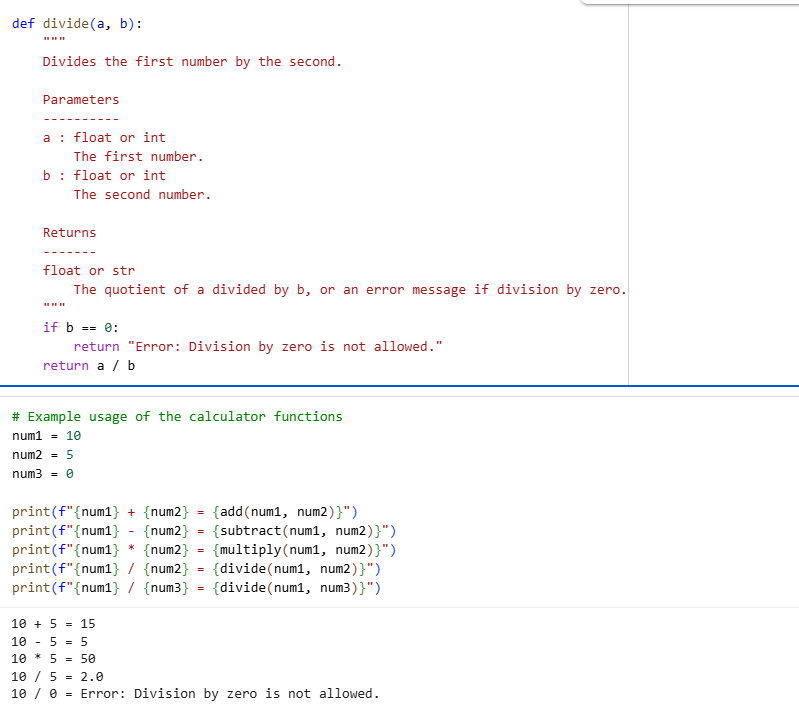
-> The use of a main() function and the if \_\_name\_\_ == "\_\_main\_\_": block is good practice, allowing the script to be run directly or imported as a module.

Task Description#3

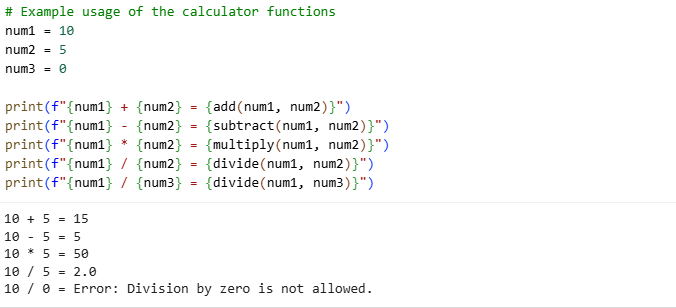
• Write a Python script with 3–4 functions (e.g., calculator: add, subtract, multiply,  
divide).  
• Incorporate manual docstring in code with NumPy Style  
• Use AI assistance to generate a module-level docstring + individual function  
docstrings.  
• Compare the AI-generated docstring with your manually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







OUTPUT:



OBSERVATION:

->Each function uses **NumPy-style docstrings**, which is a professional practice in Python projects for clear documentation.

->The divide function properly checks for division by zero, which is a common Python runtime error.

->Functions are independent, making the module reusable in other Python programs or importable into larger projects.