AI ASSISTED CODING

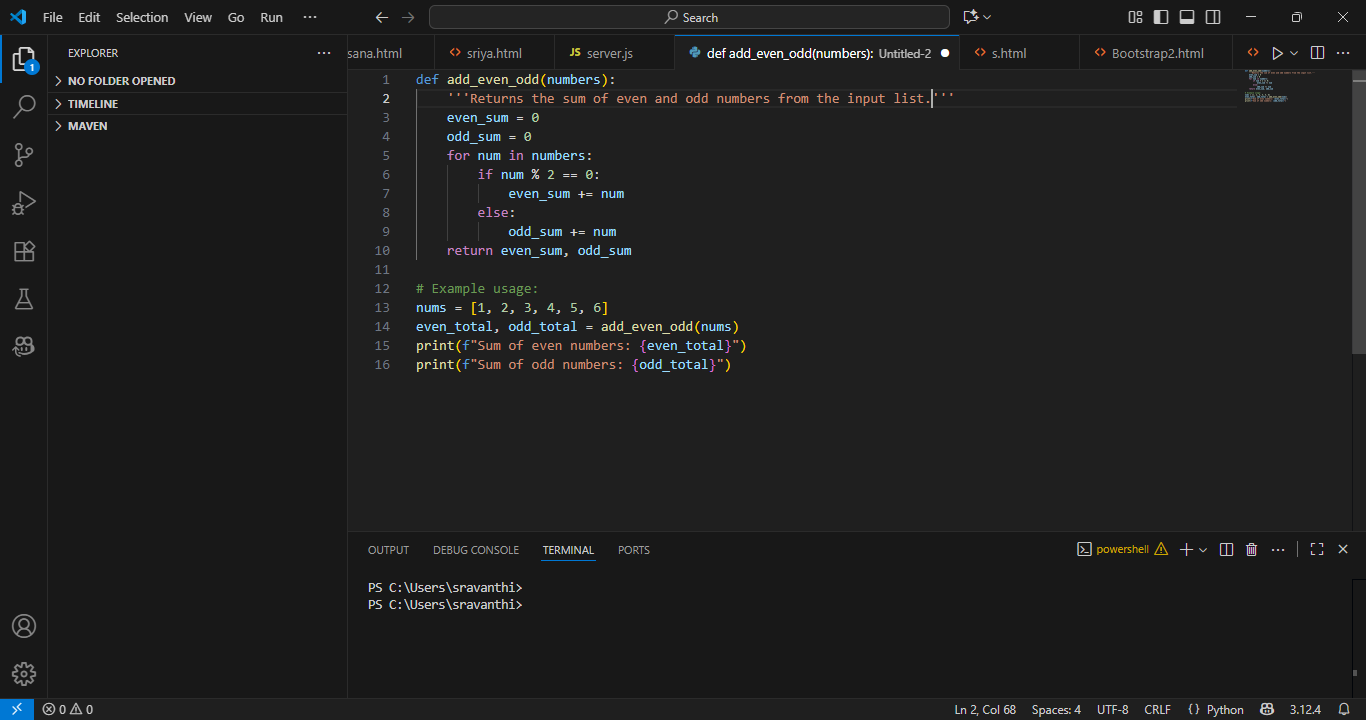
NAME : DARIPELLI SRIYA

ENROLLMENT NO : 2403A51350

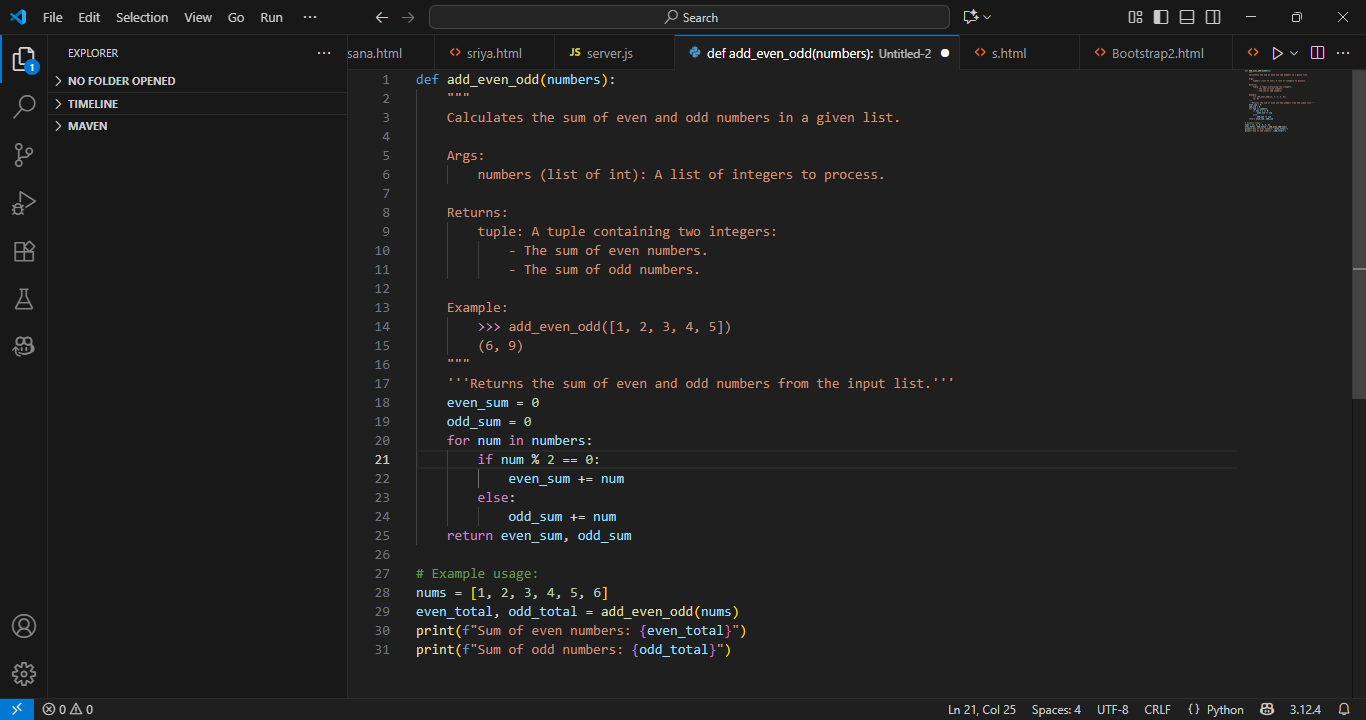
BATCH NO: 01

Task Description#1 Basic Docstring Generation  
• Write python function to return sum of even and odd numbers in the given list.  
• Incorporate manual docstring in code with Google Style  
• Use an AI-assisted tool (e.g., Copilot, Cursor AI) to generate a docstring describing  
the function.  
• Compare the AI-generated docstring with your manually written one.  
Expected Outcome#1: Students understand how AI can produce function-level documentation.

MY CODE: Generate code for adding odd and even num separately.



PROMPT: Generate doc for the code



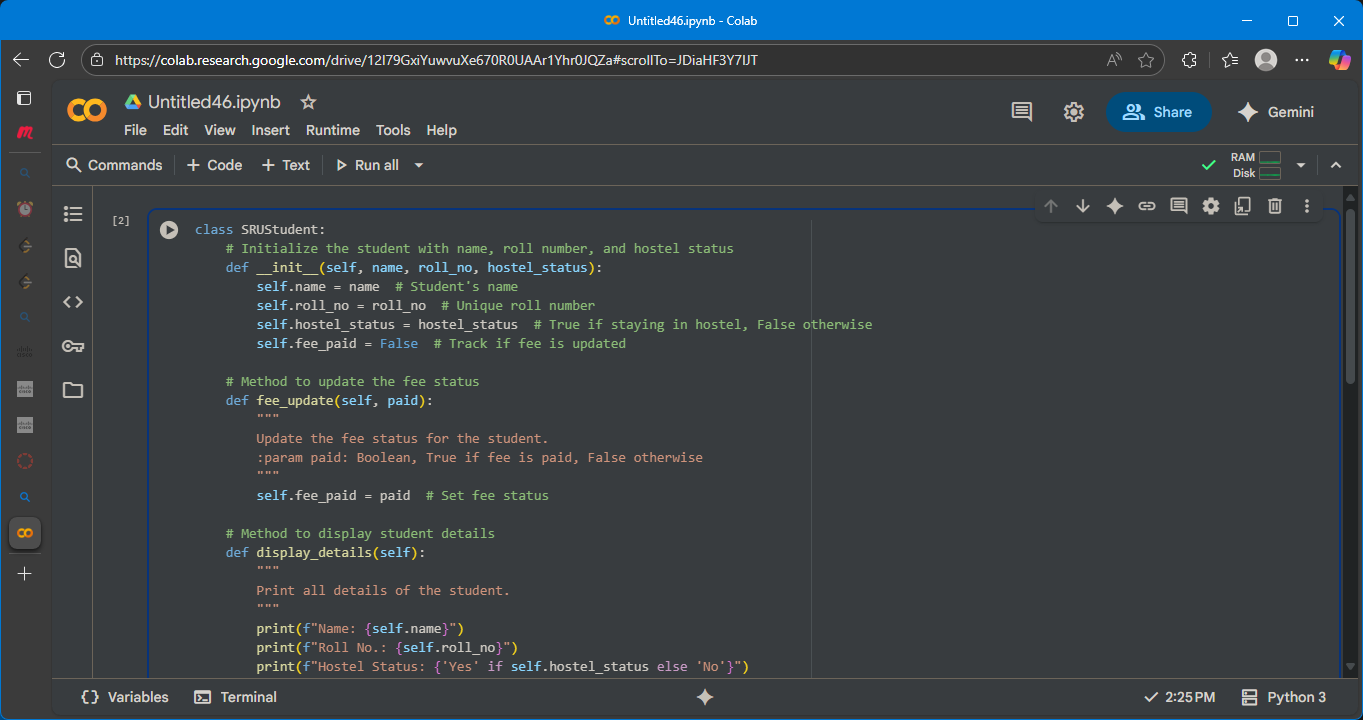
OBSERVATION:

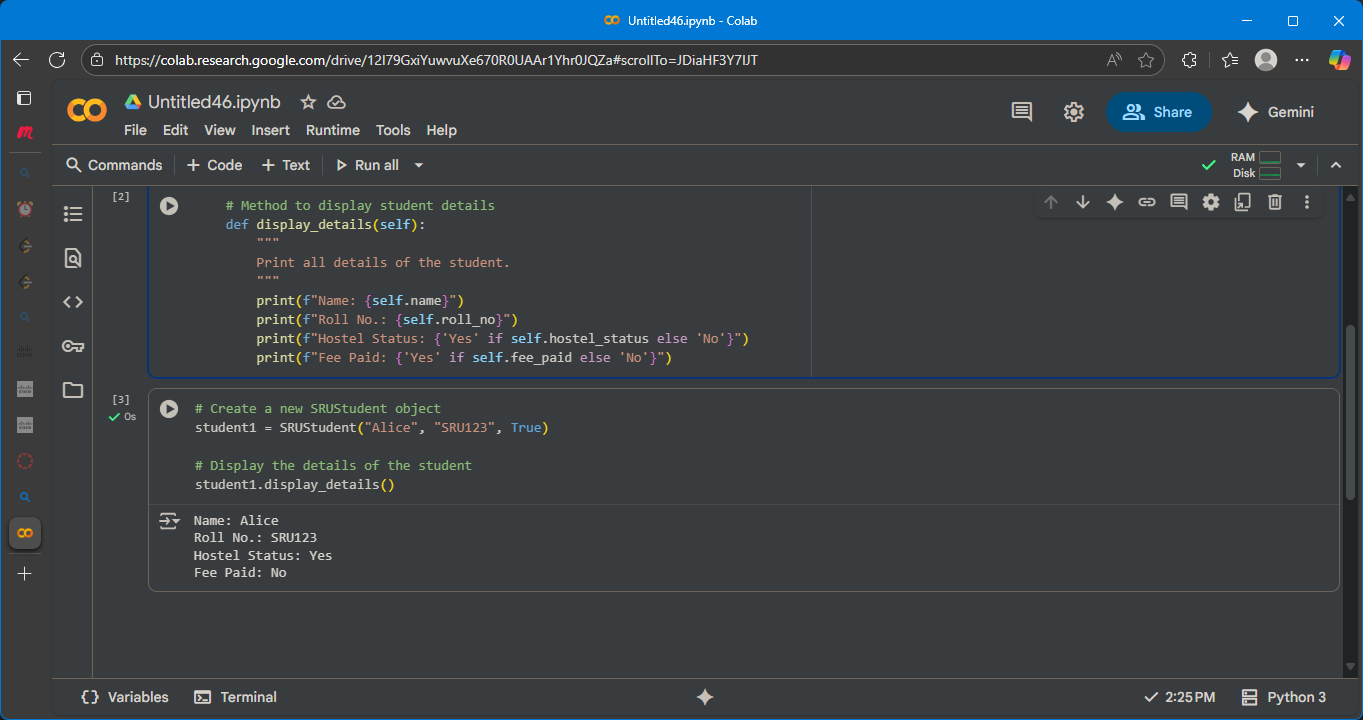
Docstring by ai is far more better than mine code ,It just explained the code very clearly each line so even a non coders or non programmers can easily understand the code.

Task Description#2 Automatic Inline Comments  
• Write python program for sru\_student class with attributes like name, roll no.,  
hostel\_status and fee\_update method and display\_details method.  
• Write comments manually for each line/code block  
• Ask an AI tool to add inline comments explaining each line/step.  
• Compare the AI-generated comments with your manually written one.  
Expected Output#2: Students critically analyze AI-generated code comments

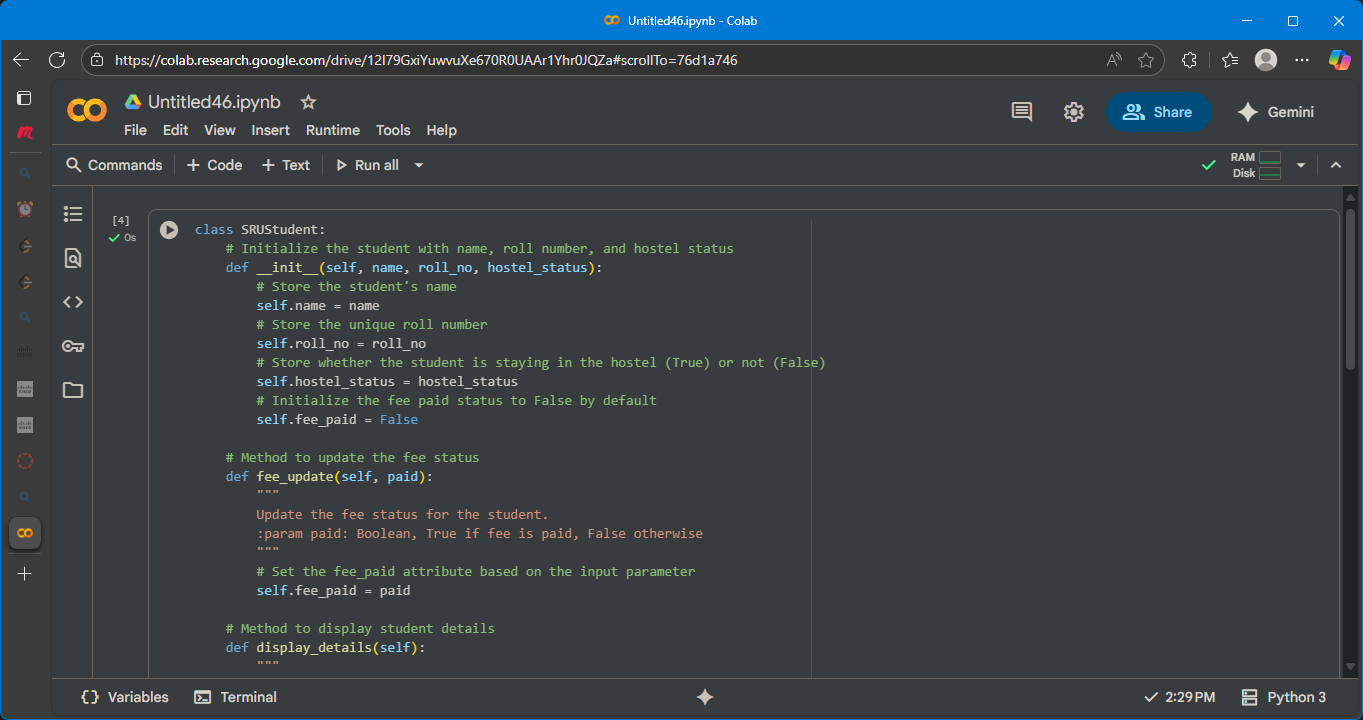
With my version:

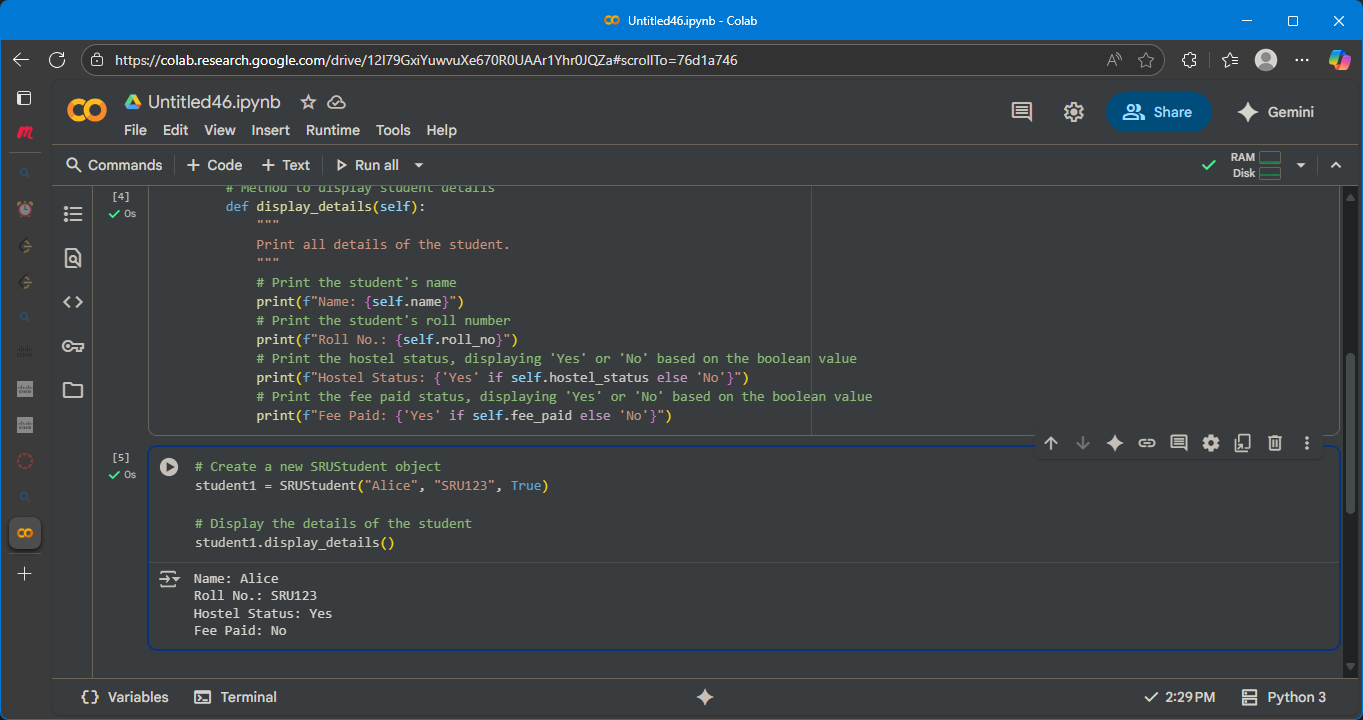
Generate python program for sru\_student class with attributes like name, roll no.,  
hostel\_status and fee\_update method and display\_details method and write comments like a human does





With ai version:





Observation:

Comments by ai is far more better than mine code It just explained the code very clearly each line so even a non coders or non programmers can easily understand the code.

**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

**PROMPT:**

Generate a python code to develop 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

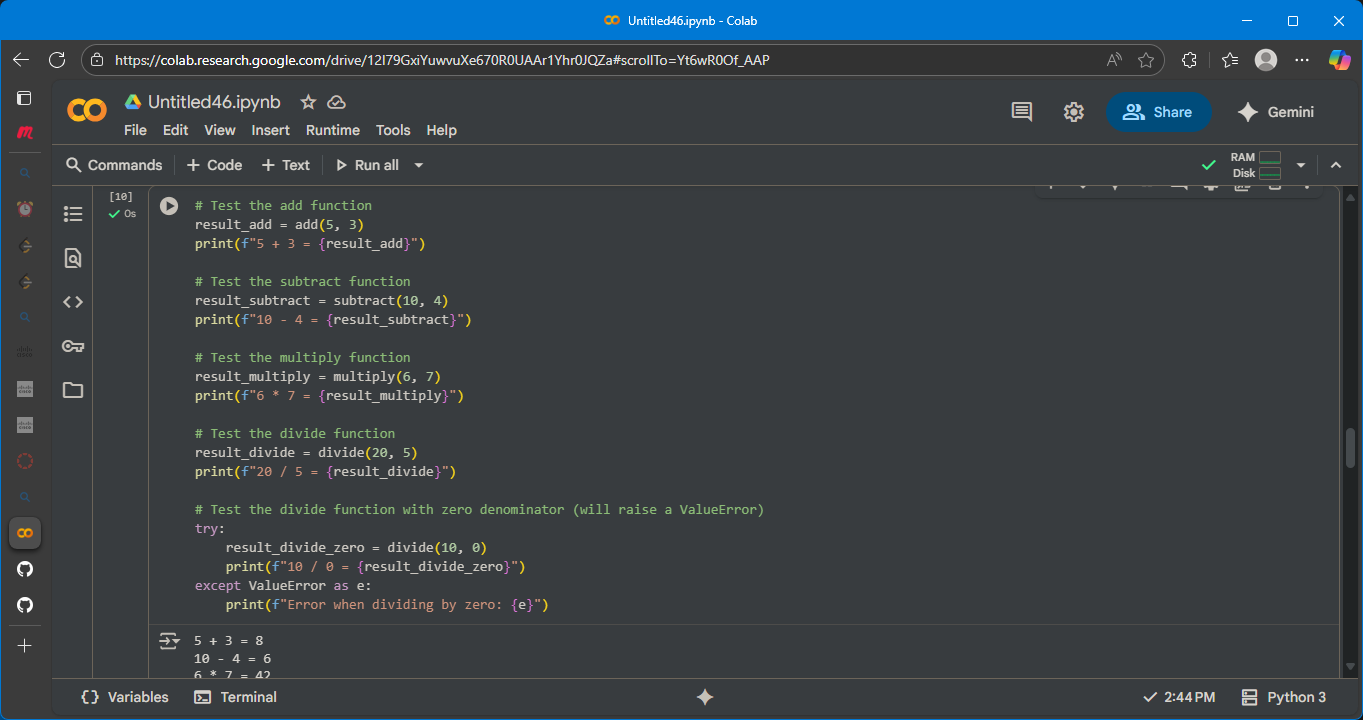
**Expected Output#3:**

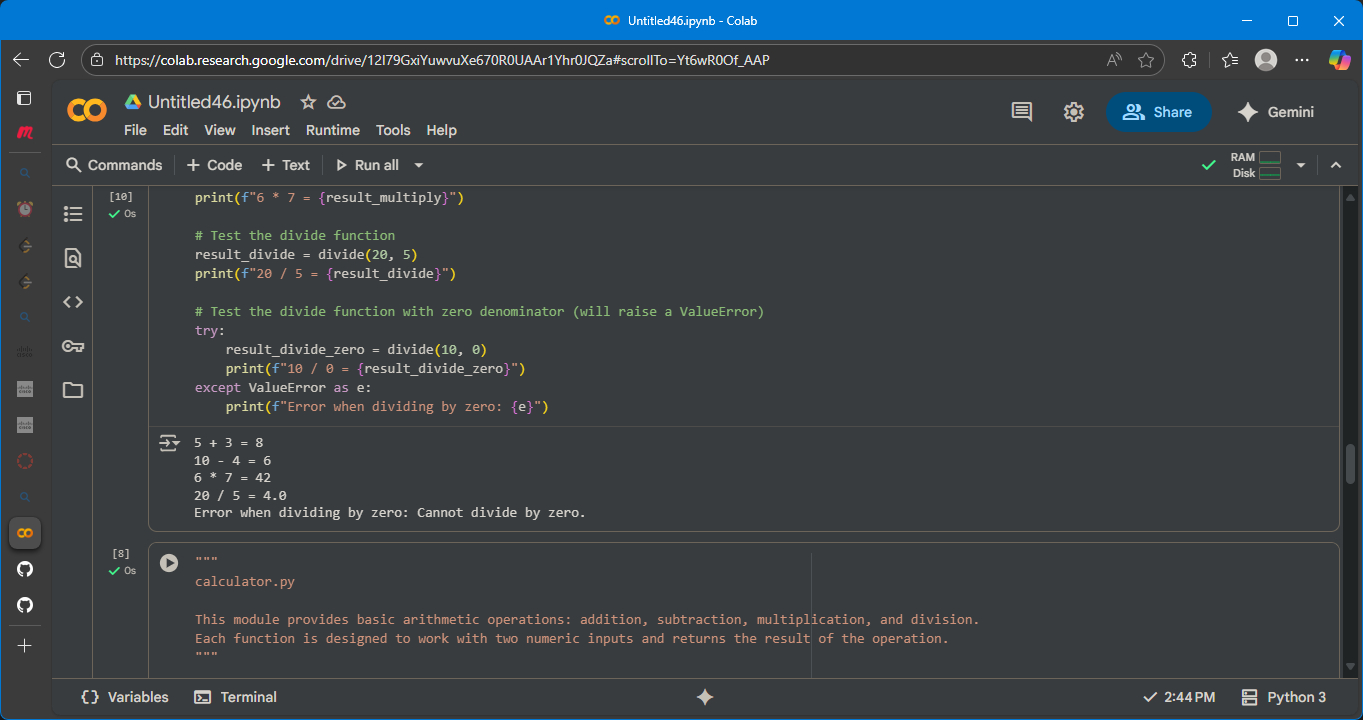
Students learn structured documentation for

multi-function scripts

Push documentation whole workspace as .md

file in GitHub Repository





Observation:

Certainly! The selected code defines a simple calculator module with four basic arithmetic functions:

* **add(a, b)**: This function takes two numbers, a and b, and returns their sum.
* **subtract(a, b)**: This function takes two numbers, a and b, and returns the result of subtracting b from a.
* **multiply(a, b)**: This function takes two numbers, a and b, and returns their product.
* **divide(a, b)**: This function takes two numbers, a and b, and returns the result of dividing a by b. It also includes error handling to raise a ValueError if the denominator b is zero, as division by zero is undefined.

The code also includes docstrings for each function, explaining what it does, its parameters, what it returns, and any potential errors it might raise.