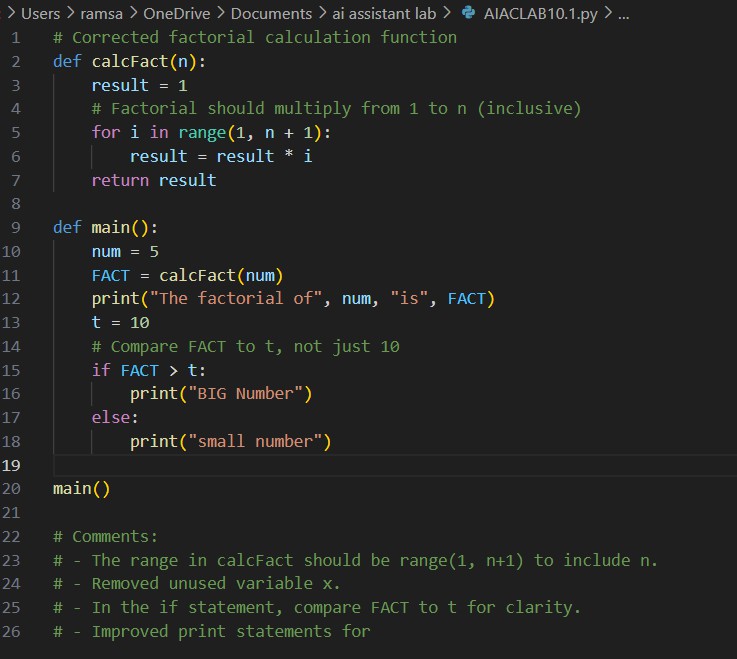
**ASSIGNMENT 10.2**

**NAME: N.Akash**

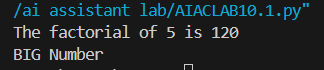
**ENROLLMENT ID: 2503A51L47 TASK 1:**

->Write python program as shown below.

->Use an AI assistant to review and suggest corrections



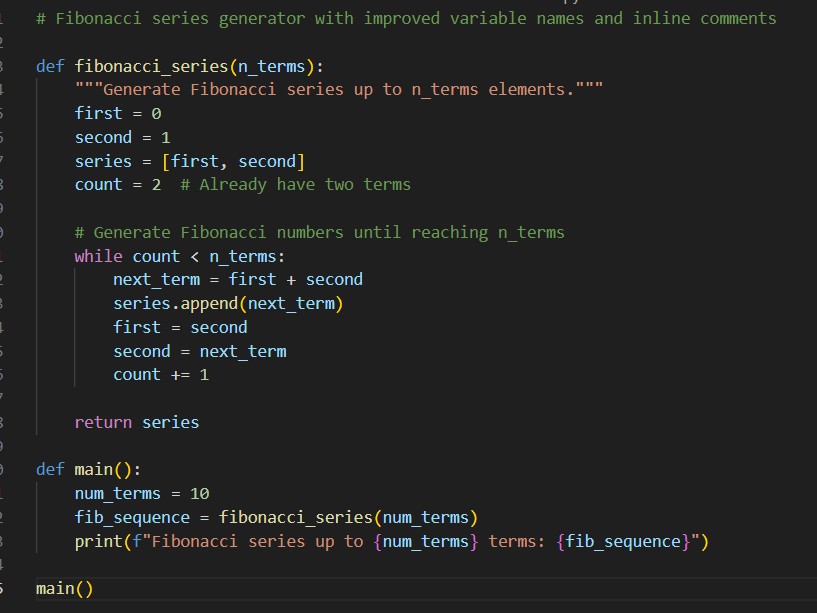
**OUTPUT:**

****

**TASK 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.



**OUTPUT:**

**A black background with white text  AI-generated content may be incorrect.**

**TASK 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.

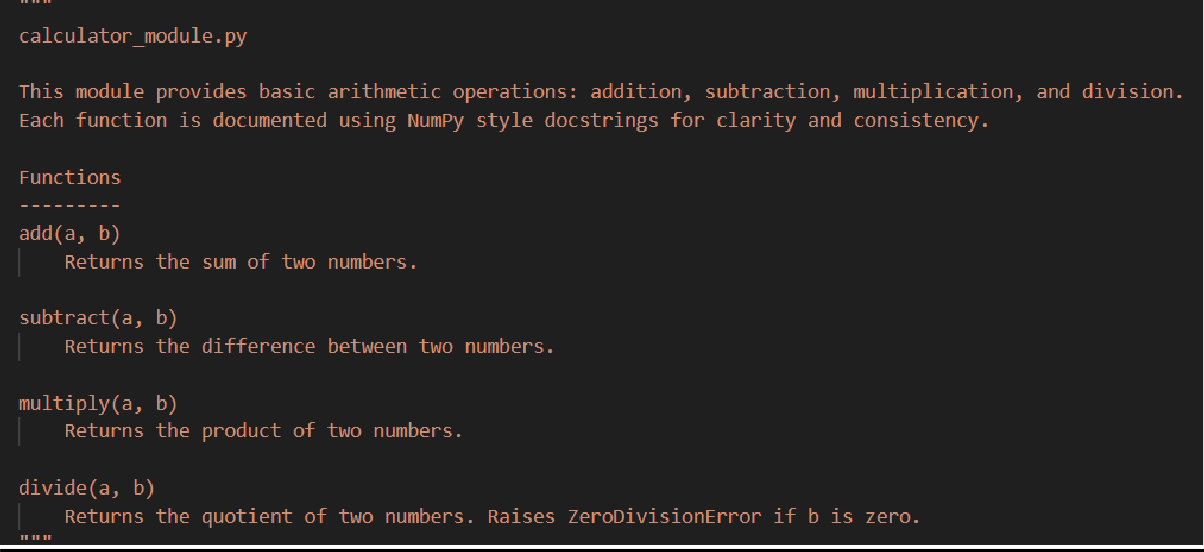
# Common Examples of Code Smells

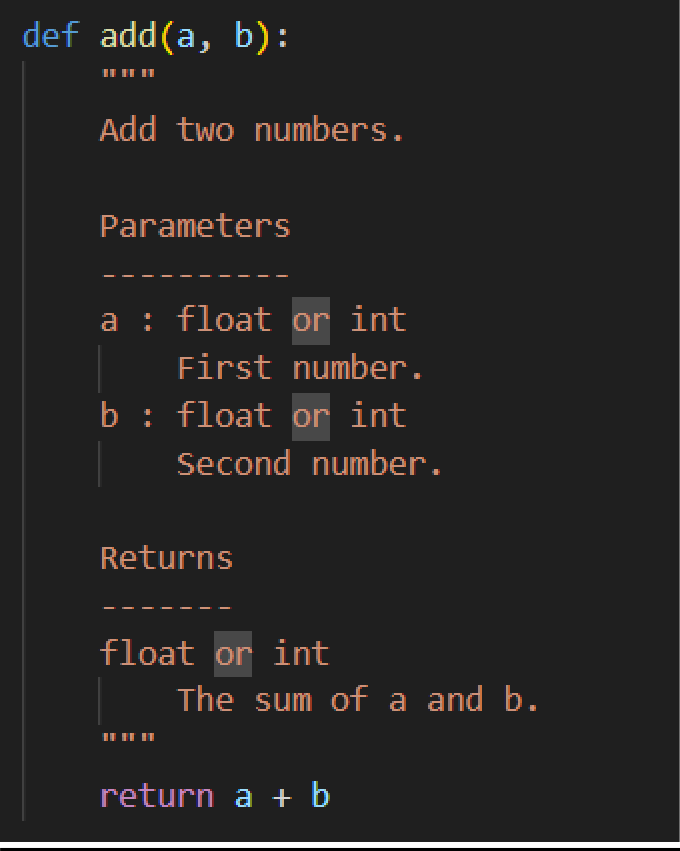
* Long Function – A single function tries to do too many things.
* Duplicate Code – Copy-pasted logic in multiple places.
* Poor Naming – Variables or functions with confusing names (x1, foo, data123).
* Unused Variables – Declaring variables but never using them.
* Magic Numbers – Using unexplained constants (3.14159 instead of PI).
* Deep Nesting – Too many if/else levels, making code hard to read.
* Large Class – A single class handling too many responsibilities.

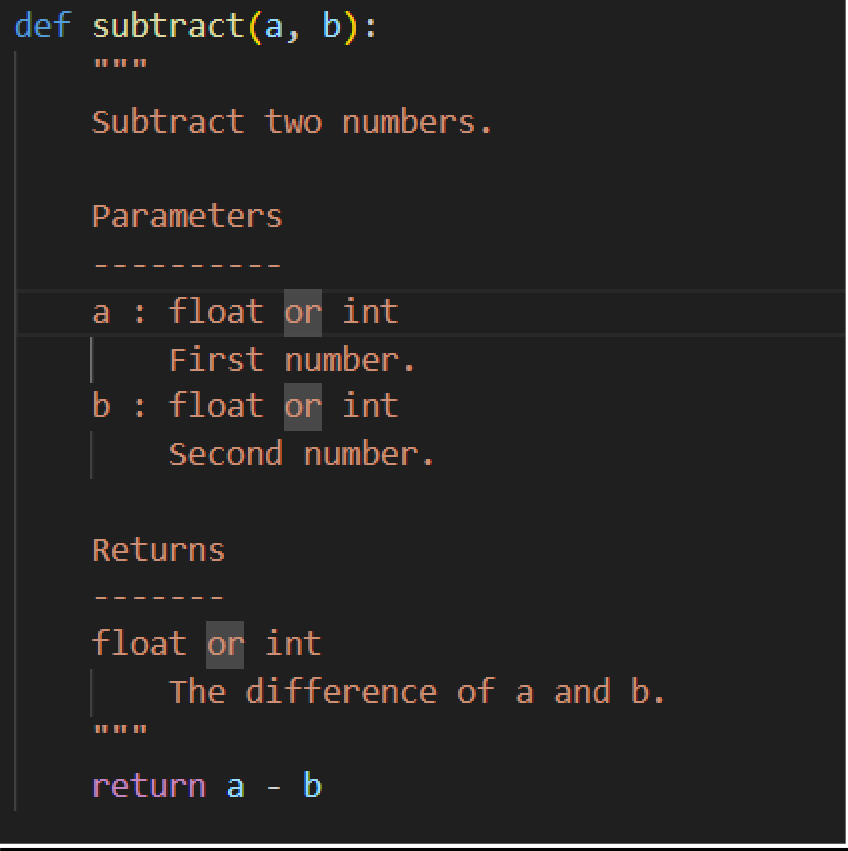
# Why Detecting Code Smells is Important

* Makes code easier to read and maintain.
* Reduces chance of bugs in future updates.
* Helps in refactoring (improving structure without changing behavior).
* Encourages clean coding practices

**CODE GENERATED:**

****



****

