LAB ASSIGNMENT – 9.1

Task Description #1 (Documentation – Google-Style Docstrings for  
Python Functions)

Original Script:

A screenshot of a computer program

AI-generated content may be incorrect.

**Observation:  
The given Python script contained multiple functions but lacked proper documentation in the form of docstrings. Although the functions performed basic tasks like greeting a user, adding numbers, squaring a value, and checking even numbers, their purpose was not immediately clear without reading the code logic. The absence of argument descriptions and return type information made the script less user-friendly, especially for beginners. Without examples, users would need to manually test functions to understand their usage. This reduces readability and makes collaboration difficult in larger projects. The lack of standardized documentation also prevents automatic documentation generation using tools. Overall, the script worked correctly but lacked clarity, maintainability, and professional presentation due to missing docstrings.**

Script after adding Google-Style Docstrings for Python Functions :

A screenshot of a computer program

AI-generated content may be incorrect.

A black screen with white text

AI-generated content may be incorrect.

**Observation:**  
The Python script was updated with Google-style docstrings for all functions. Each function now has a clear description, argument details with type hints, and return type information. Example usages were also added, showing how the functions can be called in practice. This improves readability and helps users quickly understand the code. Type hints further enhance clarity and support error detection. Testing verified that the examples match the actual outputs of the functions. The standardized documentation style makes the script more professional and user-friendly. Overall, adding docstrings improves maintainability and usability of the code.

Task Description #2 (Documentation – Inline Comments for Complex Logic)

Without inline comments:

A screenshot of a computer program

AI-generated content may be incorrect.

With inline Comments:

A screenshot of a computer program

AI-generated content may be incorrect.

**Observation:**  
The given Python program checks whether a string is a palindrome using recursion. The string is first normalized by converting it to lowercase and removing spaces, ensuring uniform comparison. The base case verifies that single-character or empty strings are palindromes. The program then compares the first and last characters of the string. If they are equal, the recursion continues with the substring excluding these characters. If they differ, the function immediately returns False. This recursive approach simplifies the problem by breaking it into smaller subproblems. Overall, the program successfully demonstrates recursion, string handling, and conditional logic in Python.

Task Description #3 (Documentation – Module-Level Documentation)

A screen shot of a computer

AI-generated content may be incorrect.

Python file link:

<https://drive.google.com/file/d/19sJSHdGdwkfbRv5FsjyKC9ivEynS79UW/view?usp=drive_link>

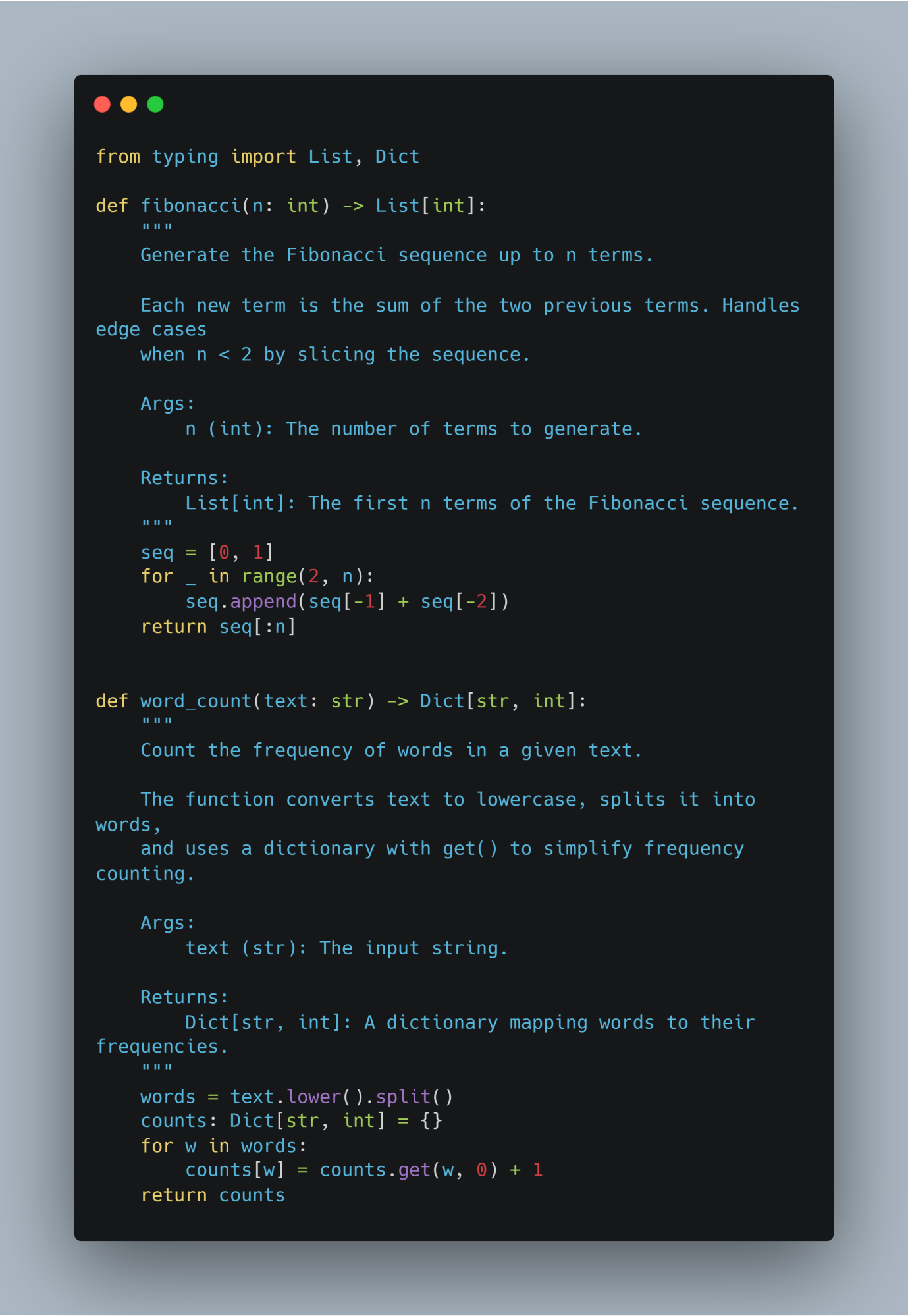
**Observation:**  
The program implements a Student Management System using object-oriented programming. The Student class handles details such as name, roll number, marks, and grade calculation. The StudentManagementSystem class manages multiple students by adding, searching, and displaying records. It demonstrates encapsulation and modular design for clarity and reusability. The program runs without external dependencies, relying only on Python’s standard features. Overall, it provides a structured and efficient way to manage student information.

**Task Description #4** (Documentation – Convert Comments to Structured Docstrings)

* Task: Use AI to transform existing inline comments into structured function docstrings following Google style.

Original code:





Observation:

Turning the comments into docstrings made the code is cleaner and easier to read. Now the purpose and details of each function are right at the top, so you don’t have to dig through the code to understand it. It feels more beginner-friendly and organized

Task Description #5 (Documentation – Review and Correct Docstrings)

Code with error:

A screen shot of a computer program

AI-generated content may be incorrect.

Code corrected by A.I:

A screen shot of a computer program

AI-generated content may be incorrect.

Observation:  
The program defines a Student class to manage student details and performance. It stores attributes such as name, roll number, and marks. The calculate\_average() method computes the average of marks, while get\_grade() assigns a grade based on defined criteria. The use of methods improves code readability and reusability. The corrected docstrings now match the actual implementation, avoiding confusion. Overall, the class demonstrates object-oriented programming with clarity and accuracy.

**Task Description #6** (Documentation – Prompt Comparison Experiment)

Python function:

A black rectangular object with a black border

AI-generated content may be incorrect.

**Vague Prompt:**

Prompt:  
“Add comments to this function.”

A black rectangular object with white text

AI-generated content may be incorrect.

**Observations for Vague Prompt:**

* **Quality:** Very basic; only a single inline comment.
* **Accuracy:** Correctly states what the function does, but very minimal.
* **Completeness:** Lacks detail about parameters, return type, units, or examples. It may not help someone unfamiliar with Python or math concepts.

**Detailed Prompt:**

**Prompt:**  
“Add Google-style docstrings with parameters, return types, and examples.”

A screenshot of a computer

AI-generated content may be incorrect.

**Observations for Detailed Prompt:**

* **Quality:** High-quality documentation; structured, readable, and professional.
* **Accuracy:** Accurately describes the purpose, parameter type, return type, and provides an example.
* **Completeness:** Fully self-contained; someone unfamiliar with the function can understand how to use it. Includes units implicitly (area), parameter constraints, and example output.

**Comparison Table:**

A screenshot of a black screen

AI-generated content may be incorrect.

**Key Takeaways:**

1. Vague prompts produce minimal and sometimes insufficient documentation.
2. Detailed prompts guide AI to create professional, complete, and usable documentation.
3. For real projects, detailed prompts save time and reduce errors for future users or developers.