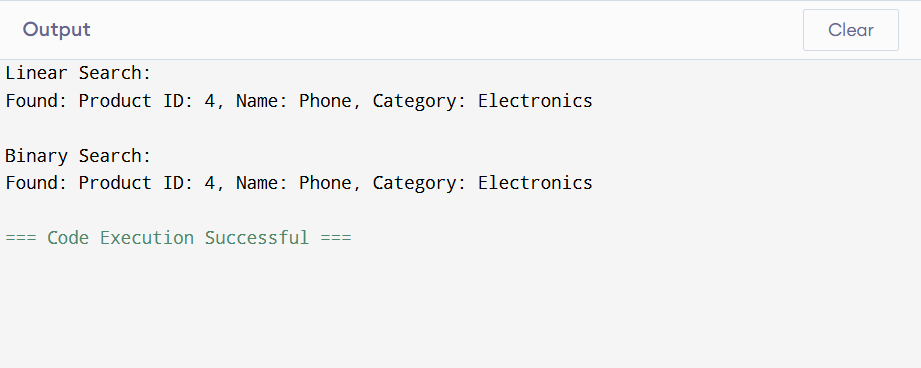
**MANDATORY HANDS-ON WEEK 1**

**ALGORITHMS DATA STRUCTURES(OUTPUT)**

**Exercise 2: E-commerce Platform Search Function**

****

**1. Understanding Asymptotic Notation**

**Big O Notation:**

Big O notation is used to describe the performance or complexity of an algorithm. It gives an upper bound on the time or space the algorithm takes as the input size increases. It helps in comparing algorithms in terms of scalability and efficiency.

**Search Scenarios:**

| Case | Linear Search | Binary Search |
| --- | --- | --- |
| Best Case | O(1) | O(1) |
| Average Case | O(n) | O(log n) |
| Worst Case | O(n) | O(log n) |

Linear search checks each element one by one, while binary search repeatedly divides the search space, which makes it more efficient for large, sorted datasets.

**4. Analysis**

Time Complexity Comparison:

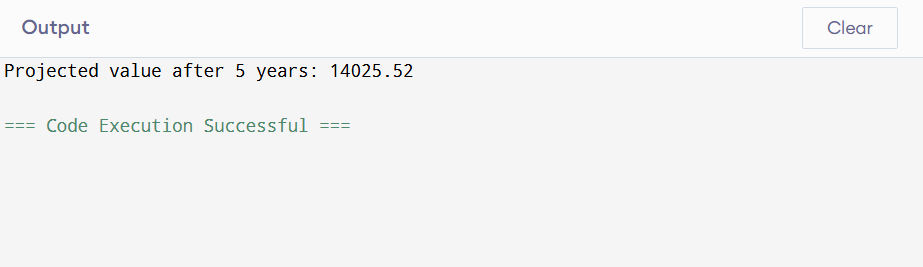
| Search Type | Best Case | Average Case | Worst Case |
| --- | --- | --- | --- |
| Linear Search | O(1) | O(n) | O(n) |
| Binary Search | O(1) | O(log n) | O(log n) |

Linear search is simple but slow for large data. Binary search is faster but requires data to be sorted beforehand.

**Conclusion:**

Linear search is suitable for small or unsorted product lists.  
Binary search is better for larger datasets that can be sorted by product name.  
In a real-world e-commerce platform where fast user response is important, binary search is the preferred option for efficiency.

**Exercise 7: Financial Forecasting**

****

**1. Understanding Recursive Algorithms**

Recursion is a programming technique where a function calls itself to solve smaller instances of the same problem. It is particularly useful when a problem can be broken down into similar sub-problems.

In financial forecasting, recursion can be used to repeatedly apply a growth formula over a number of years, using the result of the previous year as the base for the next.

**4. Analysis**

**Time Complexity:**

* Each recursive call reduces the number of years by one.
* Therefore, time complexity is **O(n)** where n is the number of years.

**Space Complexity:**

* Since each recursive call is placed on the call stack, space complexity is also **O(n)**.

**Optimization Possibility:**

To reduce redundant calculations and excessive use of the call stack, this recursive solution can be optimized using either:

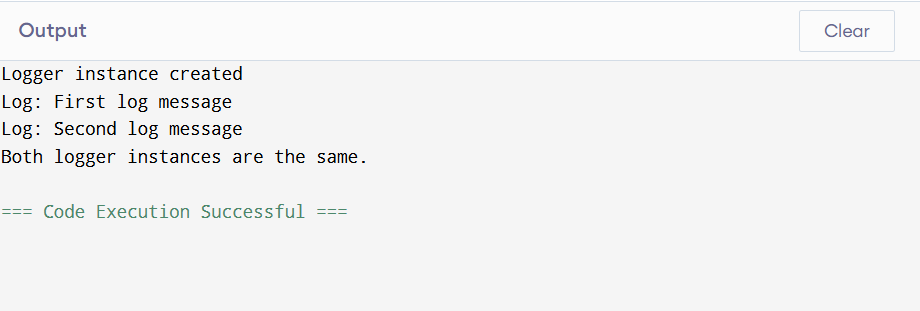
* **Memoization**: Store previously computed results in a map or array.
* **Iteration**: Rewrite the method using a loop to calculate the future value.

Conclusion:

* Recursive algorithms are useful for modeling repetitive, layered problems like financial forecasting. While easy to implement, recursive methods can be inefficient for large inputs without optimization. Iterative methods or memoized recursion are generally preferred for real-world applications due to better performance and space usage.

**DESIGN PRINCIPLES AND PATTERNS(OUTPUT) :**

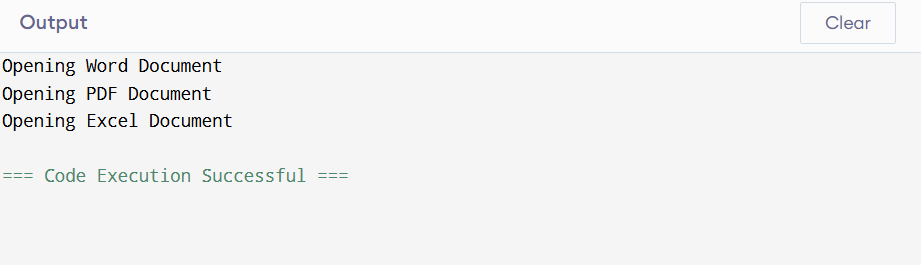
**Exercise 1 : Implementing Singleton method**

****

Explanation :-

* I made the constructor private so that no one can create a new Logger from outside the class.
* The getInstance() method checks if an object already exists. If not, it creates one.
* This makes sure only one object of Logger is ever created.
* I tested it by creating two variables and checking if they point to the same object and they do**.**

**Exercise 2 : Implementing Factory method :**

****

*Document (interface):*  
Acts like a common contract for all document types. It makes sure that every document class has an open() method.

*WordDocument, PdfDocument, ExcelDocument:*  
These are the actual document classes. Each one implements the Document interface and defines how its open() method works.

*DocumentFactory (abstract class):*This is an abstract class that declares the createDocument() method. It doesn't say how the document is created—just that there must be a way to create one.

*WordDocumentFactory, PdfDocumentFactory, ExcelDocumentFactory:*  
These are the specific factories. Each one extends DocumentFactory and knows how to create a specific type of document (Word, PDF, Excel).

*Main class:*This is where we test the whole setup. We use the factories to create documents and then call their open() methods to see them in action**.**