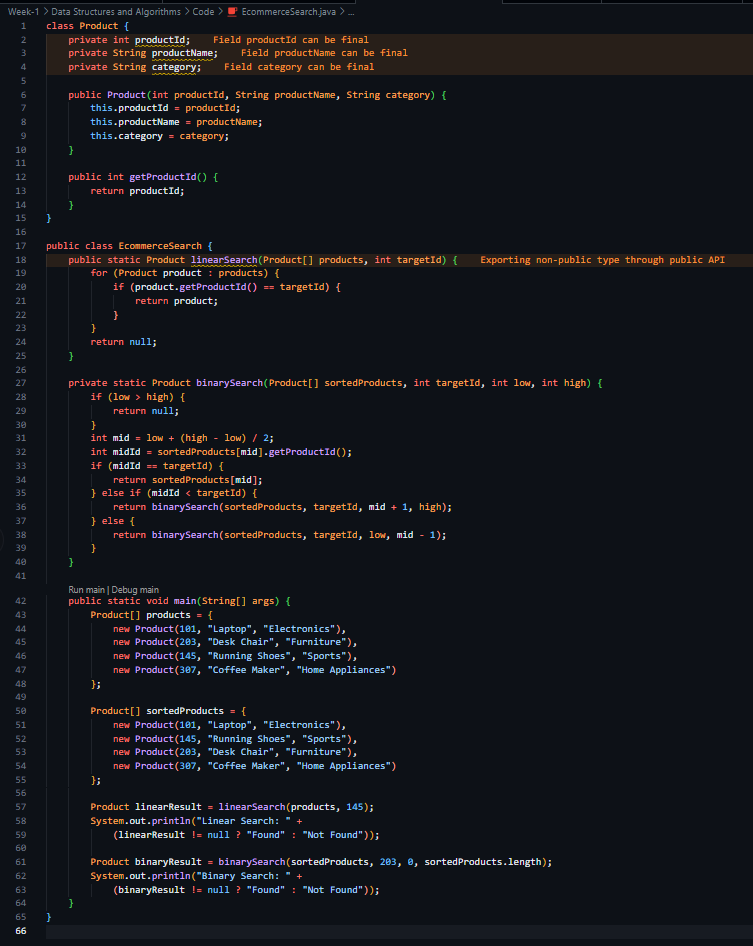
**Exercise 2: E-commerce Platform Search Function**

Code,

Output,

1. **Big O Notation,**Big O notation describes algorithm efficiency by measuring how runtime or memory usage grows as input size increases. It helps compare algorithms by focusing on worst-case performance.
2. **Search Scenarios:**
   * **Best-case:** Target found immediately (O(1))
   * **Average-case:** Target found in middle position (O(n) for linear, O(log n) for binary)
   * **Worst-case:** Target not present (O(n) for linear, O(log n) for binary)
3. **Algorithm Analysis:**

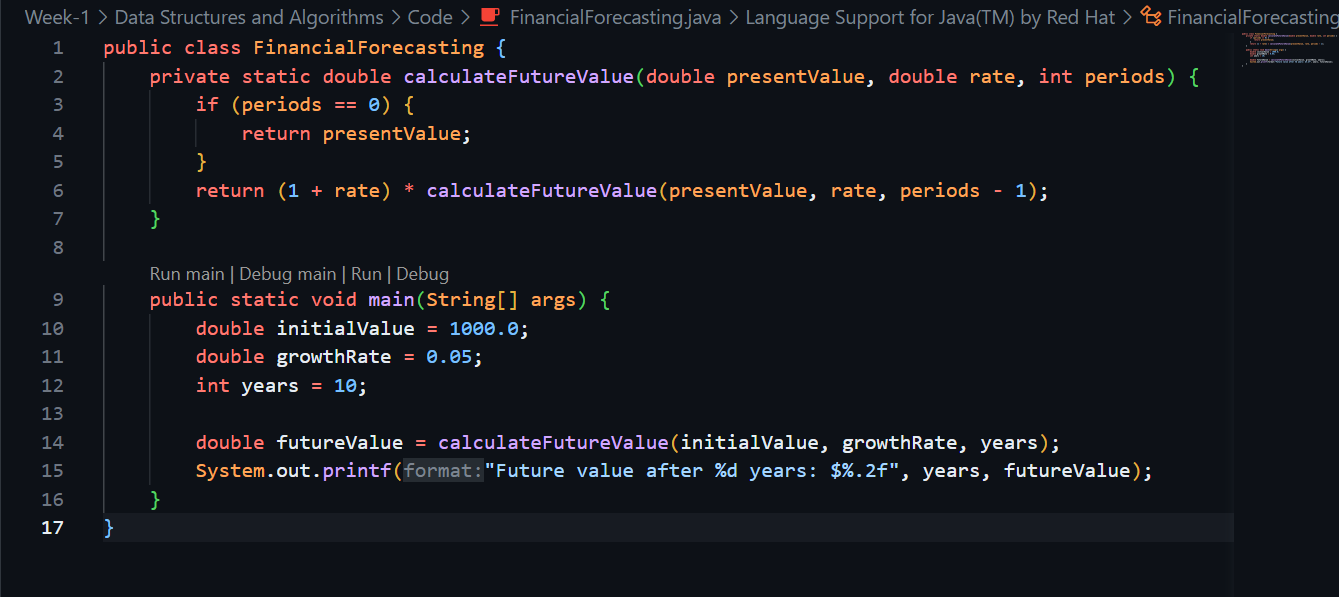
| **Algorithm** | **Time Complexity** | **Best For** |
| --- | --- | --- |
| **Linear Search** | O(n) (worst/average) | Small datasets, unsorted data |
| **Binary Search** | O(log n) (worst) | Large sorted datasets |

1. **Recommendation:**Binary search is superior for e-commerce platforms because:

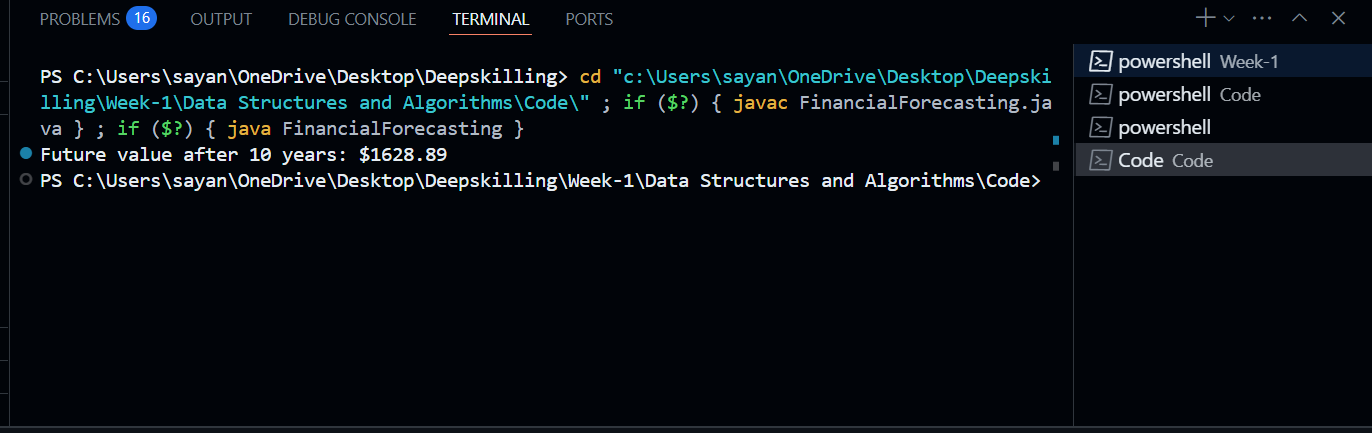
* **Efficiency**: Logarithmic time complexity (O(log n)) scales better with large product catalogues
* **Performance:** 1,000,000 products require only ~20 comparisons (vs 1,000,000 in linear search)
* **Practicality:** Products can be indexed/sorted during database updates

**Exercise 7: Financial Forecasting**

Code,



Output,



1. **Recursion Concept**:  
   Recursion solves problems by breaking them into smaller self-similar subproblems. It simplifies complex problems by reducing them to base cases and recursive steps.

**Analysis:**

1. **Time Complexity**:
   * Original: O(2^n) without optimization (exponential)
   * Optimized: O(n) using memoization (linear)
2. **Optimization Techniques**:
   * **Memoization**: Cache results of subproblems to prevent redundant calculations
   * **Iterative Approach**: Convert recursion to loops to avoid stack overflow