Understand the Problem:

Why data structures and algorithms matter:

* In a warehouse with potentially thousands of products, efficient search, update, and delete operations are necessary.
* Without good data structures, operations become slow and memory inefficient, especially as inventory grows.

Suitable Data Structures:

* HashMap is a strong choice:
  + Fast lookup (O(1) on average) using productId as the key.
  + Easy to update and delete.
* ArrayList is another option, but:
  + Requires linear search (O(n)) for lookup/update/delete by productId.
  + Best suited when order matters or productId is not unique.

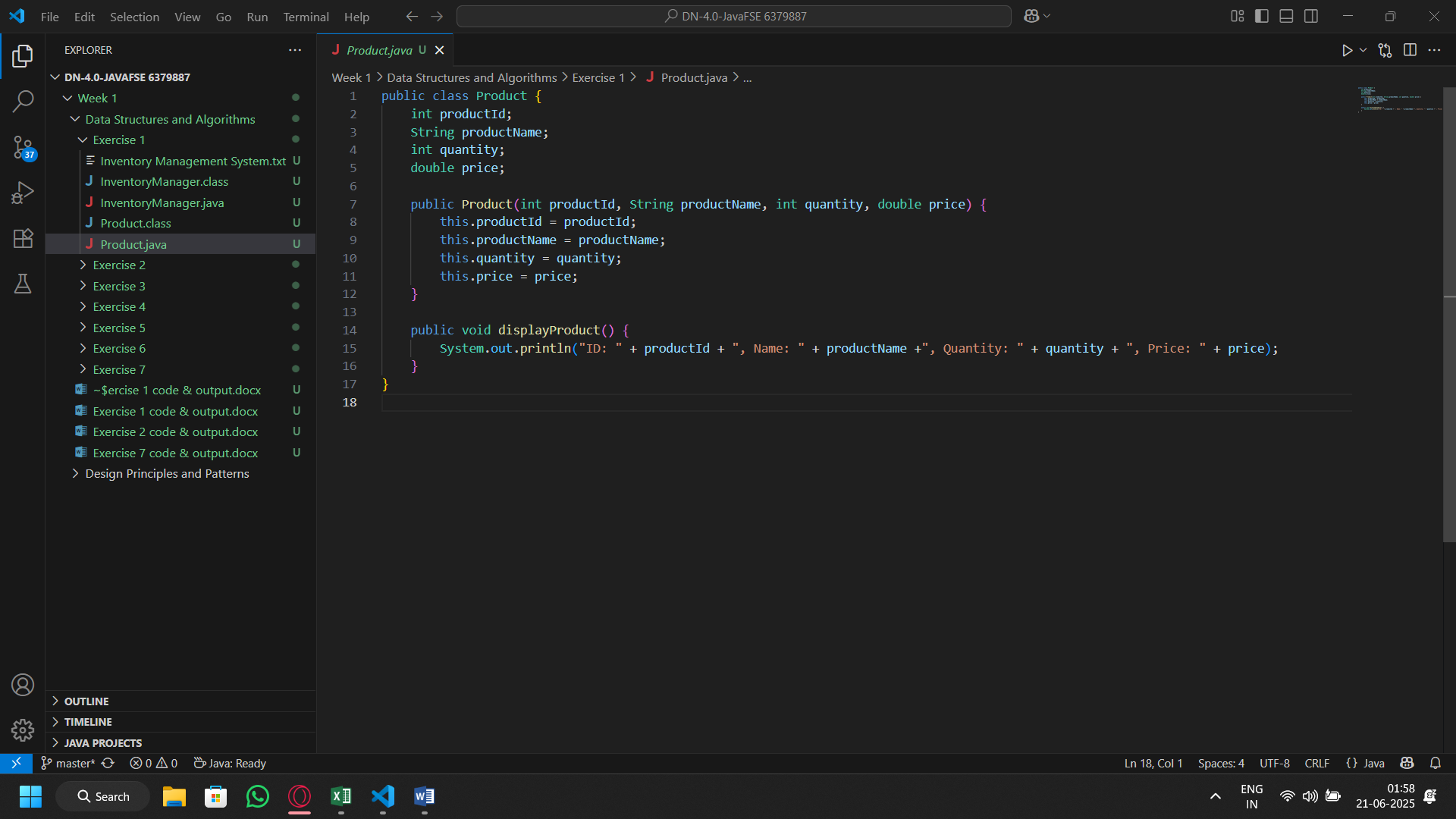
Analysis:

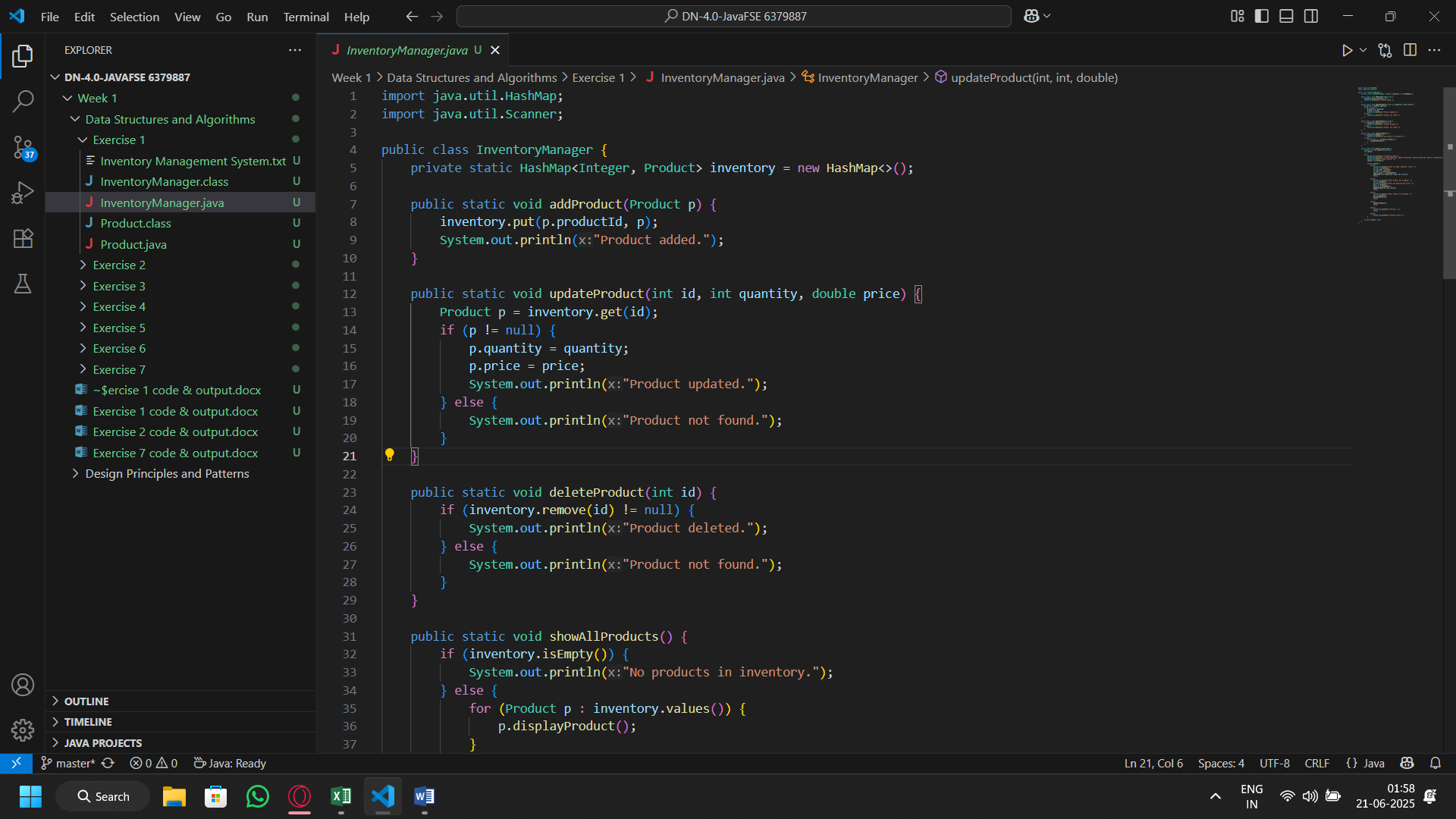
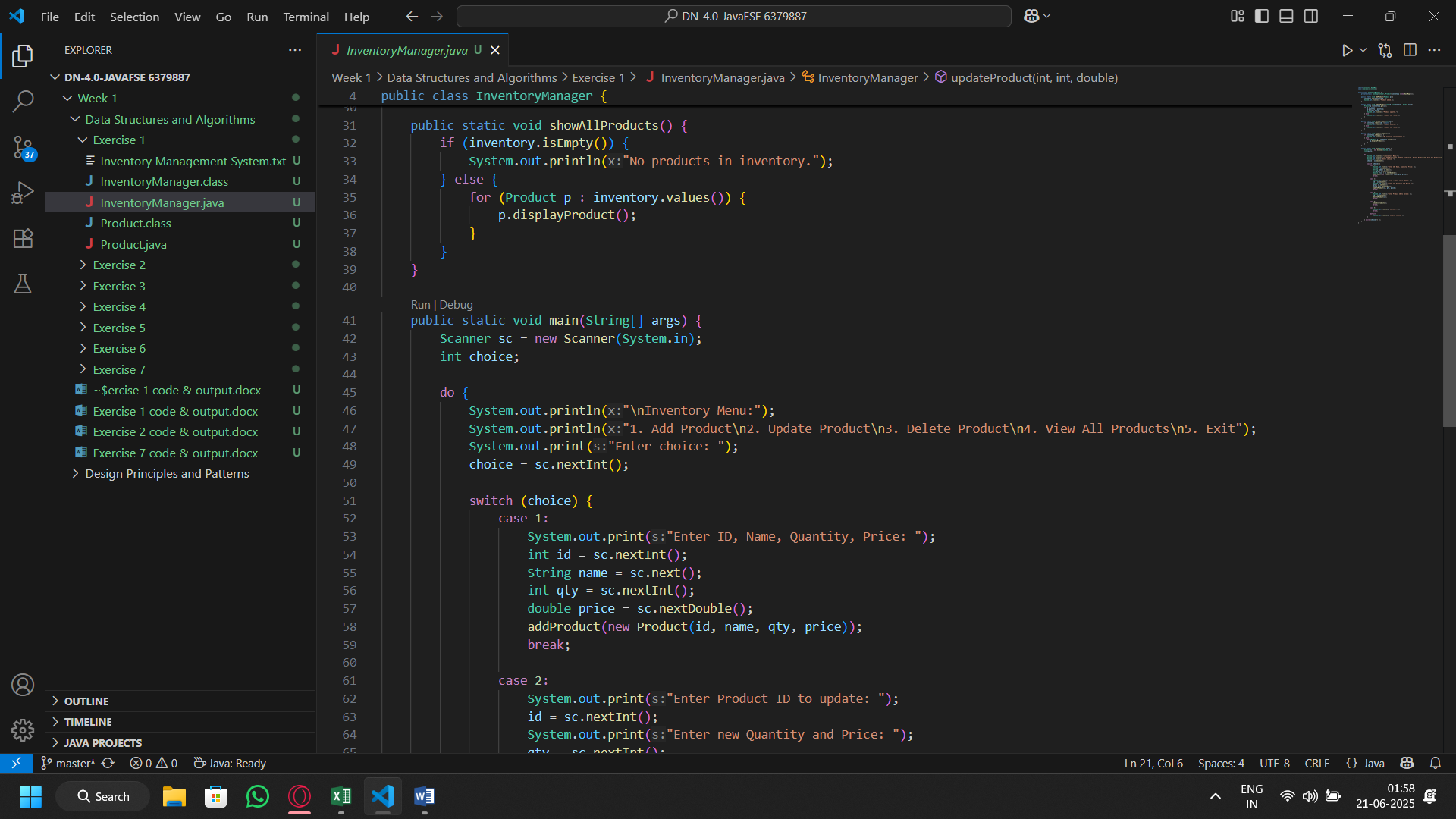
|  |  |  |
| --- | --- | --- |
| Operation | Time Complexity(HashMap) | Explanation |
| addProduct() | O(1) average | Uses put() |
| updateProduct() | O(1) average | Uses get() |
| deleteProduct() | O(1) average | Uses remove() |
| showAllProducts() | O(n) | Iterates through values |

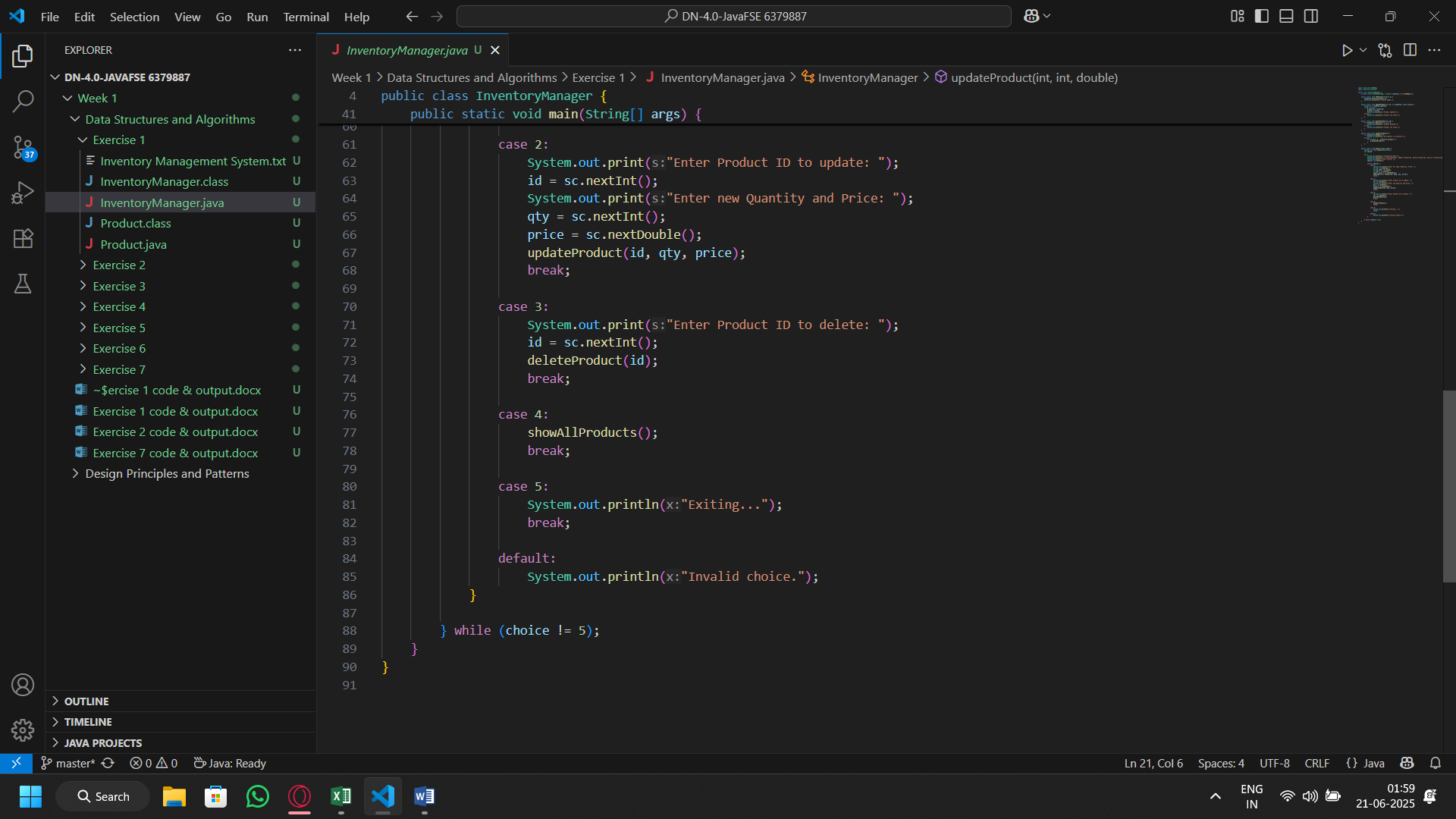
Optimization:

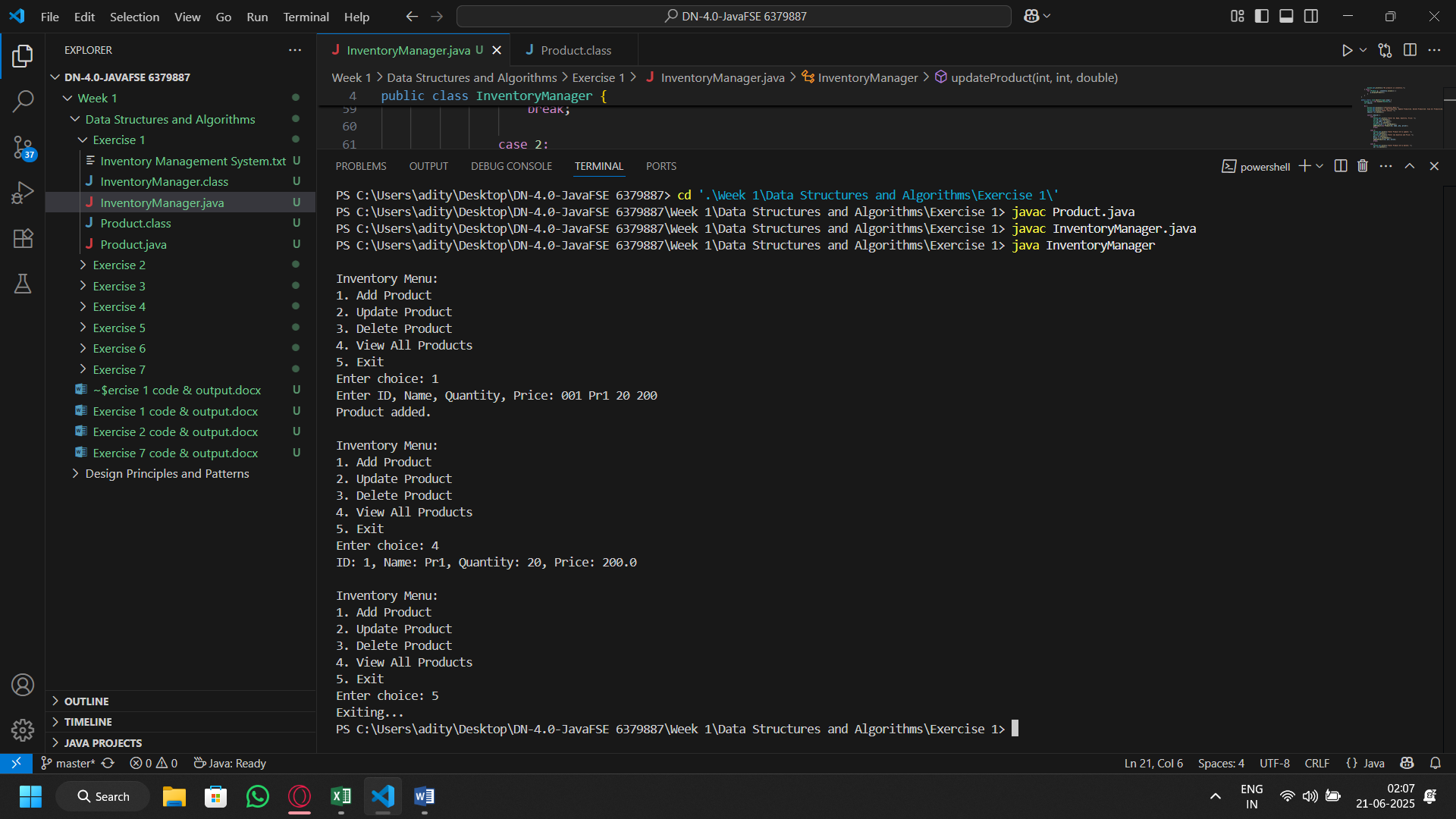
* If real-time inventory sync is needed, use ConcurrentHashMap in multithreaded environments.
* For persistence, integrate with a database (e.g., MySQL).
* Add search by name using a Map<String, List<Product>> if needed.

Code:

Product class

Inventory Manager Class



Output: