**Exercise 1: Inventory Management System**

**Explain why data structures and algorithms are essential in handling large inventories.**

Proper data structures allow quick access to inventory data, which is crucial for operations like searching for a product, updating quantities, and managing stock levels. As the inventory grows, the system must efficiently handle the increased data without significant performance degradation. Efficient data structures ensure optimal use of memory, avoiding wastage and improving performance.

**Discuss the types of data structures suitable for this problem**

1. Array list- It provides fast iteration, index-based access and useful for small-medium sized inventories.
2. HashMap- Allows fast lookups, inserts, and deletions by key, making it suitable for large inventories where quick access by product ID is necessary.
3. Useful for scenarios where frequent insertions and deletions occur at various positions in the list.
4. Tree Map- Maintains sorted order and allows fast range queries, useful for operations requiring sorted data access.

I used the **HashMap** data structure to implement the inventory management system.

**Analyze the time complexity of each operation (add, update, delete) in your chosen data structure.**

* **Add Product:** O(1) - Inserting into a HashMap is on average O(1).
* **Update Product:** O(1) - Updating a value in a HashMap is on average O(1).
* **Delete Product:** O(1) - Deleting a key-value pair from a HashMap is on average O(1)
* **Retrieve Product:** O(1) - Retrieving a value by key from a HashMap is on average O(1).

**Discuss how you can optimize these operations.**

Implemented batch updates to reduce the number of individual operations. Used Cache frequently accessed products to speed up retrieval times. Implemented secondary indexes on attributes like *productName* for faster search operations based on those attributes. Used concurrent data structures (like ConcurrentHashMap) if the system requires handling multiple threads for adding, updating, and deleting products simultaneously.