## **Exercise 1:Inventory Management System**

## **Why Are Data Structures and Algorithms Important in Handling Large Inventories?**

When managing a large inventory system, there are thousands of products to **add, search, update, and delete**.

If we don’t use the right **data structures** and **algorithms**, these operations can become **slow and inefficient** as the data grows.

### **Importance:**

* **Data Structures** help in **organizing and storing data properly**, so that it can be accessed and modified quickly.
* **Algorithms** decide **how operations like search, add, and delete are performed efficiently**.

Without them:

* Searching for a product would take too long.
* Updating or deleting records would be slow.
* System performance would drop with large data.

## **Types of Data Structures Suitable for Inventory Management**

To handle inventory efficiently, we need data structures that can **store, search, update, and delete product records quickly**. Here are some good options:

### **1. ArrayList**

* Stores products in a list.
* Easy to **add products** at the end.
* **Searching, updating, and deleting** takes time as you may need to check each product one by one.

**Good for:** Small inventories.

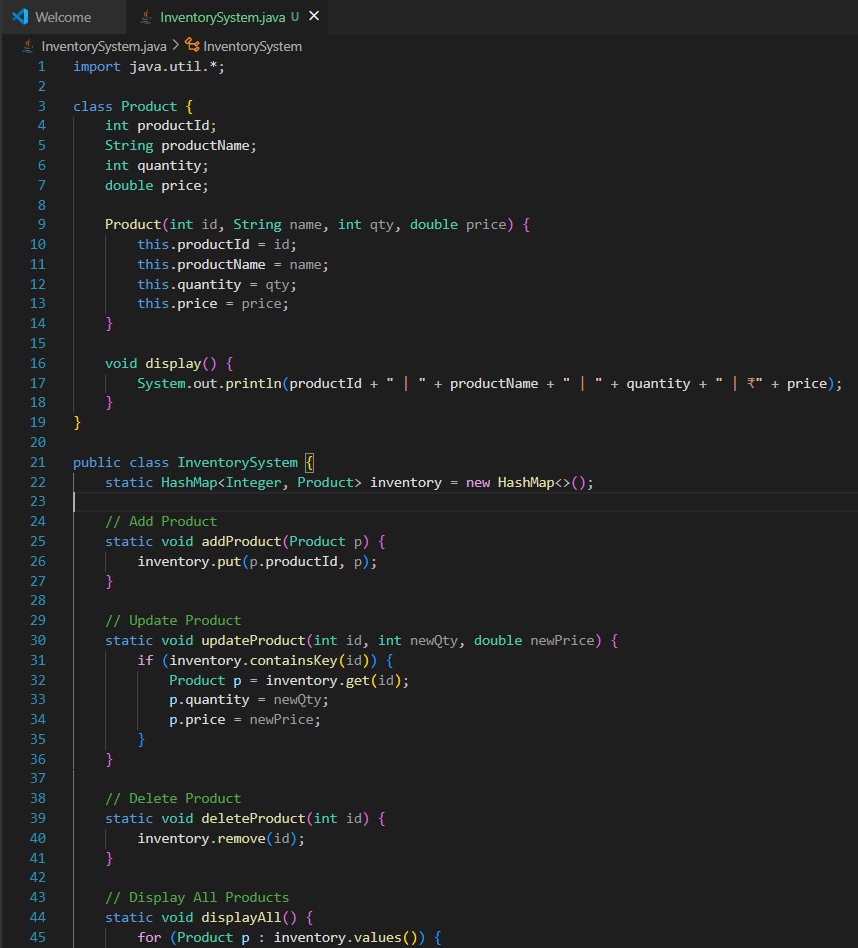
### **2. HashMap**

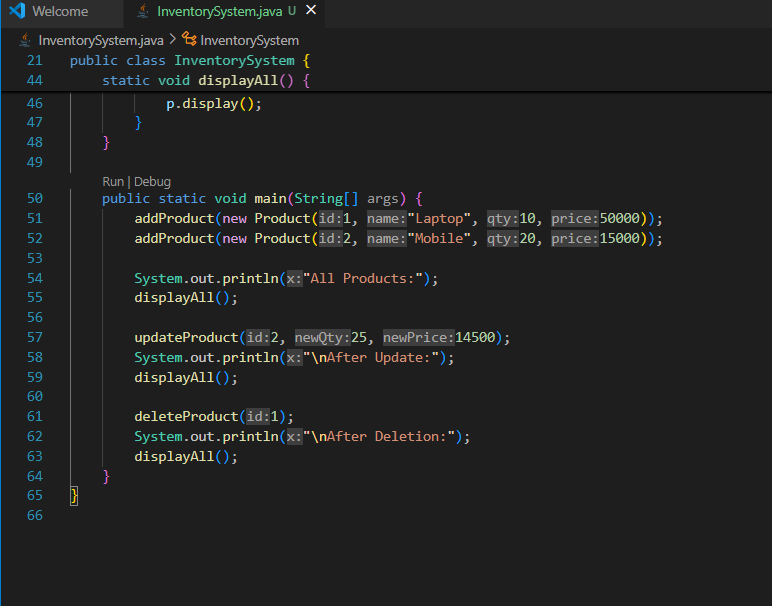
* Stores products with a **unique productId as a key**.
* Provides **fast search, add, update, and delete** operations.

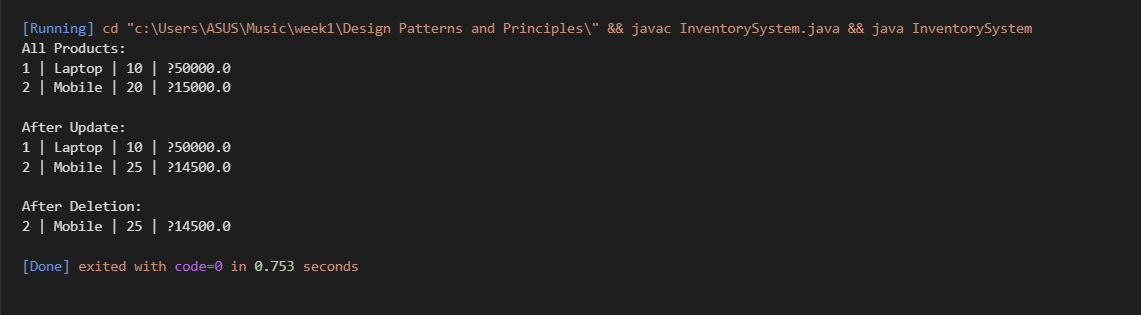
**Good for:** Large inventories where quick access is needed.

### **3. LinkedList (if needed)**

* Stores products in a connected list.
* Useful when **frequent adding/removing at various positions** is needed.
* But **searching is slower** compared to HashMap.







## **Time Complexity of Operations in Inventory Management**

|  |  |  |
| --- | --- | --- |
| **Operation** | **Time Complexity (Average Case)** | **Explanation** |
| **Add** | O(1) | Product is added using productId as key. |
| **Update** | O(1) | Directly access the product by its ID, then update fields. |
| **Delete** | O(1) | Remove the product quickly by its ID. |
| **Search** | O(1) | Retrieve product using productId as key. |

## **Why is it Fast?**

* **HashMap uses hashing technique** to directly access the product based on its unique key (productId).
* No need to search through the entire list like in ArrayList or LinkedList.

## **How to Optimize Inventory Operations**

To make **add, update, delete, and search** operations faster in an inventory system, you can:

### **1⃣. Use HashMap**

* Store products with **productId as the key**.
* Gives **O(1)** time for add, update, delete, and search.

### **2. Keep Unique Product IDs**

* Ensures no duplication.
* Makes access and updates direct and fast.

### **3. Avoid Searching by Names**

* Names can repeat and take time to search.
* Always use **productId for lookups**.

### **4.Maintain Clean, Structured Data**

* Regularly remove unused or outdated entries.
* Keeps data organized and reduces unnecessary processing.