**Exercise 1: Inventory Management System**

**1. Understand the Problem:**

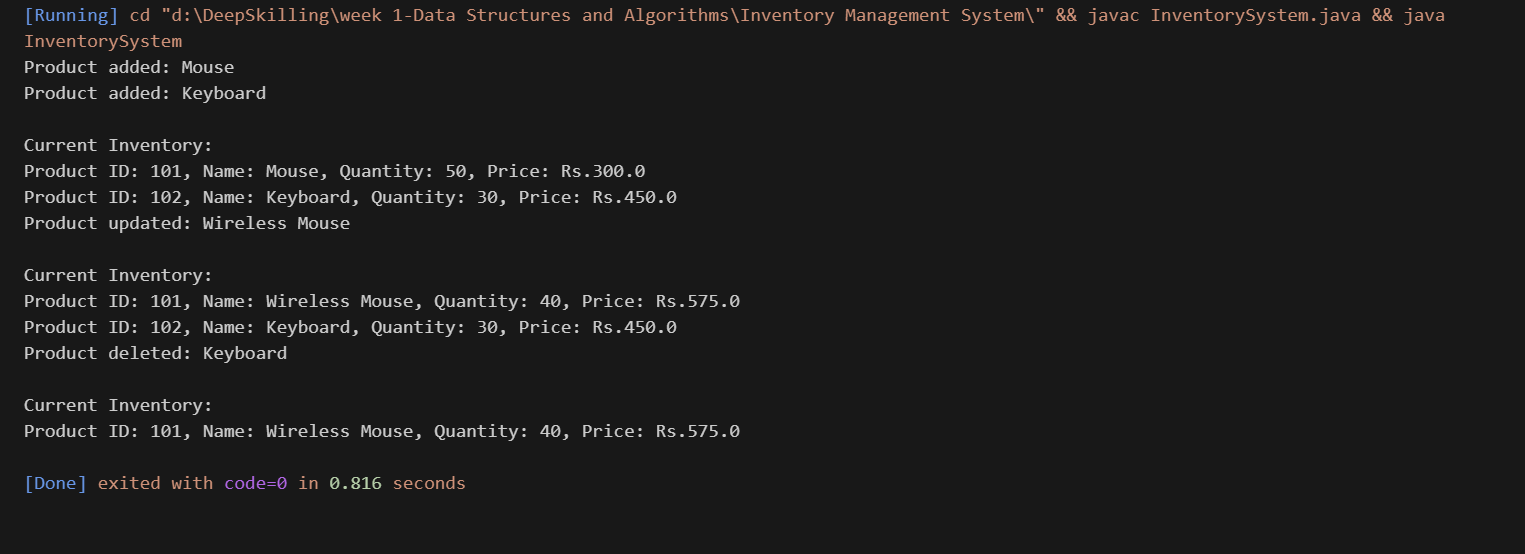
* **Efficiency**: Efficient data structures ensure quick access, search, update, and deletion of products.
* **Scalability**: Good algorithms ensure that operations scale well as the inventory grows.
* **Performance**: Operations like searching for a product by ID must be near-instant, which would be inefficient with linear search in an array.

**Suitable Data Structures**

* **ArrayList** : for small inventories or when order matters.
* **HashMap** : best for fast lookups using productid.
* **TreeMap** : when sorted view of inventory is needed.

**Best Choice:** HashMap, where productId is the key. Fast operations and easy to manage.

**Output:**



**4. Analysis:**

* **Time Complexity:**

#### addProduct(…): O(1)

#### updateProduct(...): O(1)

#### deleteProduct(…): O(1)

#### displayInventory( ): O(n)

* **Optimization Suggestions:**

#### **Avoid Redundant Checks** – Use get() and check for null.

#### **Use TreeMap** – For sorted inventory output.

#### **Add Secondary Indexes** – For fast name/price searches.

#### **Database Integration** – For large-scale data handling.

#### **Caching** – Speed up frequent product lookups.

#### **Pagination** – Limit inventory display output.

#### **Use ConcurrentHashMap** – For thread-safe operations.

|  |
| --- |
|  |

|  |
| --- |
|  |