Exercise 1: Inventory Management System

# 1. Understanding the Problem

Why are Data Structures and Algorithms essential?  
  
Efficient data structures and algorithms help manage large inventories with:  
- Faster retrieval (e.g., product search by ID or name).  
- Optimized memory usage.  
- Quick updates and easy scalability.  
  
Suitable Data Structures:  
- HashMap: Ideal for quick lookup, insertion, and deletion by productId.  
- ArrayList: Useful if ordered traversal or sorting is required.  
  
We'll use HashMap<Integer, Product> for this task.

# 2. Project Setup

Structure your folder like this:  
  
InventoryManagementSystem/  
├── src/  
│ └── InventoryManager.java  
│ └── Product.java  
├── README.md

# 3. Implementation

Visit the code in repository for the full implementation of:  
- Product.java: Defines product attributes and methods.  
- InventoryManager.java: Manages inventory using HashMap and includes methods to add, update, delete, and display products.

# 4. Time Complexity Analysis

Operation | Data Structure Used | Time Complexity | Explanation  
------------------|---------------------|------------------|-------------  
Add Product | HashMap | O(1) | Constant time for insertion by key.  
Update Product | HashMap | O(1) | Direct access using productId.  
Delete Product | HashMap | O(1) | Efficient removal using key.

Optimizations:  
- For bulk uploads: use batch processing.  
- For product search by name: maintain a second HashMap<String, List<Product>>.  
- For sorting: temporarily convert inventory.values() to a List and use Collections.sort().