**Exercise 1: Inventory Management System**

**Scenario:**

You are developing an inventory management system for a warehouse. Efficient data storage and retrieval are crucial.

**Steps:**

1. **Understand the Problem:**
   * Explain why data structures and algorithms are essential in handling large inventories.
   * Discuss the types of data structures suitable for this problem.
2. **Setup:**
   * Create a new project for the inventory management system.
3. **Implementation:**
   * Define a class Product with attributes like **productId**, **productName**, **quantity**, and **price**.
   * Choose an appropriate data structure to store the products (e.g., ArrayList, HashMap).
   * Implement methods to add, update, and delete products from the inventory.
4. **Analysis:**
   * Analyze the time complexity of each operation (add, update, delete) in your chosen data structure.
   * Discuss how you can optimize these operations.

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

Managing a large inventory—such as thousands of products in a warehouse—requires more than just storing data. It demands **efficient ways to store, search, update, and delete** that data quickly and reliably.

Here’s why **data structures** and **algorithms** are critical in this context:

1. Efficient Data Access & Retrieval
2. Fast Insertion and Deletion
3. Optimized Memory Usage
4. Scalability
5. Real-Time Decision Making
6. Data Consistency and Integrity
7. Time and Cost Savings

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

In an inventory management system, the data structure you choose directly affects how **efficiently** you can store, retrieve, and update product information. Each data structure has its own strengths depending on the specific use case (e.g., fast lookups, sorted lists, or frequent insertions/deletions).

1. HashMap (or Dictionary / Hash Table)
2. ArrayList (or Dynamic Array)
3. TreeMap (Red-Black Tree / BST)
4. LinkedList etc…

**Code:**

import java.util.HashMap;

import java.util.Scanner;

class Product {

    private String productId;

    private String productName;

    private int quantity;

    private double price;

    public Product(String productId, String productName, int quantity, double price) {

        this.productId = productId;

        this.productName = productName;

        this.quantity = quantity;

        this.price = price;

    }

    public String getProductId() {

        return productId;

    }

    public String getProductName() {

        return productName;

    }

    public int getQuantity() {

        return quantity;

    }

    public double getPrice() {

        return price;

    }

    public void setProductName(String productName) {

        this.productName = productName;

    }

    public void setQuantity(int quantity) {

        this.quantity = quantity;

    }

    public void setPrice(double price) {

        this.price = price;

    }

    @Override

    public String toString() {

        return "Product ID: " + productId + ", Name: " + productName +

               ", Quantity: " + quantity + ", Price: $" + price;

    }

}

class InventoryManager {

    private HashMap<String, Product> inventory = new HashMap<>();

    public void addProduct(Product product) {

        if (inventory.containsKey(product.getProductId())) {

            System.out.println("Product ID already exists!");

        } else {

            inventory.put(product.getProductId(), product);

            System.out.println("Product added successfully.");

        }

    }

    public void updateProduct(String productId, String name, int qty, double price) {

        Product p = inventory.get(productId);

        if (p != null) {

            p.setProductName(name);

            p.setQuantity(qty);

            p.setPrice(price);

            System.out.println("Product updated successfully.");

        } else {

            System.out.println("Product not found.");

        }

    }

    public void deleteProduct(String productId) {

        if (inventory.remove(productId) != null) {

            System.out.println("Product deleted successfully.");

        } else {

            System.out.println("Product not found.");

        }

    }

    public void displayInventory() {

        if (inventory.isEmpty()) {

            System.out.println("Inventory is empty.");

        } else {

            System.out.println("\n--- Inventory List ---");

            for (Product p : inventory.values()) {

                System.out.println(p);

            }

        }

    }

}

public class InventoryApp {

    public static void main(String[] args) {

        InventoryManager manager = new InventoryManager();

        Scanner scanner = new Scanner(System.in);

        while (true) {

            System.out.println("\n--- Inventory Management System ---");

            System.out.println("1. Add Product");

            System.out.println("2. Update Product");

            System.out.println("3. Delete Product");

            System.out.println("4. Display Inventory");

            System.out.println("5. Exit");

            System.out.print("Enter your choice: ");

            int choice;

            try {

                choice = Integer.parseInt(scanner.nextLine());

            } catch (NumberFormatException e) {

                System.out.println("Invalid input. Please enter a number from 1 to 5.");

                continue;

            }

            switch (choice) {

                case 1:

                    System.out.print("Enter Product ID: ");

                    String id = scanner.nextLine();

                    System.out.print("Enter Product Name: ");

                    String name = scanner.nextLine();

                    System.out.print("Enter Quantity: ");

                    int qty = Integer.parseInt(scanner.nextLine());

                    System.out.print("Enter Price: ");

                    double price = Double.parseDouble(scanner.nextLine());

                    Product newProduct = new Product(id, name, qty, price);

                    manager.addProduct(newProduct);

                    break;

                case 2:

                    System.out.print("Enter Product ID to update: ");

                    id = scanner.nextLine();

                    System.out.print("Enter New Product Name: ");

                    name = scanner.nextLine();

                    System.out.print("Enter New Quantity: ");

                    qty = Integer.parseInt(scanner.nextLine());

                    System.out.print("Enter New Price: ");

                    price = Double.parseDouble(scanner.nextLine());

                    manager.updateProduct(id, name, qty, price);

                    break;

                case 3:

                    System.out.print("Enter Product ID to delete: ");

                    id = scanner.nextLine();

                    manager.deleteProduct(id);

                    break;

                case 4:

                    manager.displayInventory();

                    break;

                case 5:

                    System.out.println("Exiting system. Goodbye!");

                    scanner.close();

                    return;

                default:

                    System.out.println("Invalid choice! Please select between 1 and 5.");

            }

        }

    }

}

**Output:**

**A screen shot of a computer

AI-generated content may be incorrect.**

**A black screen with white text

AI-generated content may be incorrect.**