**DIGITAL NURTURE 4.0 DEEP SKILLING JAVA FSE-WEEK1**

**NAME: SIVITHA GUNASEKARAN**

**SUPERSET ID: 6413354**

**WEEK 1: ALGORITHMS DATA STRUCTURES**

**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.

**CODE SAMPLES:**

import java.util.HashMap;

import java.util.Map;

class Product {

int productId;

String productName;

int quantity;

double price;

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

this.productId = productId;

this.productName = productName;

this.quantity = quantity;

this.price = price;

}

public String toString() {

return productId + " | " + productName + " | Qty: " + quantity + " | ₹" + price;

}

}

class Inventory {

Map<Integer, Product> products = new HashMap<>();

public void addProduct(Product p) {

products.put(p.productId, p);

System.out.println("Added: " + p);

}

public void updateProduct(int productId, int quantity, double price) {

if (products.containsKey(productId)) {

Product p = products.get(productId);

p.quantity = quantity;

p.price = price;

System.out.println("Updated: " + p);

} else {

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

}

}

public void deleteProduct(int productId) {

if (products.containsKey(productId)) {

Product removed = products.remove(productId);

System.out.println("Deleted: " + removed);

} else {

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

}

}

public void displayInventory() {

System.out.println("\nCurrent Inventory:");

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

System.out.println(p);

}

}

}

public class InventoryManagementSystem {

public static void main(String[] args) {

Inventory inventory = new Inventory();

inventory.addProduct(new Product(101, "Laptop", 10, 75000));

inventory.addProduct(new Product(102, "Mouse", 50, 500));

inventory.addProduct(new Product(103, "Keyboard", 30, 1500));

inventory.displayInventory();

inventory.updateProduct(102, 60, 550);

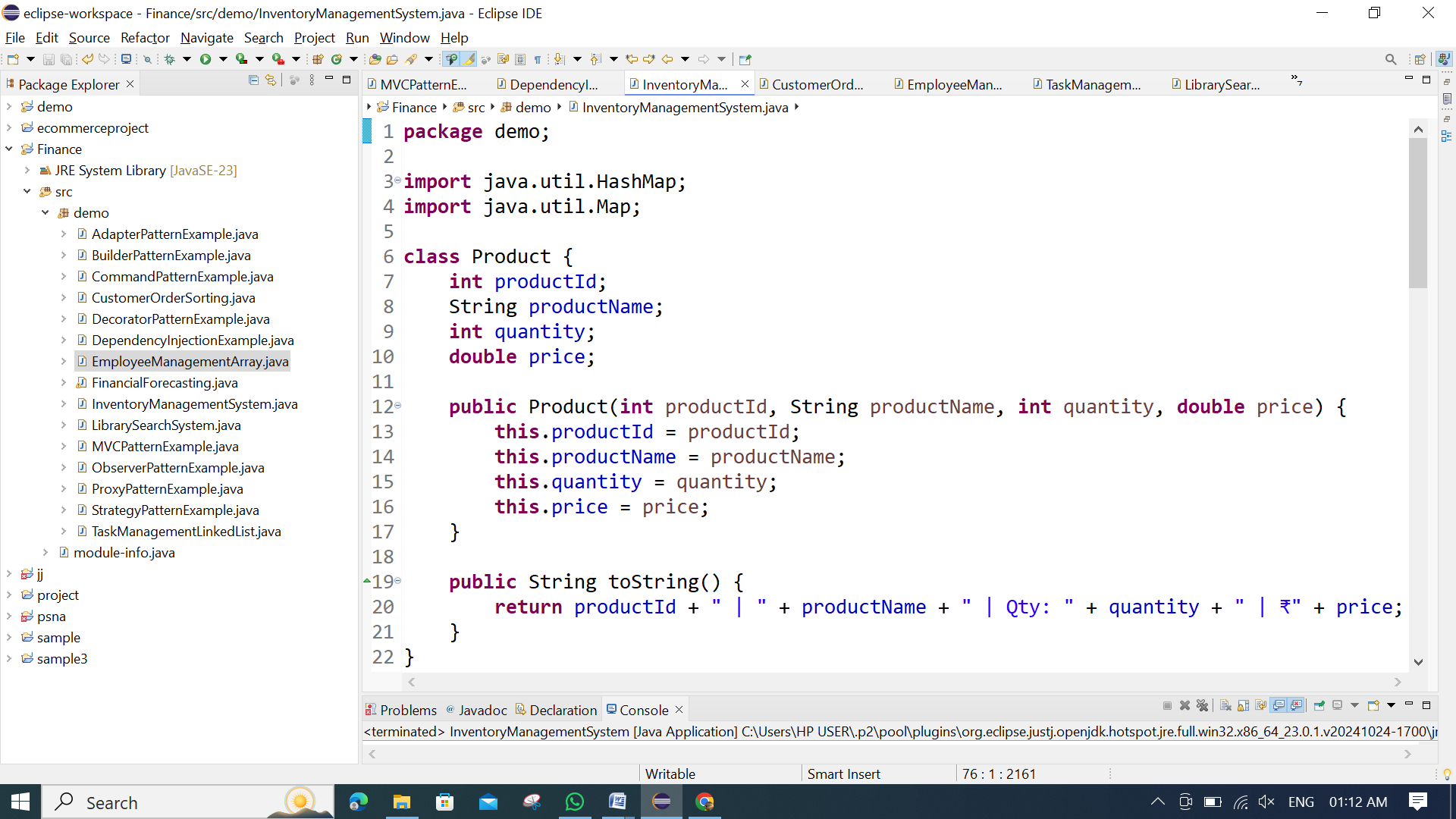
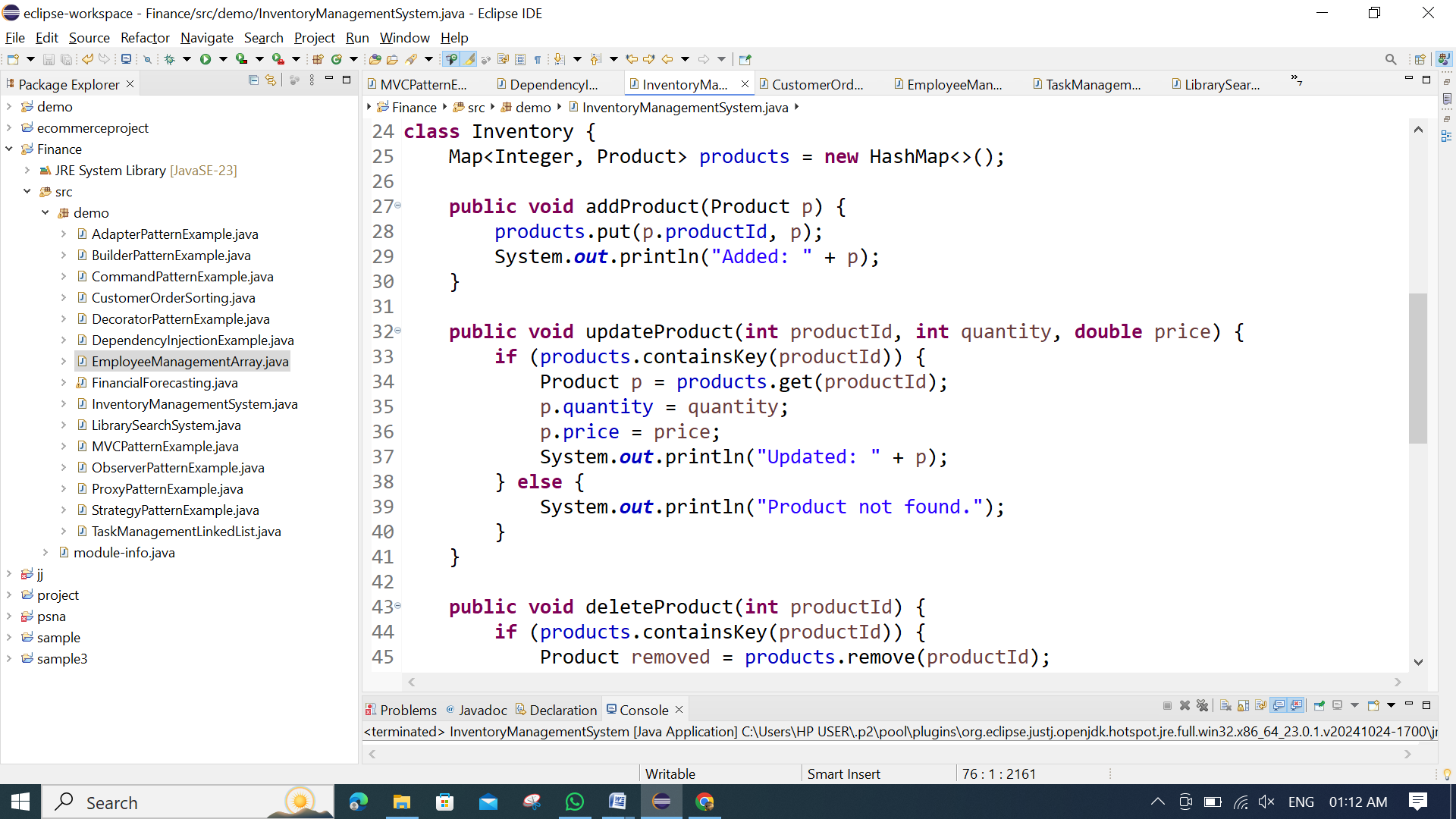
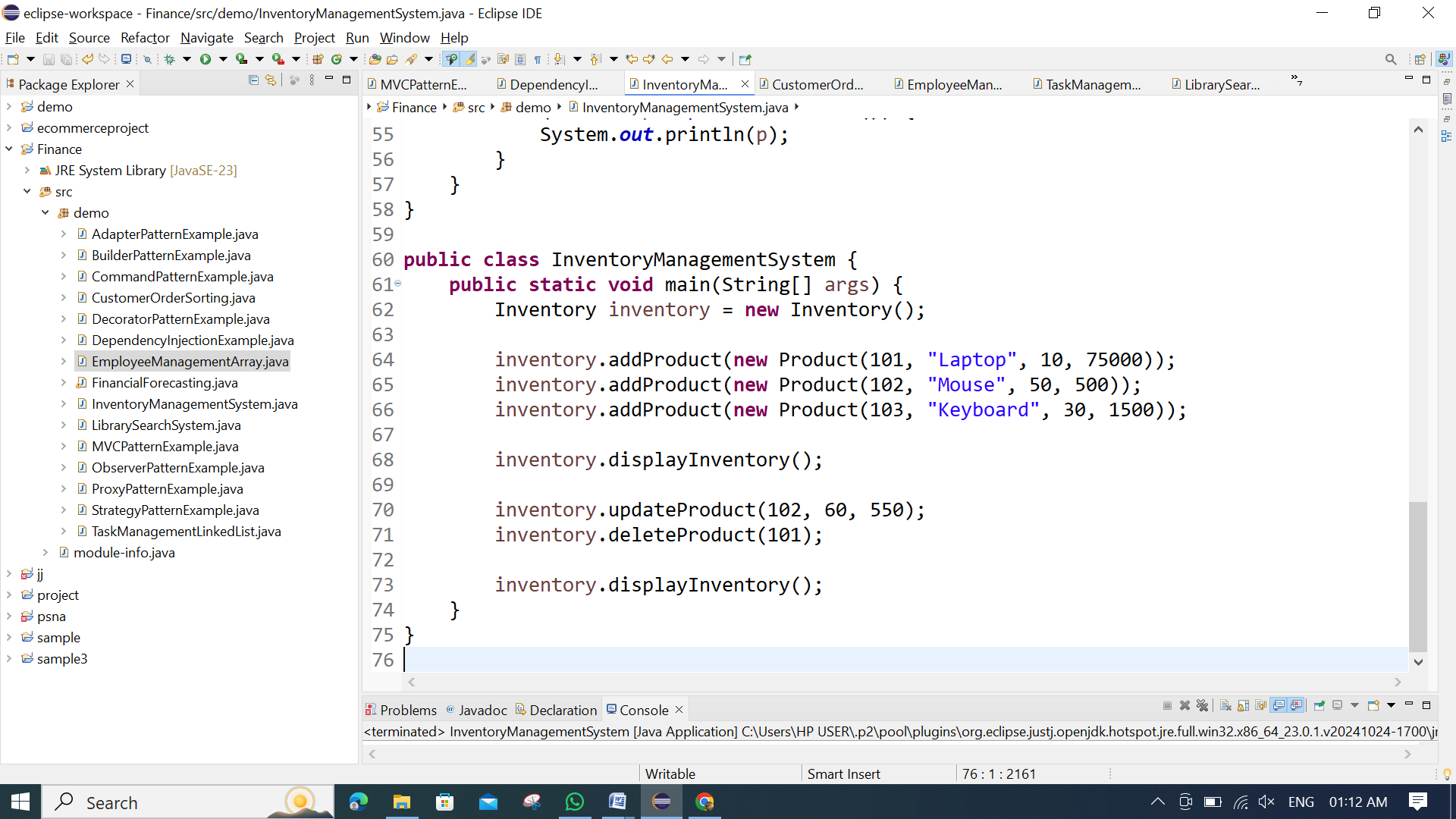
inventory.deleteProduct(101);

inventory.displayInventory();

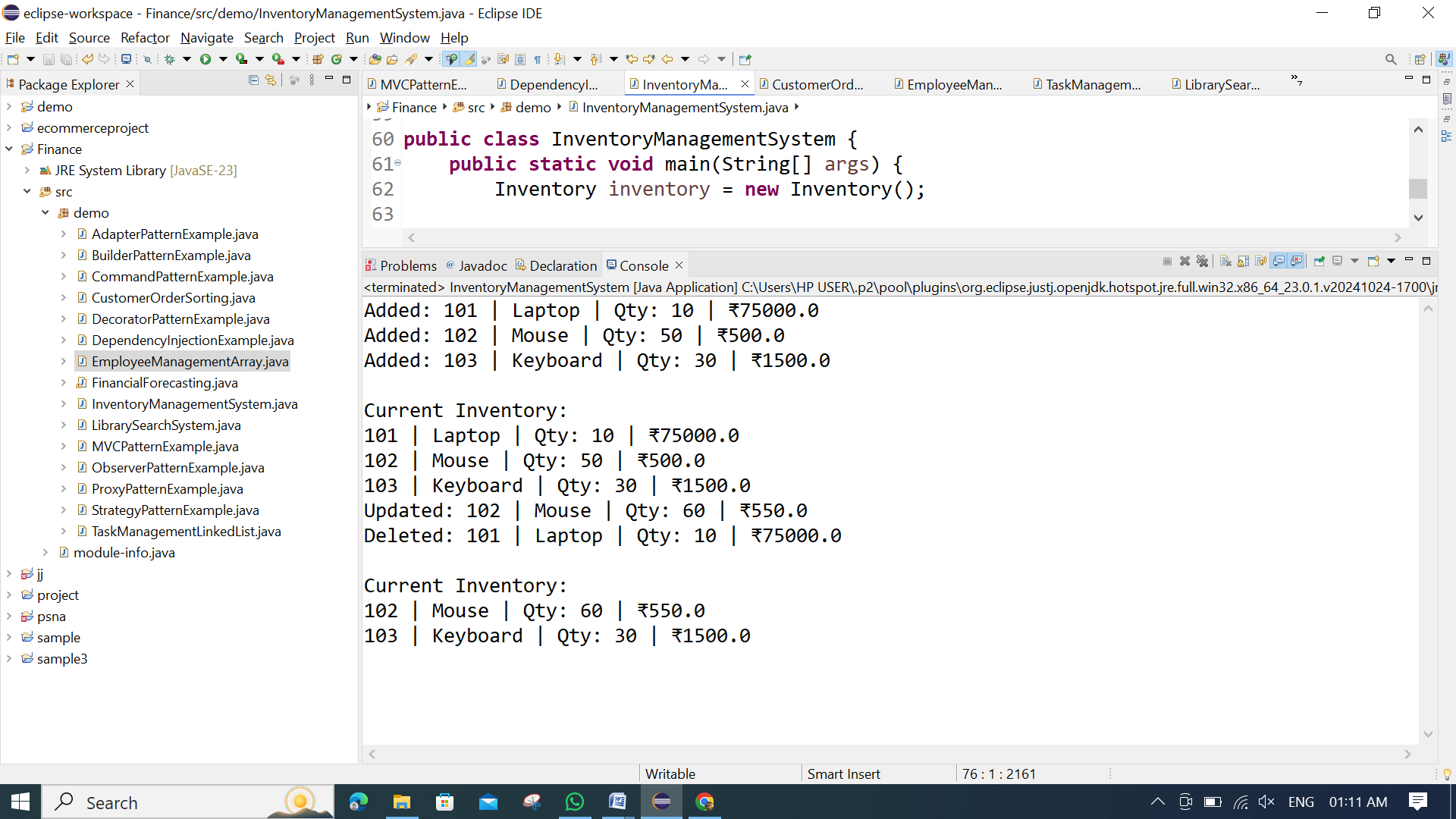
}

}

**MY SCREENSHOT PROOF:**

**  **

**OUTPUT:**

****