Week1 (Data Structure-Algorithms)

Exercise 1 - E - commerce platform search function

import java.util.\*;

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

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

    }

}

// Inventory manager class

public class InventoryManager {

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

    // Add new product

    public void addProduct(Product product) {

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

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

        } else {

            inventory.put(product.productId, product);

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

        }

    }

    // Update product by ID

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

        Product product = inventory.get(productId);

        if (product != null) {

            product.quantity = quantity;

            product.price = price;

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

        } else {

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

        }

    }

    public void deleteProduct(int 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 {

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

                System.out.println(p);

            }

        }

    }

    public static void main(String[] args) {

        InventoryManager manager = new InventoryManager();

        manager.addProduct(new Product(101, "Laptop", 10, 85000));

        manager.addProduct(new Product(102, "Smartphone", 50, 30000));

        manager.addProduct(new Product(103, "Tablet", 25, 15000));

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

        manager.displayInventory();

        manager.updateProduct(102, 60, 29000);

        System.out.println("\nInventory after update:");

        manager.displayInventory();

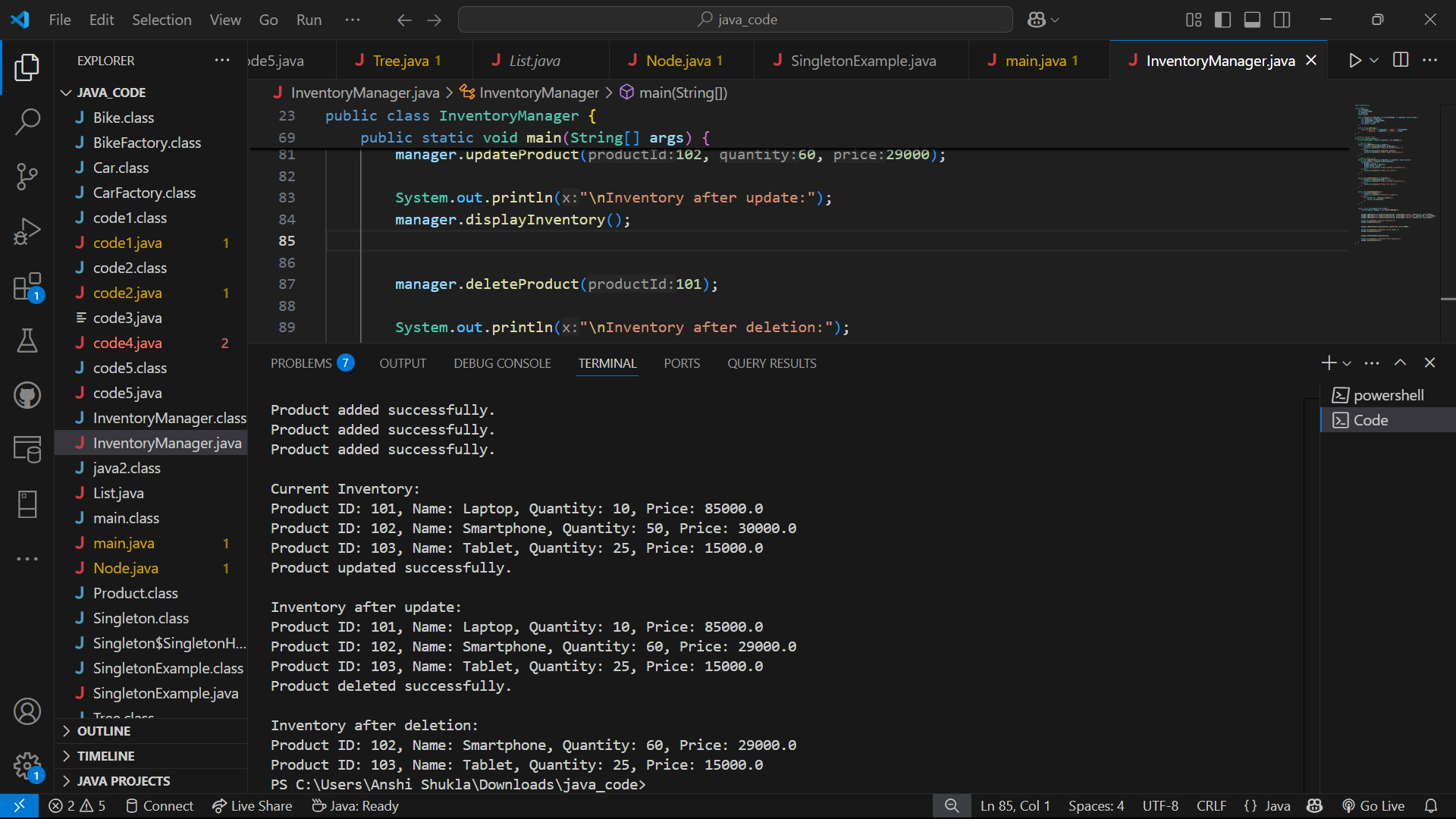
        manager.deleteProduct(101);

        System.out.println("\nInventory after deletion:");

        manager.displayInventory();

    }

}



Exercise - 2 - Financial Forecasting

public class FinancialForecast {

    public static double forecastFutureValue(double presentValue, double growthRate, int years) {

        if (years == 0) {

            return presentValue;

        }

        return (1 + growthRate) \* forecastFutureValue(presentValue, growthRate, years - 1);

    }

    public static void main(String[] args) {

        double presentValue = 10000;

        double growthRate = 0.05;

        int years = 10;

        double futureValue = forecastFutureValue(presentValue, growthRate, years);

        System.out.printf("Future value after %d years: %.2f\n", years, futureValue);

    }

}

