**Exercise 1: Implementing the Singleton Pattern**

import java.util.HashMap;

import java.util.Scanner;

public class Main {

    static 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 + " | " + quantity + " | $" + price;

        }

    }

    static class InventoryManager {

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

        public void addProduct(Product product) {

            inventory.put(product.productId, product);

        }

        public void updateProduct(int productId, Product updatedProduct) {

            if (inventory.containsKey(productId)) {

                inventory.put(productId, updatedProduct);

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

            } else {

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

            }

        }

        public void deleteProduct(int productId) {

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

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

            } else {

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

            }

        }

        public void printInventory() {

            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();

        Scanner scanner = new Scanner(System.in);

        while (true) {

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

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

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

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

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

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

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

            int choice = scanner.nextInt();

            if (choice == 5) break;

            switch (choice) {

                case 1:

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

                    int id = scanner.nextInt();

                    scanner.nextLine();

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

                    String name = scanner.nextLine();

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

                    int qty = scanner.nextInt();

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

                    double price = scanner.nextDouble();

                    manager.addProduct(new Product(id, name, qty, price));

                    break;

                case 2:

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

                    int uid = scanner.nextInt();

                    scanner.nextLine();

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

                    String uname = scanner.nextLine();

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

                    int uqty = scanner.nextInt();

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

                    double uprice = scanner.nextDouble();

                    manager.updateProduct(uid, new Product(uid, uname, uqty, uprice));

                    break;

                case 3:

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

                    int did = scanner.nextInt();

                    manager.deleteProduct(did);

                    break;

                case 4:

                    System.out.println("Current Inventory:");

                    manager.printInventory();

                    break;

                default:

                    System.out.println("Invalid choice. Try again.");

            }

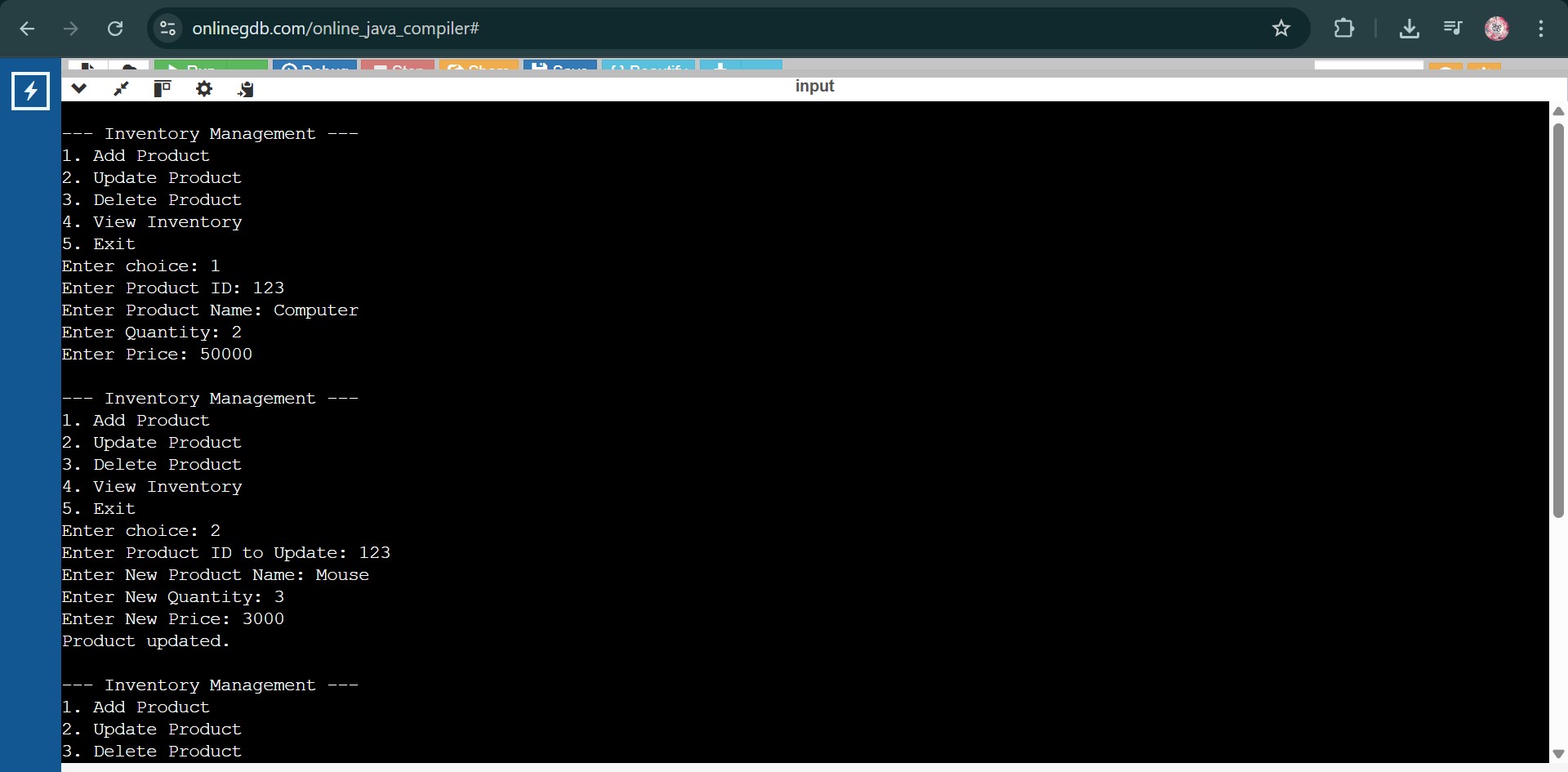
        }

        scanner.close();

        System.out.println("Program terminated.");

    }

}

OUTPUT: