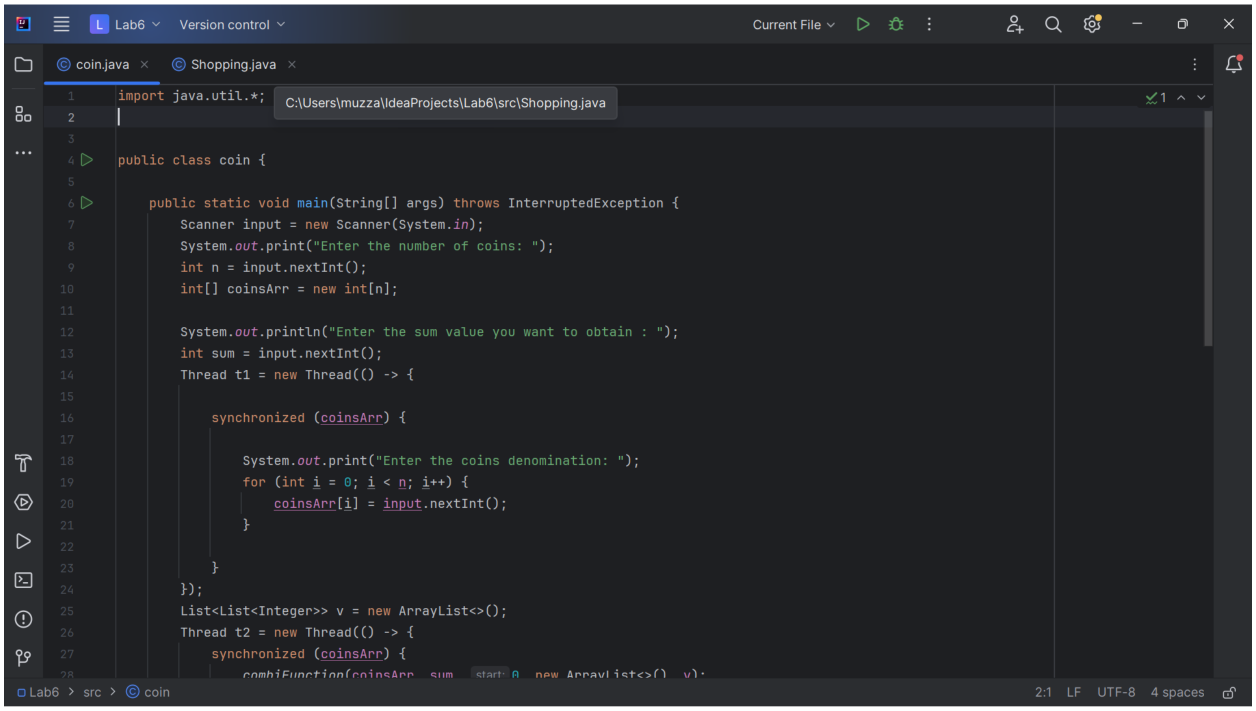
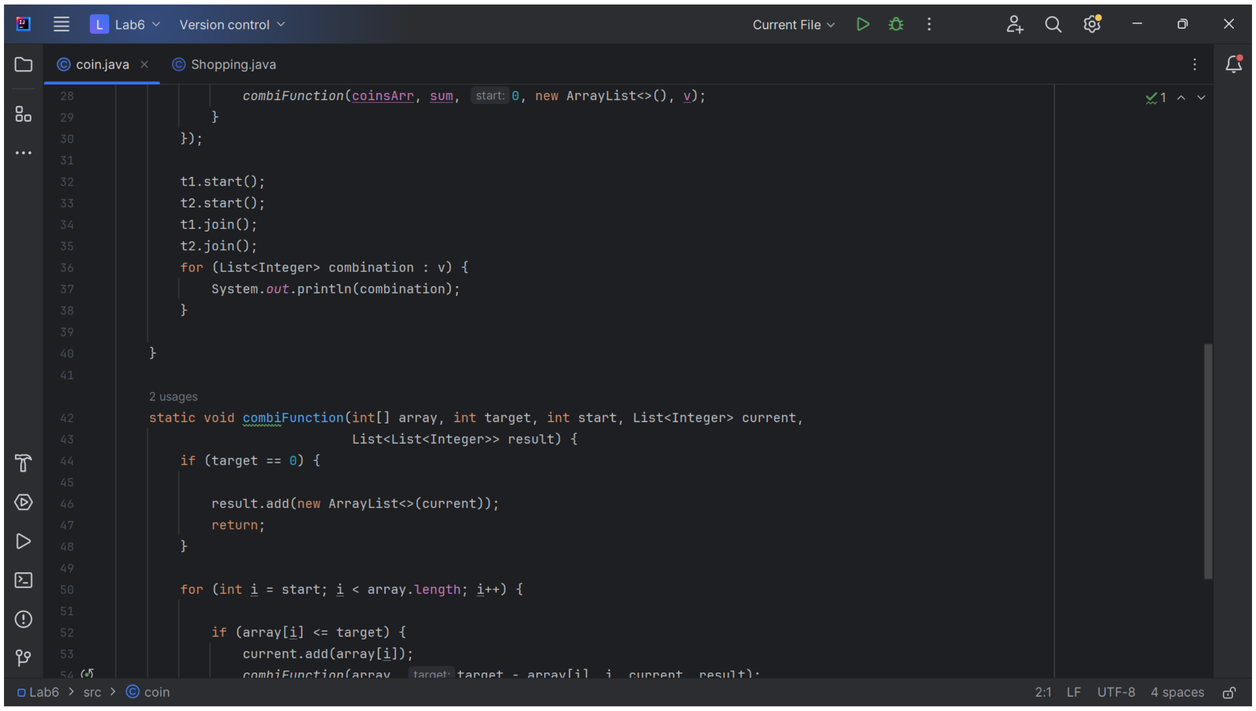
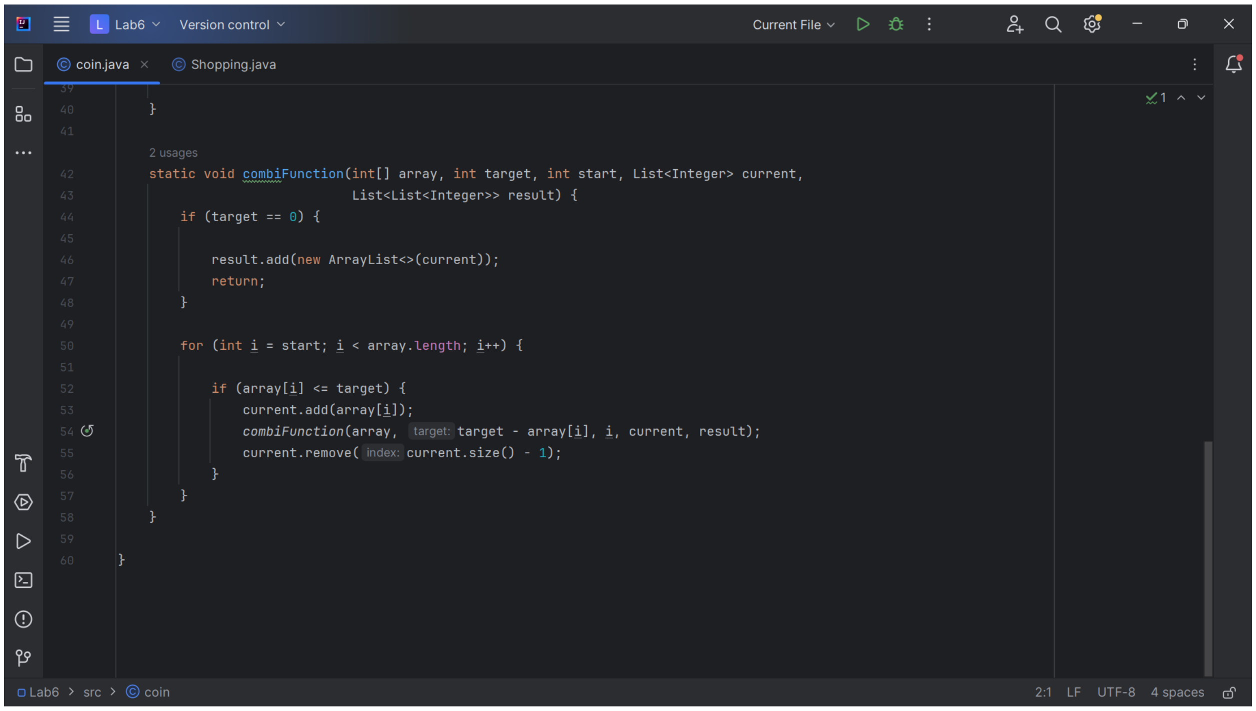
Program 1:


C:\Users\muzza\IdeaProjects\Lab6\src\Shopping.java

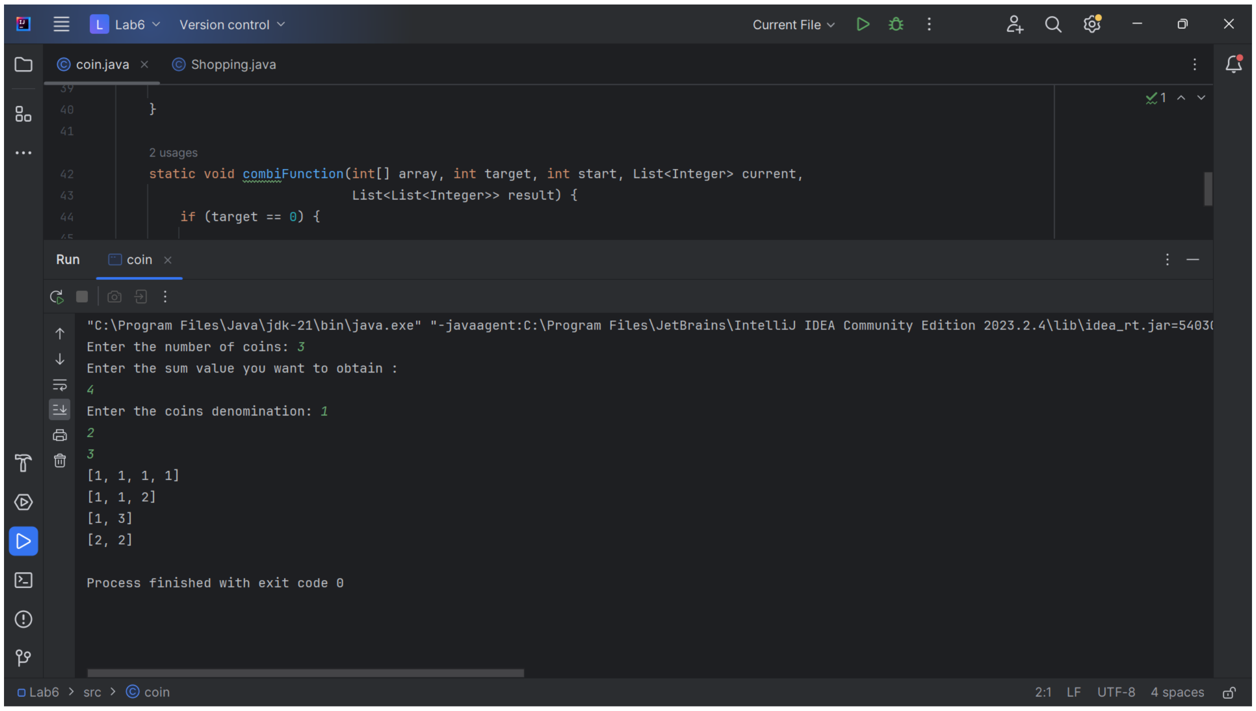
```java
import java.util.*;


public class coin {

    public static void main(String[] args) throws InterruptedException {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter the number of coins: ");
        int n = input.nextInt();
        int[] coinsArr = new int[n];

        System.out.println("Enter the sum value you want to obtain : ");
        int sum = input.nextInt();
        Thread t1 = new Thread(() -> {

            synchronized (coinsArr) {

                System.out.print("Enter the coins denomination: ");
                for (int i = 0; i < n; i++) {
                    coinsArr[i] = input.nextInt();
                }

            }
        });
        List<List<Integer>> v = new ArrayList<>();
        Thread t2 = new Thread(() -> {
            synchronized (coinsArr) {
                combiFunction(coinsArr, sum, start: 0, new ArrayList<>(), v);
```


```java
                combiFunction(coinsArr, sum, start: 0, new ArrayList<>(), v);
            }
        });

        t1.start();
        t2.start();
        t1.join();
        t2.join();
        for (List<Integer> combination : v) {
            System.out.println(combination);
        }

    }

    2 usages
    static void combiFunction(int[] array, int target, int start, List<Integer> current,
                              List<List<Integer>> result) {
        if (target == 0) {

            result.add(new ArrayList<>(current));
            return;
        }

        for (int i = start; i < array.length; i++) {

            if (array[i] <= target) {
                current.add(array[i]);
                combiFunction(array, target: target - array[i], i, current, result);
```



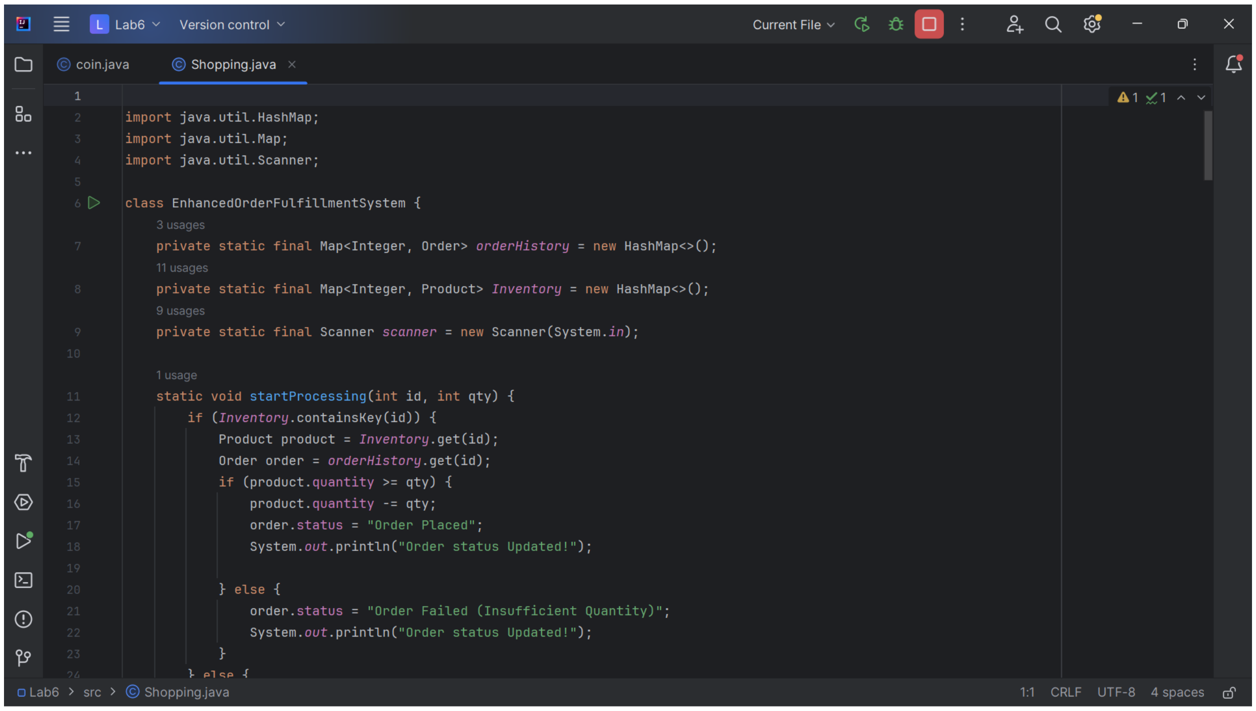
```java
    }

    static void combiFunction(int[] array, int target, int start, List<Integer> current,
                              List<List<Integer>> result) {
        if (target == 0) {

            result.add(new ArrayList<>(current));
            return;
        }

        for (int i = start; i < array.length; i++) {

            if (array[i] <= target) {
                current.add(array[i]);
                combiFunction(array, target - array[i], i, current, result);
                current.remove(current.size() - 1);
            }
        }
    }

}
```

Output:

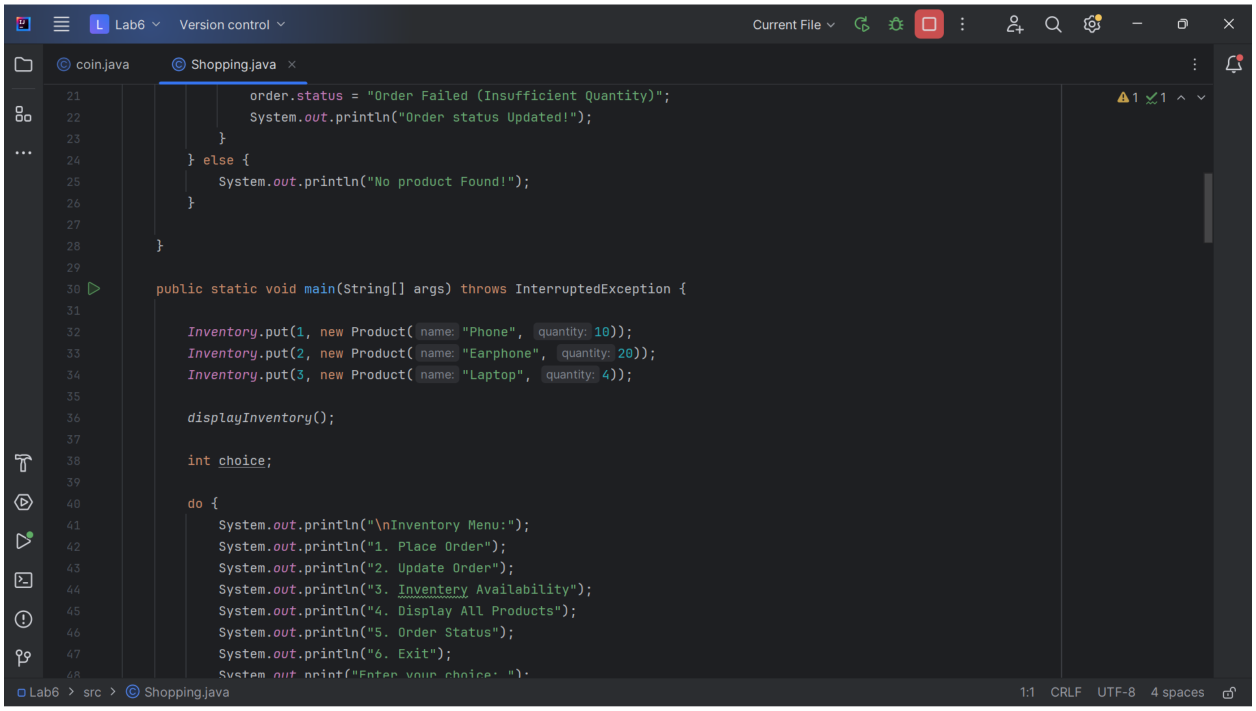


```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2023.2.4\lib\idea_rt.jar=5403
Enter the number of coins: 3
Enter the sum value you want to obtain :
4
Enter the coins denomination: 1
2
3
[1, 1, 1, 1]
[1, 1, 2]
[1, 3]
[2, 2]

Process finished with exit code 0
```

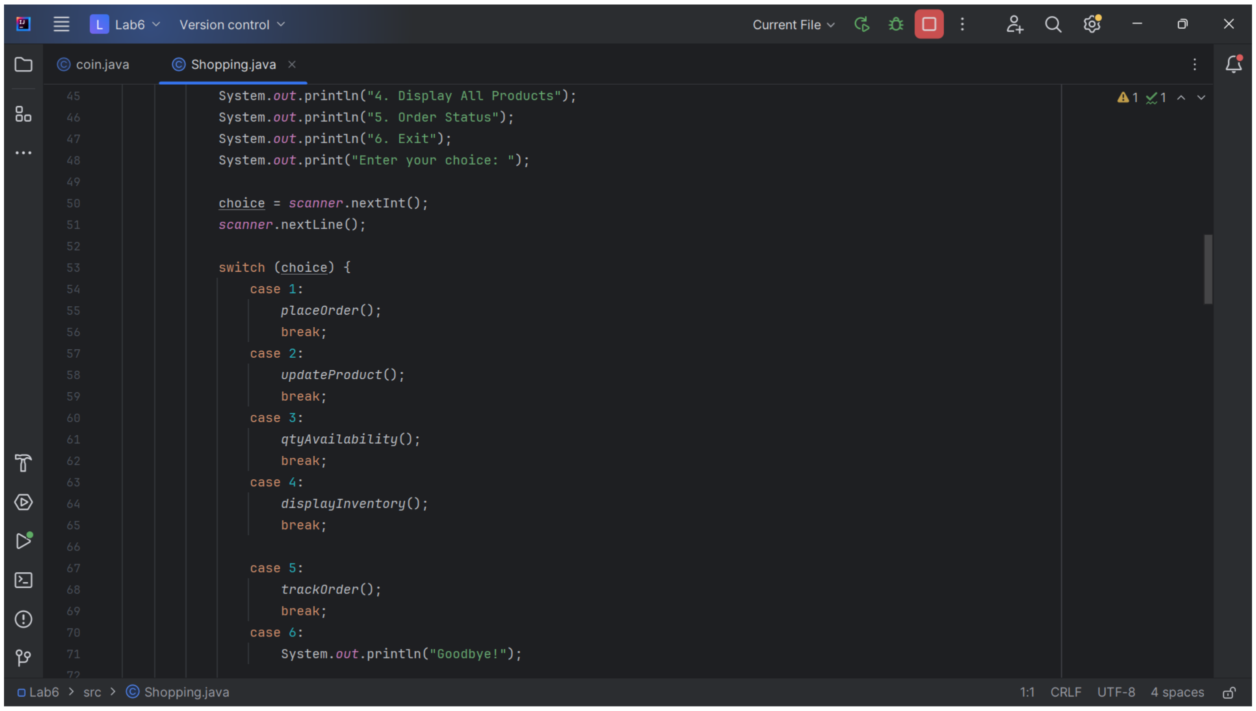
Program 2:



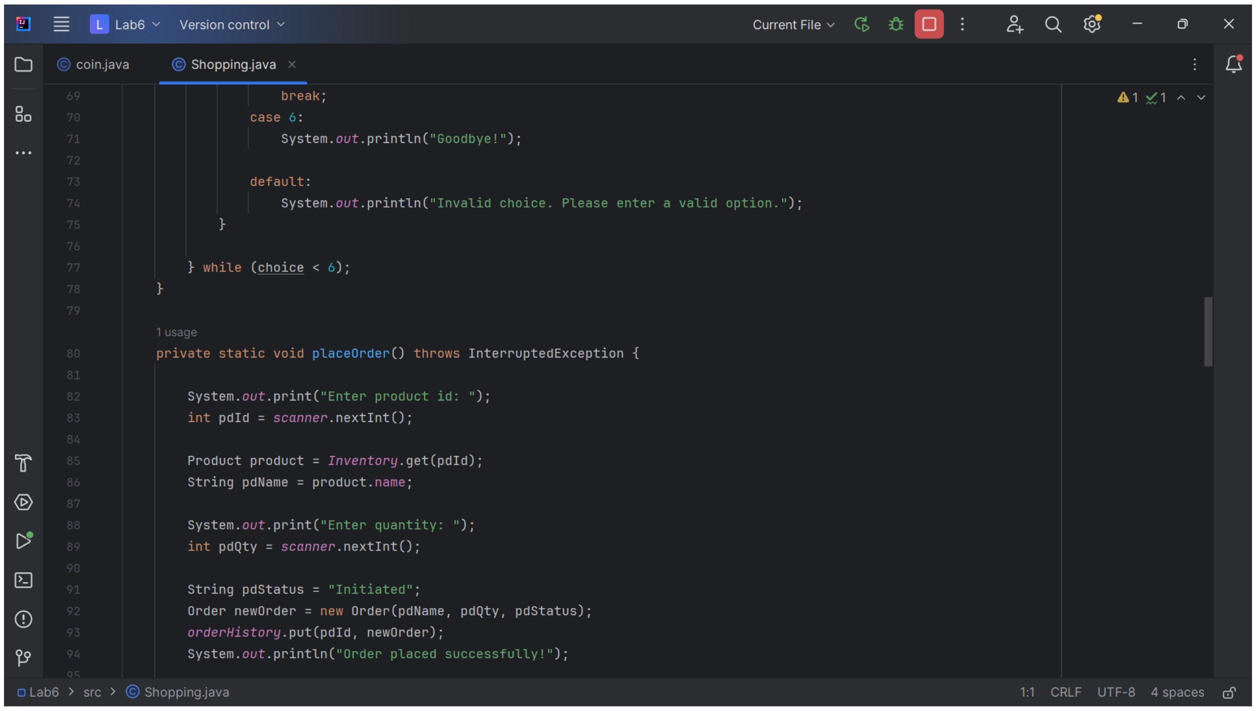
```java

import java.util.HashMap;
import java.util.Map;
import java.util.Scanner;

class EnhancedOrderFulfillmentSystem {
    3 usages
    private static final Map<Integer, Order> orderHistory = new HashMap<>();
    11 usages
    private static final Map<Integer, Product> Inventory = new HashMap<>();
    9 usages
    private static final Scanner scanner = new Scanner(System.in);

    1 usage
    static void startProcessing(int id, int qty) {
        if (Inventory.containsKey(id)) {
            Product product = Inventory.get(id);
            Order order = orderHistory.get(id);
            if (product.quantity >= qty) {
                product.quantity -= qty;
                order.status = "Order Placed";
                System.out.println("Order status Updated!");

            } else {
                order.status = "Order Failed (Insufficient Quantity)";
                System.out.println("Order status Updated!");
            }
```



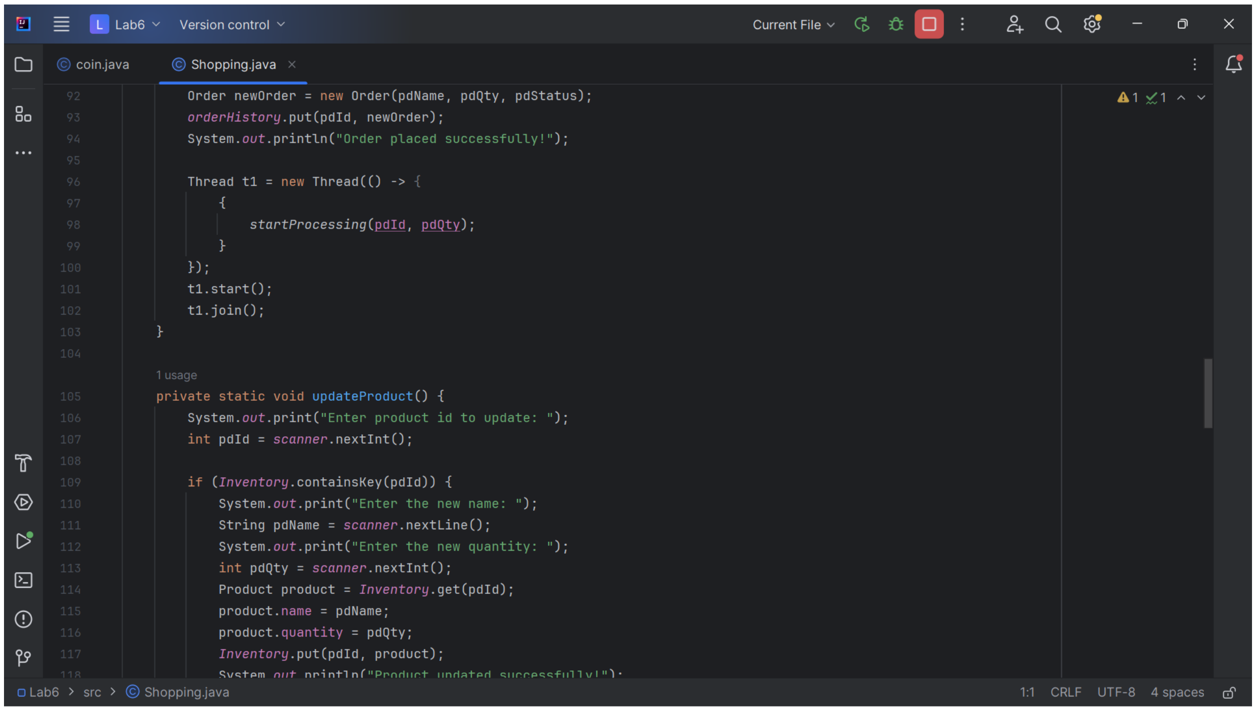
```java
                order.status = "Order Failed (Insufficient Quantity)";
                System.out.println("Order status Updated!");
            }
        } else {
            System.out.println("No product Found!");
        }

    }

    public static void main(String[] args) throws InterruptedException {

        Inventory.put(1, new Product( name: "Phone",  quantity: 10));
        Inventory.put(2, new Product( name: "Earphone",  quantity: 20));
        Inventory.put(3, new Product( name: "Laptop",  quantity: 4));

        displayInventory();

        int choice;

        do {
            System.out.println("\nInventory Menu:");
            System.out.println("1. Place Order");
            System.out.println("2. Update Order");
            System.out.println("3. Inventery Availability");
            System.out.println("4. Display All Products");
            System.out.println("5. Order Status");
            System.out.println("6. Exit");
```



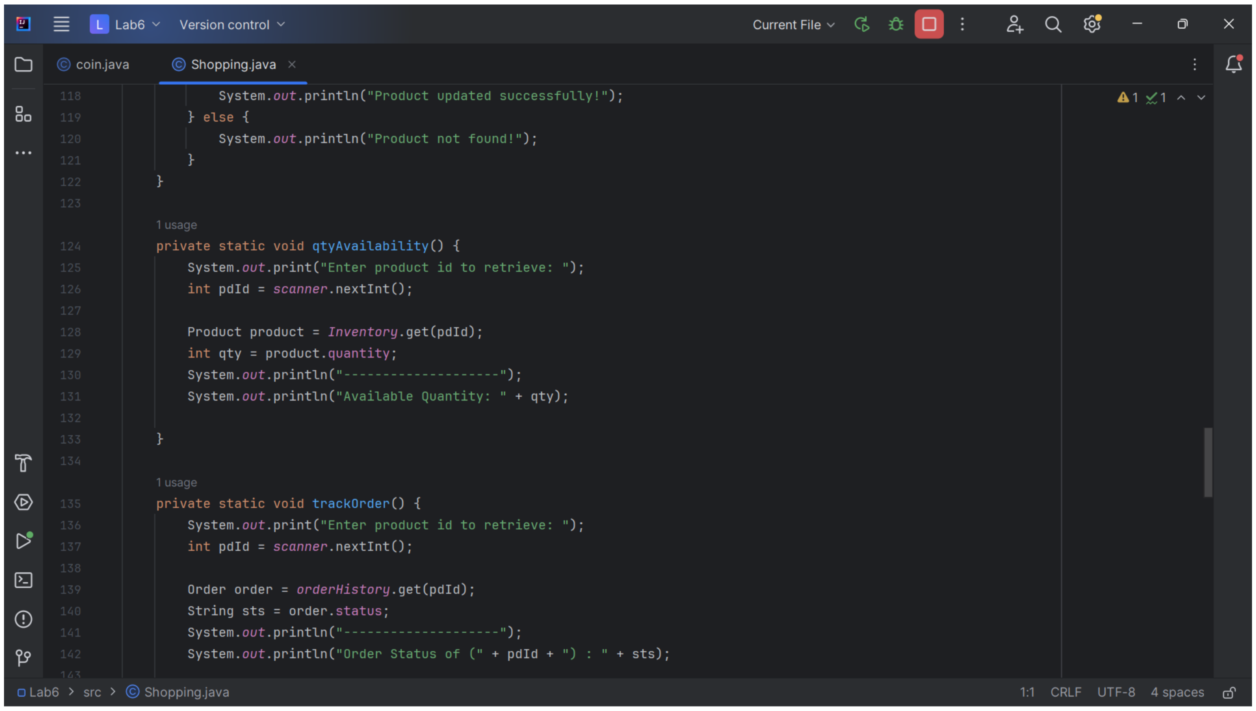
```java
            System.out.println("4. Display All Products");
            System.out.println("5. Order Status");
            System.out.println("6. Exit");
            System.out.print("Enter your choice: ");

            choice = scanner.nextInt();
            scanner.nextLine();

            switch (choice) {
                case 1:
                    placeOrder();
                    break;
                case 2:
                    updateProduct();
                    break;
                case 3:
                    qtyAvailability();
                    break;
                case 4:
                    displayInventory();
                    break;

                case 5:
                    trackOrder();
                    break;
                case 6:
                    System.out.println("Goodbye!");
```



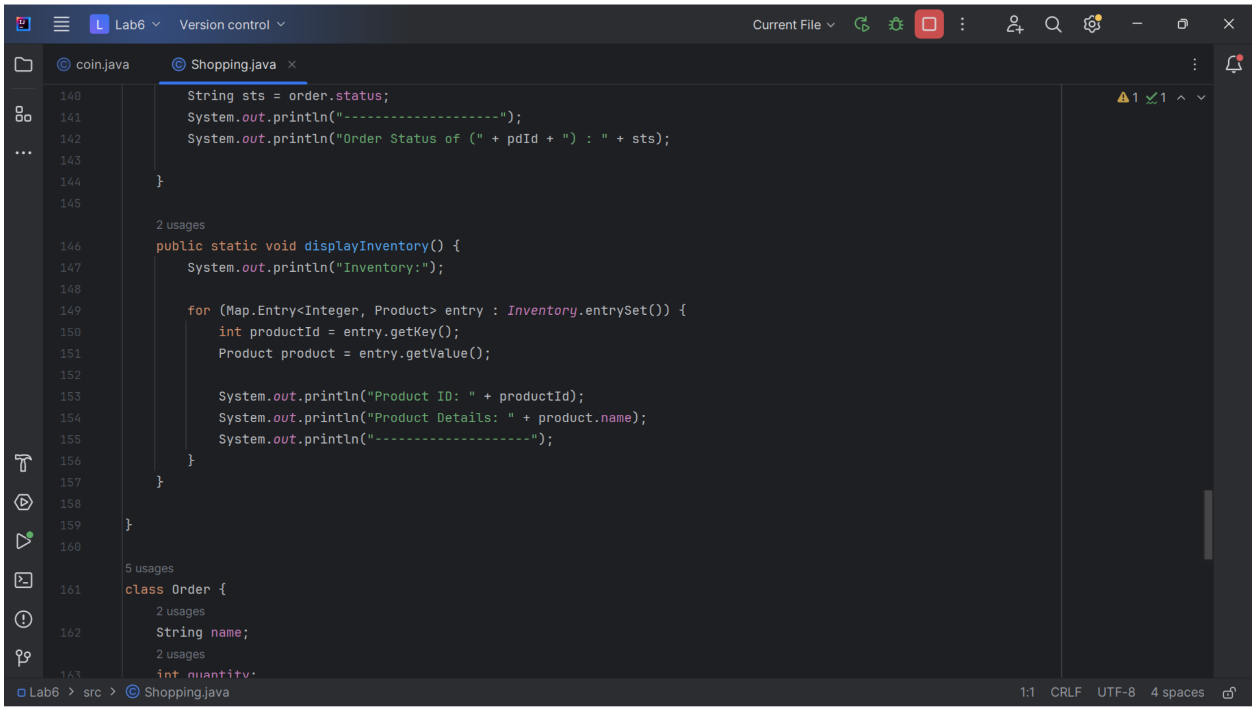
```java
                    break;
                case 6:
                    System.out.println("Goodbye!");

                default:
                    System.out.println("Invalid choice. Please enter a valid option.");
            }

        } while (choice < 6);
    }

    1 usage
    private static void placeOrder() throws InterruptedException {

        System.out.print("Enter product id: ");
        int pdId = scanner.nextInt();

        Product product = Inventory.get(pdId);
        String pdName = product.name;

        System.out.print("Enter quantity: ");
        int pdQty = scanner.nextInt();

        String pdStatus = "Initiated";
        Order newOrder = new Order(pdName, pdQty, pdStatus);
        orderHistory.put(pdId, newOrder);
        System.out.println("Order placed successfully!");
```



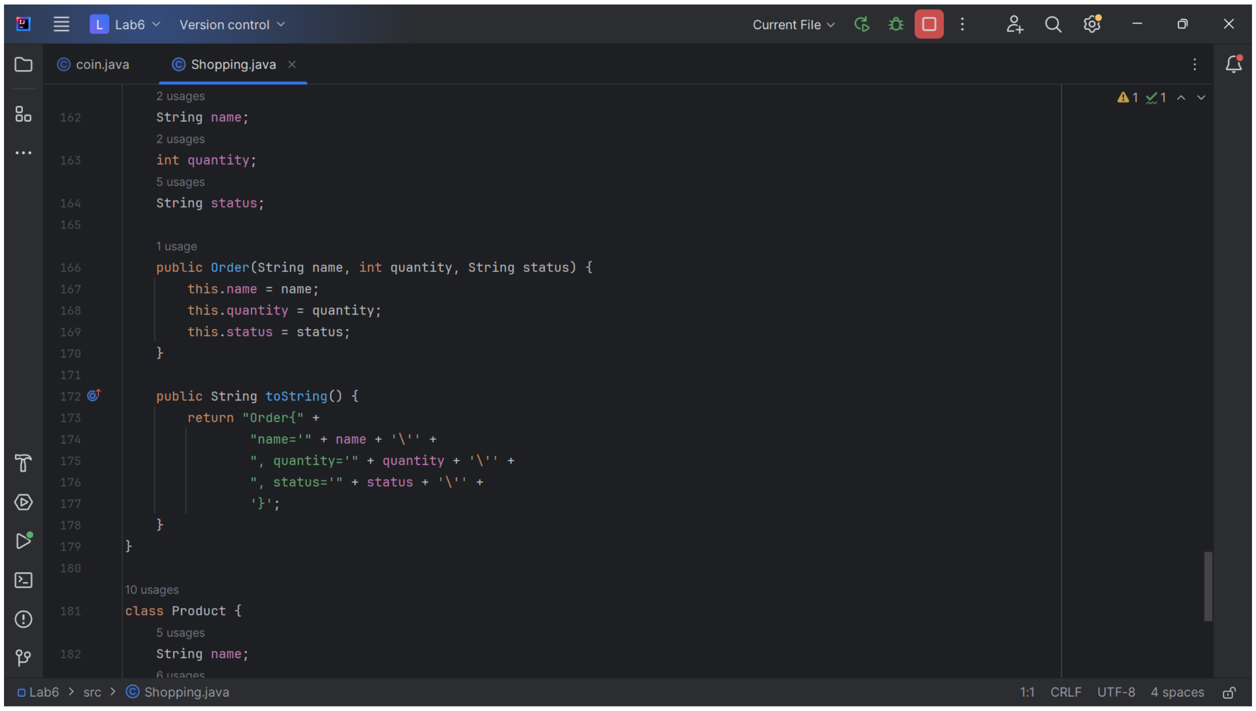
```java
        Order newOrder = new Order(pdName, pdQty, pdStatus);
        orderHistory.put(pdId, newOrder);
        System.out.println("Order placed successfully!");

        Thread t1 = new Thread(() -> {
            {
                startProcessing(pdId, pdQty);
            }
        });
        t1.start();
        t1.join();
    }

    1 usage
    private static void updateProduct() {
        System.out.print("Enter product id to update: ");
        int pdId = scanner.nextInt();

        if (Inventory.containsKey(pdId)) {
            System.out.print("Enter the new name: ");
            String pdName = scanner.nextLine();
            System.out.print("Enter the new quantity: ");
            int pdQty = scanner.nextInt();
            Product product = Inventory.get(pdId);
            product.name = pdName;
            product.quantity = pdQty;
            Inventory.put(pdId, product);
            System.out.println("Product updated successfully!");
```



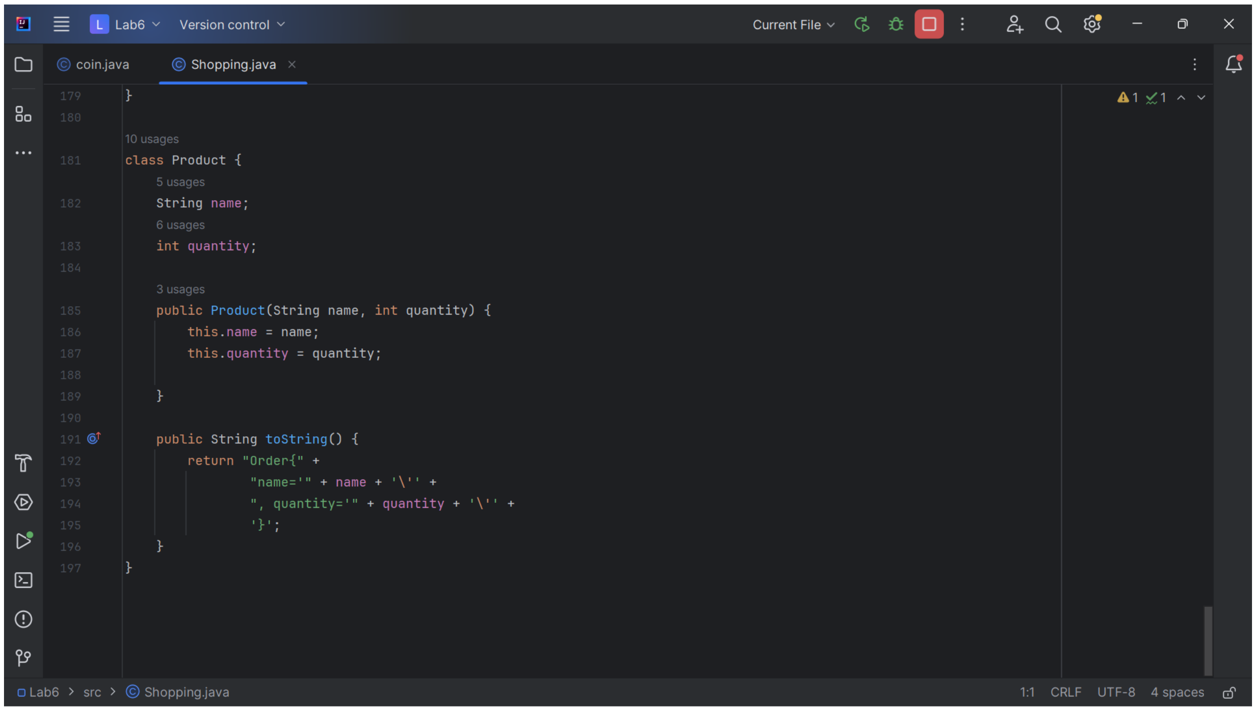
```java
            System.out.println("Product updated successfully!");
        } else {
            System.out.println("Product not found!");
        }
    }

    1 usage
    private static void qtyAvailability() {
        System.out.print("Enter product id to retrieve: ");
        int pdId = scanner.nextInt();

        Product product = Inventory.get(pdId);
        int qty = product.quantity;
        System.out.println("--------------------");
        System.out.println("Available Quantity: " + qty);

    }

    1 usage
    private static void trackOrder() {
        System.out.print("Enter product id to retrieve: ");
        int pdId = scanner.nextInt();

        Order order = orderHistory.get(pdId);
        String sts = order.status;
        System.out.println("--------------------");
        System.out.println("Order Status of (" + pdId + ") : " + sts);
```



```java
        String sts = order.status;
        System.out.println("--------------------");
        System.out.println("Order Status of (" + pdId + ") : " + sts);

    }

    public static void displayInventory() {
        System.out.println("Inventory:");

        for (Map.Entry<Integer, Product> entry : Inventory.entrySet()) {
            int productId = entry.getKey();
            Product product = entry.getValue();

            System.out.println("Product ID: " + productId);
            System.out.println("Product Details: " + product.name);
            System.out.println("--------------------");
        }
    }

}

class Order {
    String name;
    int quantity;
```

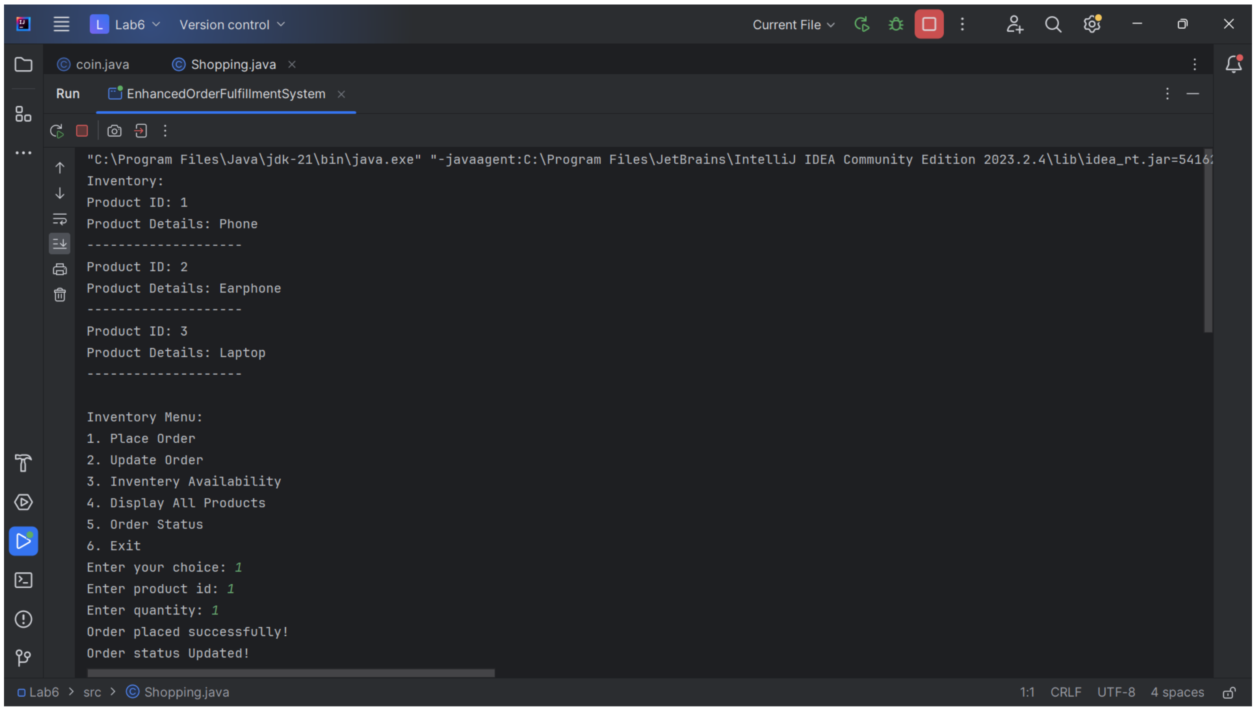


```java
    String name;
    int quantity;
    String status;

    public Order(String name, int quantity, String status) {
        this.name = name;
        this.quantity = quantity;
        this.status = status;
    }

    public String toString() {
        return "Order{" +
                "name='" + name + '\'' +
                ", quantity='" + quantity + '\'' +
                ", status='" + status + '\'' +
                '}';
    }
}

class Product {
    String name;
```



```
}

class Product {
    String name;
    int quantity;

    public Product(String name, int quantity) {
        this.name = name;
        this.quantity = quantity;

    }

    public String toString() {
        return "Order{" +
                "name='" + name + '\'' +
                ", quantity='" + quantity + '\'' +
                '}';
    }
}
```

Output:



```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2023.2.4\lib\idea_rt.jar=5416
Inventory:
Product ID: 1
Product Details: Phone
--------------------
Product ID: 2
Product Details: Earphone
--------------------
Product ID: 3
Product Details: Laptop
--------------------

Inventory Menu:
1. Place Order
2. Update Order
3. Inventery Availability
4. Display All Products
5. Order Status
6. Exit
Enter your choice: 1
Enter product id: 1
Enter quantity: 1
Order placed successfully!
Order status Updated!
```