Phase 5: Apex Programming (Developer)

🎯 Objective

Implement custom business logic using Apex to handle complex scenarios such as bulk order processing, inventory updates, notifications, and integrations that can't be handled through Flows or Process Builder alone.

1. Apex Classes & Objects

• OrderHandler.cls → Central class for order processing logic.

   - Methods: createOrder(), updateOrderStatus(), calculateInventory()

• InventoryManager.cls → Handle inventory calculations and stock updates.

• NotificationService.cls → Send custom email/SMS notifications.

• DeliveryScheduler.cls → Calculate delivery dates and assign carriers.

2. Apex Triggers

Before Insert / Update

Order Trigger:

• Before insert: Assign unique Order ID, validate stock availability.

• Before update: Check if status changes to 'Dispatched' → ensure delivery record exists.

After Insert / Update

• After insert: Reduce inventory quantity based on ordered products.

• After insert: Send confirmation email to retailer.

• After update: If Order Status = Delivered → Notify retailer + create task for sales follow-up.

OrderHandler.apxc ✕ | InventoryManager.apxc ✕ | NotificationService.apxc ✕ | DeliveryScheduler.apxc ✕ | OrderTrigger.apxt ✕ | OrderHandlerTest.apxc ✕

Code Coverage: None ▾   API Version: 64 ▾

```
1  ▼ public class OrderHandler {
2
3        // Create a new order and return it
4  ▼     public Order createOrder(Order ord){
5  ▼         if(ord.Status == null){
6                 ord.Status = 'Draft';
7             }
8             insert ord;
9             return ord;
10        }
11
```

| Logs | Tests | Checkpoints | Query Editor | View State | Progress | Problems |

Name	Line	Problem