Phase 5: Apex Programming (Developer)

1. Apex Classes & Objects

Apex Classes are used to encapsulate business logic.

- **LeadAssignmentHandler** → Assign leads dynamically based on territory, budget, or interest score.
- **VisitSchedulerService** → Creates Property Visit records automatically when customers confirm availability.
- **DealManager** → Handles custom logic for offer negotiation and closing.
- **BookingService** → Validates booking amount and creates booking records with linked documents.

2. Apex Triggers

Triggers allow us to run logic when records are created, updated, or deleted.

Lead Trigger

- **Before Insert/Update:**
  - Validate Interest Score (0–100).
  - Auto-set Territory based on City/State.
- **After Insert:**
  - Auto-assign Lead to an Agent.
  - Send “Thank You for Inquiry” email to the customer.

Property Trigger

- **After Update:**
  - If Property Status = “Sold” → lock the record for editing.

Deal Trigger

- **After Insert:**
  - Auto-create related Booking record with default status = “Pending.”

Booking Trigger

- **Before Insert:**
  - Ensure Booking Amount > 0.

- **After Insert:**
    - Send confirmation email + task to Sales Manager for approval.

3. Trigger Design Pattern

Instead of writing multiple triggers per object, we'll use a **Handler Class Pattern**:

- One trigger per object (e.g., LeadTrigger)
- That trigger calls a handler class (e.g., LeadHandler)
- Logic lives in the handler → clean, maintainable code

4. SOQL & SOSL Queries

We'll use queries to fetch records dynamically:

- **SOQL Example (Properties):**
- List<Property__c> availableProps = [
- SELECT Id, Name, Price__c, Location__c
- FROM Property__c
- WHERE Status__c = 'Available' AND Price__c <= 5000000
-];

Find all available properties under 50 Lakhs.

- **SOSL Example (Leads):**
- List<List<SObject>> searchResults = [
- FIND 'Hyderabad' IN ALL FIELDS RETURNING Lead(Name, Email, City__c)
-];

5. Collections: List, Set, Map

- **List:** Store multiple properties from a query.
- **Set:** Track unique cities of interest from leads.
- **Map:** Map LeadId → Assigned AgentId for bulk assignments.

💻 **6. Control Statements**

- If/Else logic for **lead scoring thresholds.**
- For Loops to iterate over **property visit records.**
- Switch statements for **deal stages (Enquiry, Negotiation, Closure).**

7. Asynchronous Apex

Batch Apex

- Update thousands of Property records (e.g., mark expired listings as “Inactive”).

Queueable Apex

- Queue up long-running tasks like document verification after booking.

Scheduled Apex

- Run nightly job at 11 PM: “Send daily performance summary to Sales Managers.”

Future Methods

- Call external APIs (e.g., property aggregator integration) without delaying the user.

8. Exception Handling

- Catch invalid booking entries → show user-friendly error messages.
- Log system errors into a **Custom Error Log object** for Admin review.

9. Test Classes

- Required to deploy Apex code (≥75% code coverage).
- Write test methods for:
    - Lead assignment logic
    - Visit scheduling trigger
    - Deal creation with linked booking

Example:

```
@isTest
private class TestLeadAssignment {
   @isTest static void testLeadTerritoryAssignment() {
      Lead testLead = new Lead(FirstName='John', LastName='Doe', City__c='Hyderabad', Status='New');
      insert testLead;
      System.assertEquals('South Territory', testLead.Territory__c);
   }
}
```