Phase 5: Apex Programming (Developer)

1. Classes & Objects

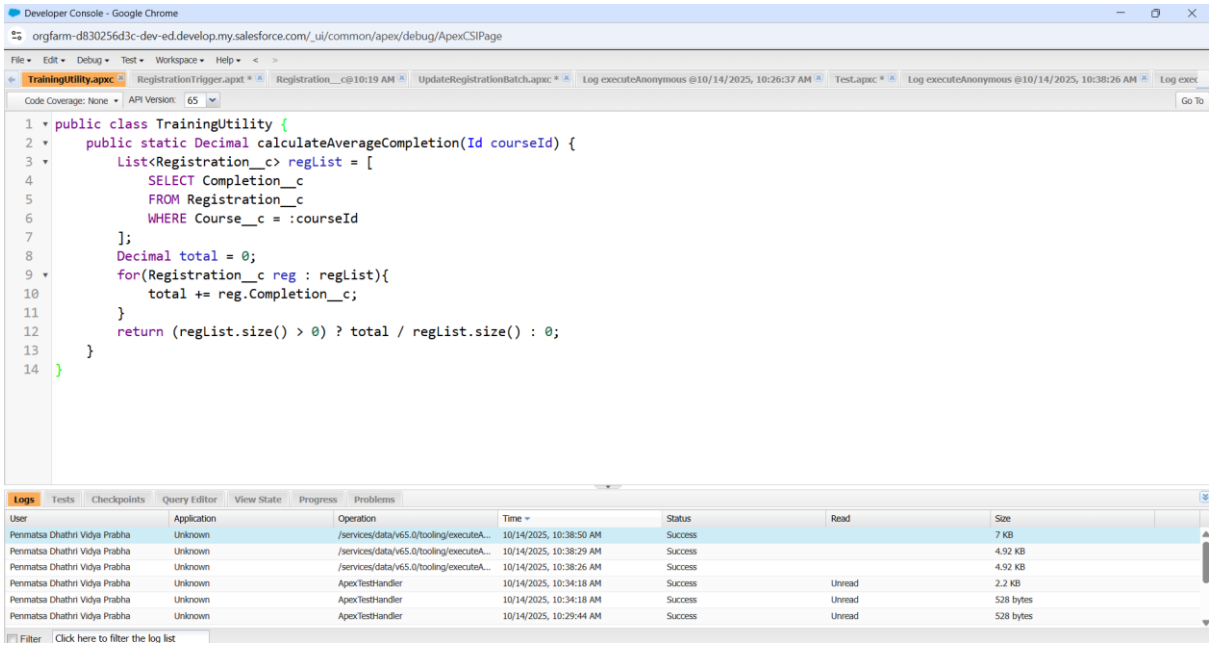
Use Case:

In the Corporate Training CRM, classes are used to encapsulate logic related to employees and course registrations.

Implementation:

Class Name: TrainingUtility

```
public class TrainingUtility {
    public static Decimal calculateAverageCompletion(Id courseId) {
        List<Registration__c> regList = [
            SELECT Completion__c
            FROM Registration__c
            WHERE Course__c = :courseId
        ];
        Decimal total = 0;
        for(Registration__c reg : regList){
            total += reg.Completion__c;
        }
        return (regList.size() > 0) ? total / regList.size() : 0;
    }
}
```



2. Apex Trigger

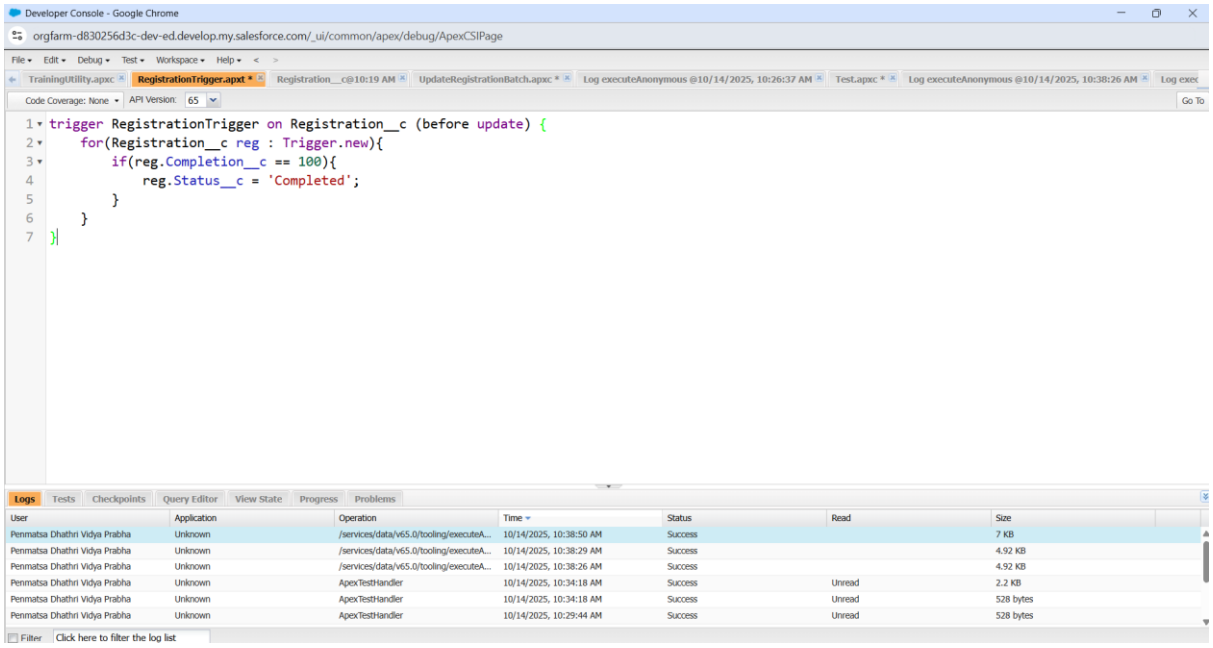
Use Case:

Automatically update the Status field in the Registration record whenever the Completion (%) reaches 100.
This ensures the data consistency even if the Flow fails or is bypassed.

Implementation:

Trigger Name: RegistrationTrigger

```
trigger RegistrationTrigger on Registration__c (before update) {
    for(Registration__c reg : Trigger.new){
        if(reg.Completion__c == 100){
            reg.Status__c = 'Completed';
        }
    }
}
```



3. SOQL (Salesforce Object Query Language)

Use Case:

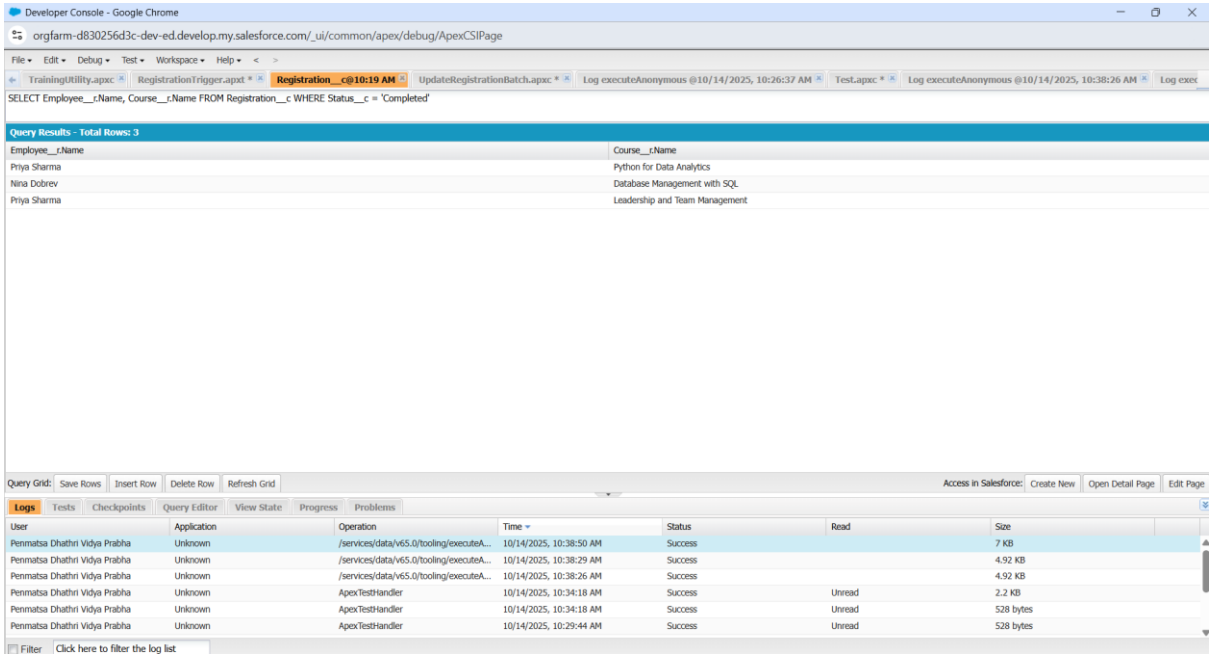
To retrieve all employees who have completed their training courses.

Implementation:

SELECT Employee__r.Name, Course__r.Course_Name__c

FROM Registration__c

WHERE Status__c = 'Completed'



4. Control Statements

Use Case:

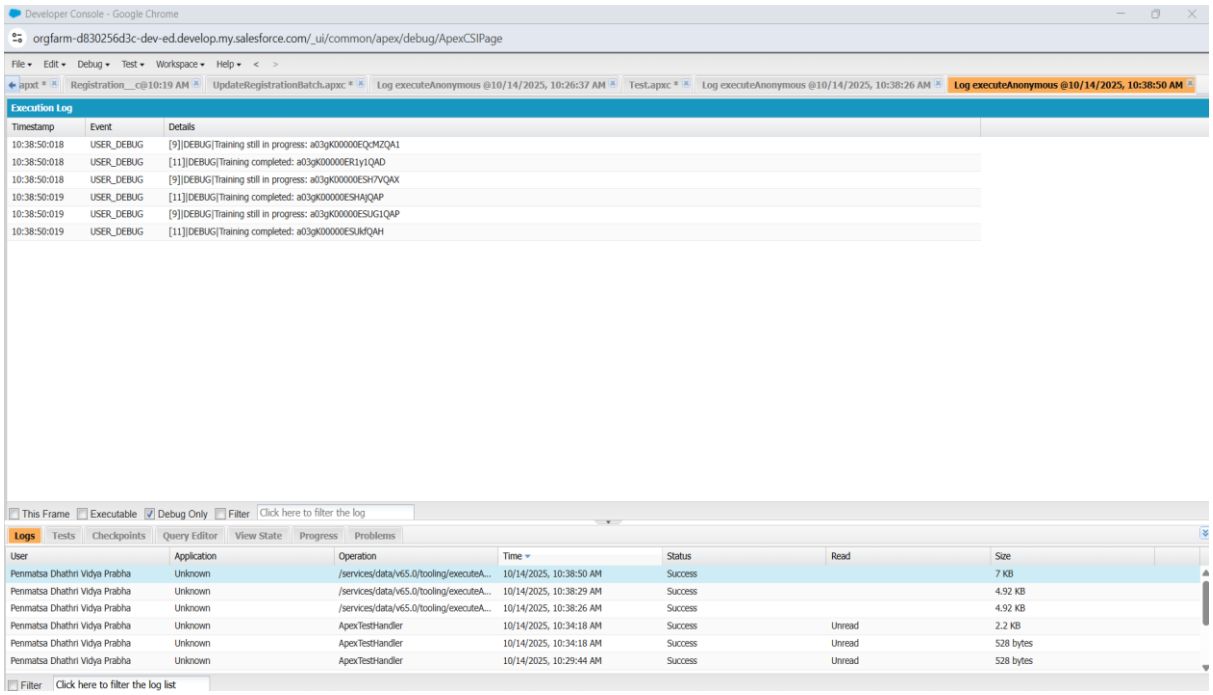
Control statements help apply conditional logic like IF, FOR, WHILE loops within Apex code.

Implementation:

List<Registration__c> regList = [SELECT Id, Completion__c FROM Registration__c];

// Iterate through each record and apply logic

```
for (Registration__c reg : regList) {
    if (reg.Completion__c < 100) {
        System.debug('Training still in progress: ' + reg.Id);
    } else {
        System.debug('Training completed: ' + reg.Id);
    }
}
```



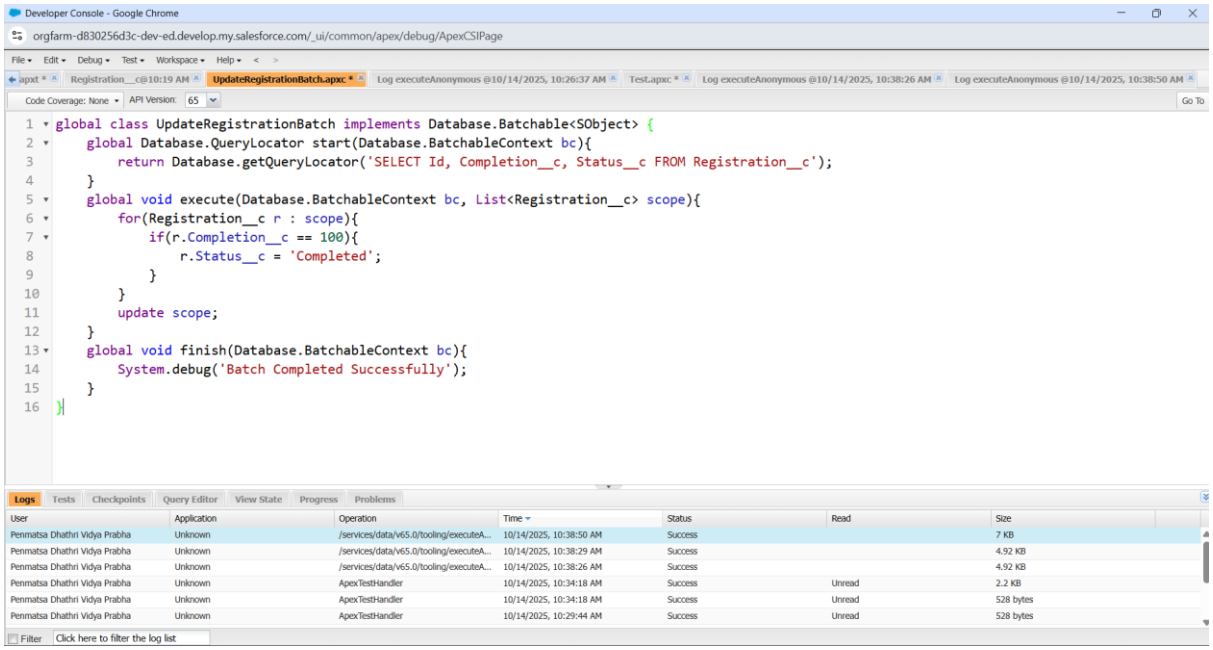
5. Batch Apex

Use Case:

To update the Status of multiple Registration records in bulk (useful for large datasets).

Implementation:

```
global class UpdateRegistrationBatch implements Database.Batchable<SObject> {
    global Database.QueryLocator start(Database.BatchableContext bc){
        return Database.getQueryLocator('SELECT Id, Completion__c, Status__c FROM Registration__c');
    }
    global void execute(Database.BatchableContext bc, List<Registration__c> scope){
        for(Registration__c r : scope){
            if(r.Completion__c == 100){
                r.Status__c = 'Completed';
            }
        }
        update scope;
    }
    global void finish(Database.BatchableContext bc){
        System.debug('Batch Completed Successfully');
    }
}
```



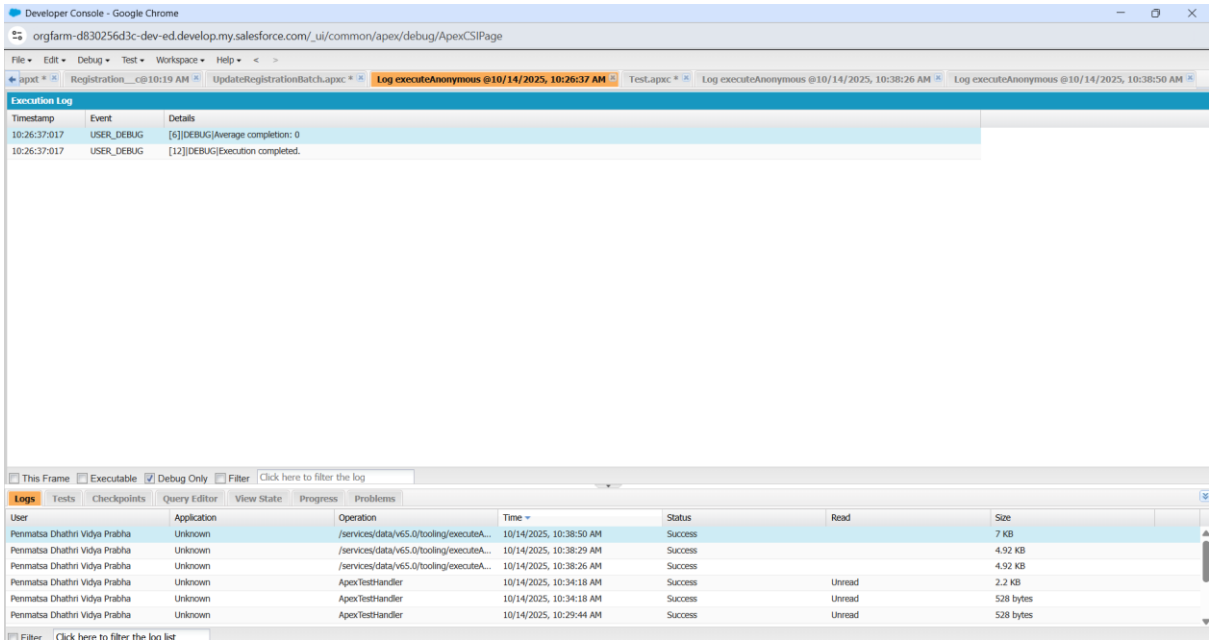
6. Exception Handling

Use Case:

To manage runtime errors (e.g., division by zero, null pointer).

Implementation:

```
try {
    // Intentionally causing a null pointer exception
    Decimal avg = TrainingUtility.calculateAverageCompletion(null);
    System.debug('Average completion: ' + avg);
}
catch (Exception e) {
    System.debug('⚠ Error occurred: ' + e.getMessage());
}
finally {
    System.debug('Execution completed.');
}
```



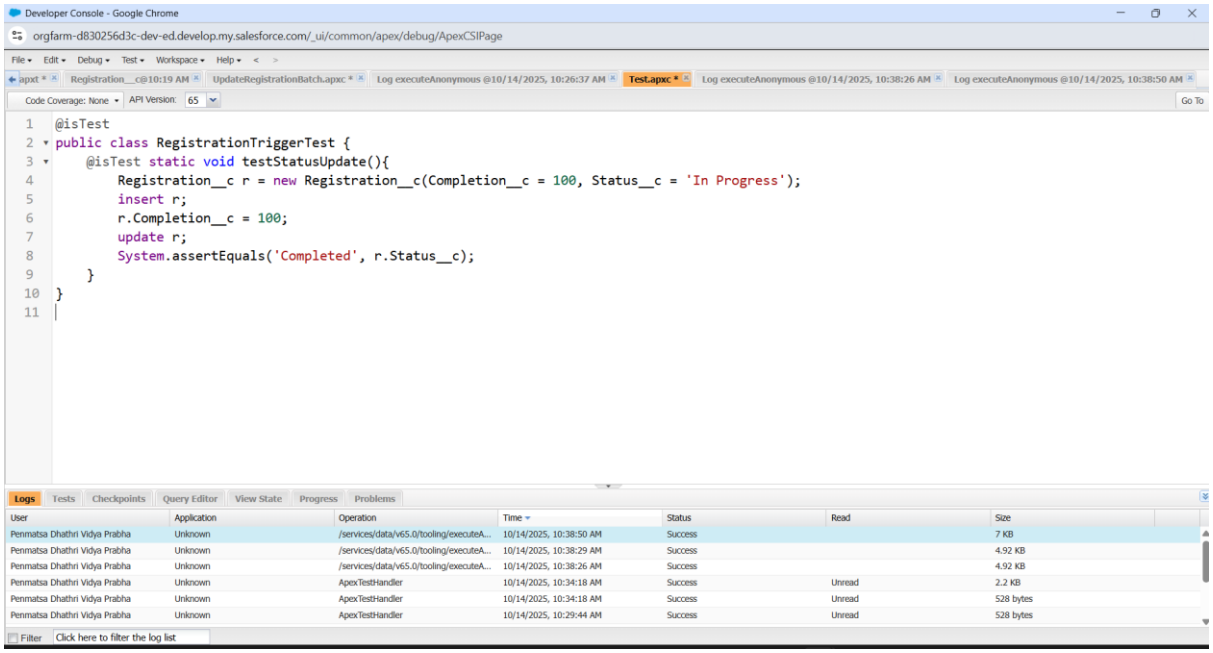
7. Test Class

Use Case:

To ensure all Apex logic runs correctly and passes Salesforce deployment requirements.

Implementation:

```
@isTest
public class RegistrationTriggerTest {
    @isTest static void testStatusUpdate(){
        Registration__c r = new Registration__c(Completion__c = 100, Status__c = 'In Progress');
        insert r;
        r.Completion__c = 100;
        update r;
        System.assertEquals('Completed', r.Status__c);
    }
}
```


Developer Console - Google Chrome
orgfarm-d830256d3c-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File Edit Debug Test Workspace Help
apxt * Registration__c@10:19 AM UpdateRegistrationBatch.apxc * Log executeAnonymous @10/14/2025, 10:26:37 AM Test.apxc * Log executeAnonymous @10/14/2025, 10:38:26 AM Log executeAnonymous @10/14/2025, 10:38:50 AM
Code Coverage: None API Version: 65 Go To
1 @isTest
2 public class RegistrationTriggerTest {
3 @isTest static void testStatusUpdate(){
4 Registration__c r = new Registration__c(Completion__c = 100, Status__c = 'In Progress');
5 insert r;
6 r.Completion__c = 100;
7 update r;
8 System.assertEquals('Completed', r.Status__c);
9 }
10 }
11
Logs Tests Checkpoints Query Editor View State Progress Problems
User Application Operation Time Status Read Size
Penmatsa Dhathri Vidya Prabha Unknown /services/data/v65.0/tooling/executeA... 10/14/2025, 10:38:50 AM Success 7 KB
Penmatsa Dhathri Vidya Prabha Unknown /services/data/v65.0/tooling/executeA... 10/14/2025, 10:38:29 AM Success 4.92 KB
Penmatsa Dhathri Vidya Prabha Unknown /services/data/v65.0/tooling/executeA... 10/14/2025, 10:38:26 AM Success 4.92 KB
Penmatsa Dhathri Vidya Prabha Unknown ApexTestHandler 10/14/2025, 10:34:18 AM Success Unread 2.2 KB
Penmatsa Dhathri Vidya Prabha Unknown ApexTestHandler 10/14/2025, 10:34:18 AM Success Unread 528 bytes
Penmatsa Dhathri Vidya Prabha Unknown ApexTestHandler 10/14/2025, 10:29:44 AM Success Unread 528 bytes
Filter Click here to filter the log list