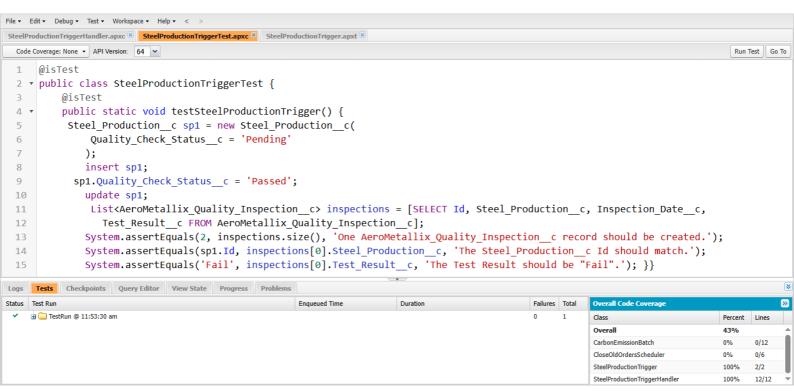


```
SteelProductionTriggerHandler.apxc X SteelProductionTriggerTest.apxc X SteelProductionTrigger.apxt X
 Code Coverage: None ▼ API Version: 64 ▼
                                                                                                                                                  Go To
 1 • public class SteelProductionTriggerHandler {
 2 🔻
           public static void handleAfterUpdate(List<Steel_Production__c> newRecords, Map<Id, Steel_Production__c> oldRecords) {
 3
               List<AeroMetallix_Quality_Inspection__c> inspections = new List<AeroMetallix_Quality_Inspection__c>();
 4 •
                for (Steel_Production__c sp : newRecords) {
 5
                     Steel_Production__c oldRecord = oldRecords.get(sp.Id);
                            if (sp.Quality_Check_Status__c == 'Passed' && oldRecord.Quality_Check_Status__c != 'Passed') {
 6 ▼
                                        AeroMetallix_Quality_Inspection__c qi = new AeroMetallix_Quality_Inspection__c(
                                        Steel_Production__c = sp.Id,
 8
                                        Inspection_Date__c = Date.today(),
Test_Result__c = 'Pending' );
 9
 10
                                        inspections.add(qi);
 11
 12
                               }
 13
 14 🔻
                  if (!inspections.isEmpty()) {
 15
                      insert inspections;
 16
 17
            }
       }
 18
```

File • Edit • Debug • Test • Workspace • Help • < >



```
File • Edit • Debug • Test • Workspace • Help • < >
SteelProductionTriggerHandler.apxc 🗷 SteelProductionTriggerTest.apxc 🗷 SteelProductionTrigger.apxt 🗷 CarbonEmissionBatch.apxc 🗵
 1 v global class CarbonEmissionBatch implements Database.Batchable<SObject> {
          global Database.QueryLocator start(Database.BatchableContext BC) {
              return Database.getQueryLocator('SELECT Plant_Location__c, SUM(Production_Quantity__c) FROM Steel_Production__c WHERE Produ
          global void execute(Database.BatchableContext BC, List<AggregateResult> records) {
  5 v
              List<AeroMetallix_Plant__c> plantsToUpdate = new List<AeroMetallix_Plant__c>();
  6
  7 •
              for (AggregateResult ar : records) {
  8
                   AeroMetallix_Plant__c plant = new AeroMetallix_Plant__c(
                       Id = (Id) ar.get('Plant_Location__c'),
  9
                       Monthly_Carbon_Emissions__c = (Decimal) ar.get('expr0'));
 10
                   plantsToUpdate.add(plant);
 11
 12
              if (!plantsToUpdate.isEmpty()) {
 13 •
                   update plantsToUpdate;
 14
 15
 16
 17 ▼
          global void finish(Database.BatchableContext BC) {
 18
              System.debug('Carbon Emissions Calculation Completed.');
 19
 20 }
```

```
SteelProductionTriggerHandler.apxc 🗷 SteelProductionTriggerTest.apxc 🗷 SteelProductionTrigger.apxt 🗷 CarbonEmissionBatch.apxc 🗷 CloseOldOrdersScheduler.apxc * 🗵
Code Coverage: None ▼ API Version: 64 ▼
                                                                                                                                                       Go To
 1 • global class CloseOldOrdersScheduler implements Schedulable {
           global void execute(SchedulableContext sc) {
  3
               List<AeroMetallix_Order__c> oldOrders = [SELECT Id FROM AeroMetallix_Order__c WHERE Status__c = 'Processing'
                                                              AND Order_Date__c <= LAST_N_DAYS:60];
  4
  5 🔻
                for (AeroMetallix_Order__c ord : oldOrders) {
                    ord.Status__c = 'Canceled';
                if (!oldOrders.isEmpty()) {
  8 •
                    update oldOrders;
  10
  11
  12 }
Logs, Tests, and Problems ①
```

File • Edit • Debug • Test • Workspace • Help • < >