Milestone 12- Apex

Apex Overview

Apex is a strongly typed, object-oriented programming language that allows developers to execute flow and transaction control statements on the Lightning platform server in conjunction with calls to the Lightning Platform? API. Using syntax that looks like Java and acts like database stored procedures, Apex enables developers to add business logic to most system events, including button clicks, related record updates, and Visualforce pages. Apex code can be initiated by Web service requests and from triggers on objects.

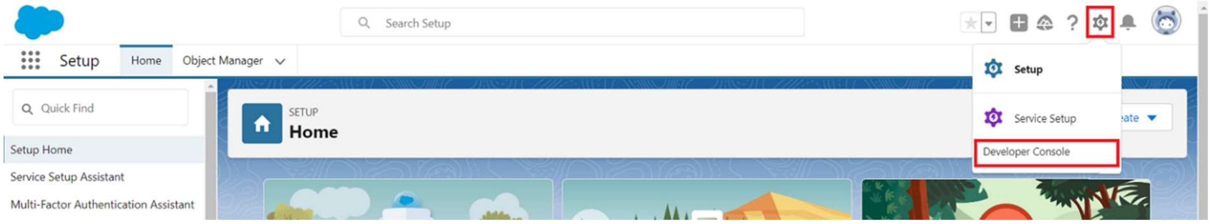
It is as similar as java i.e, it also supports OOP(Object oriented programming) like Classes, objects, methods.

Use Case 1:

Scenario: The Airline Management wants to make the passengers phone field as a mandatory field. So whenever a record gets inserted in a passenger's object that record should not get saved into the database, if the user missed the phone field.

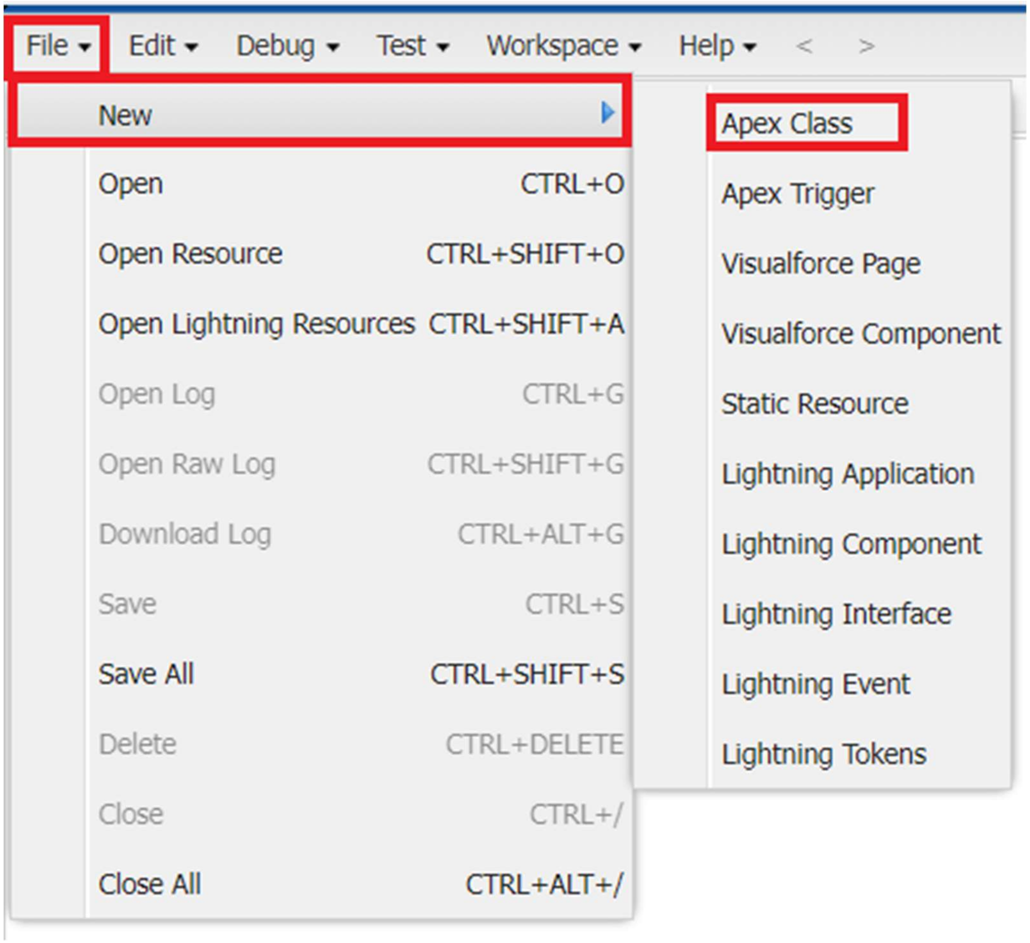
Activity 1 : Create an Apex Class

1. Go to Setup --> Click on the gear icon --> Select Developer Console.

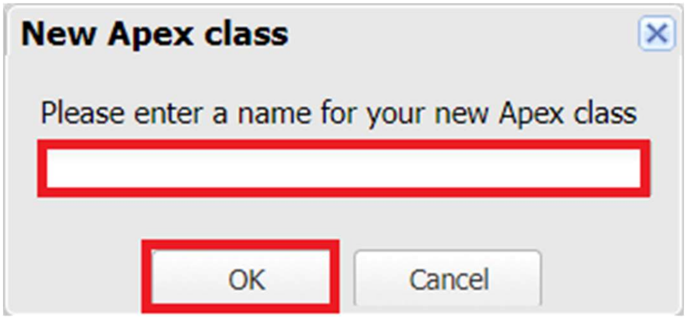


2. Then we can see the Developer console. Click on the developer console and you will navigate to a new console window.

3. To create a new Apex Class follow the below steps:
   Click on the file --> New --> Apex Class.

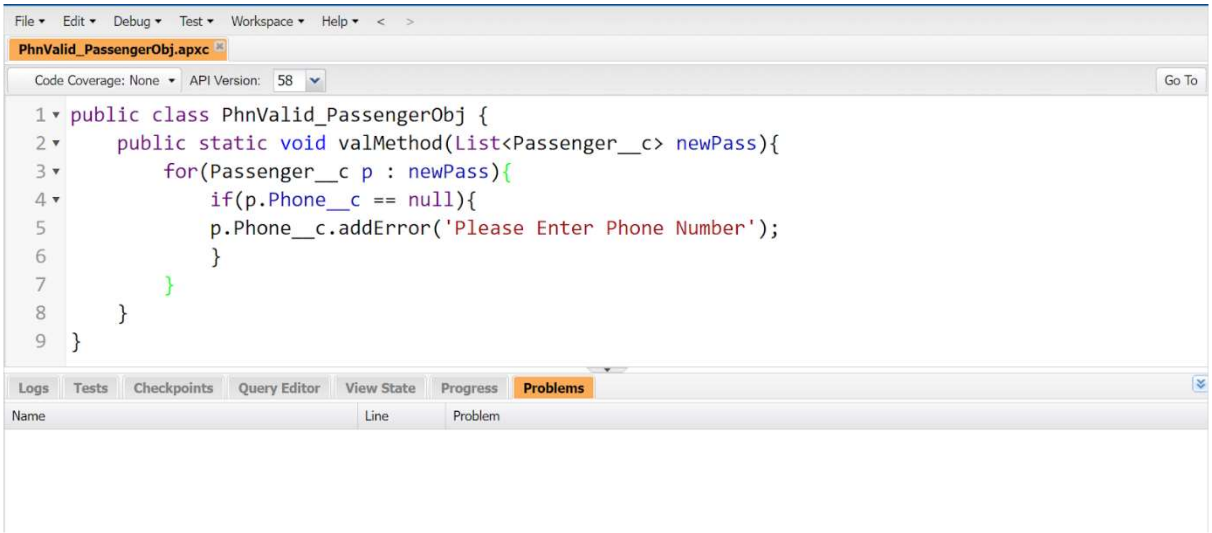


4. Give the Apex Class name as “PhnValid_PassengerObj”.



5. Click ok.

6. Now write the code logic here



Code Coverage: None API Version: 58

```
public class PhnValid_PassengerObj {
    public static void valMethod(List<Passenger__c> newPass){
        for(Passenger__c p : newPass){
            if(p.Phone__c == null){
            p.Phone__c.addError('Please Enter Phone Number');
            }
        }
    }
}
```

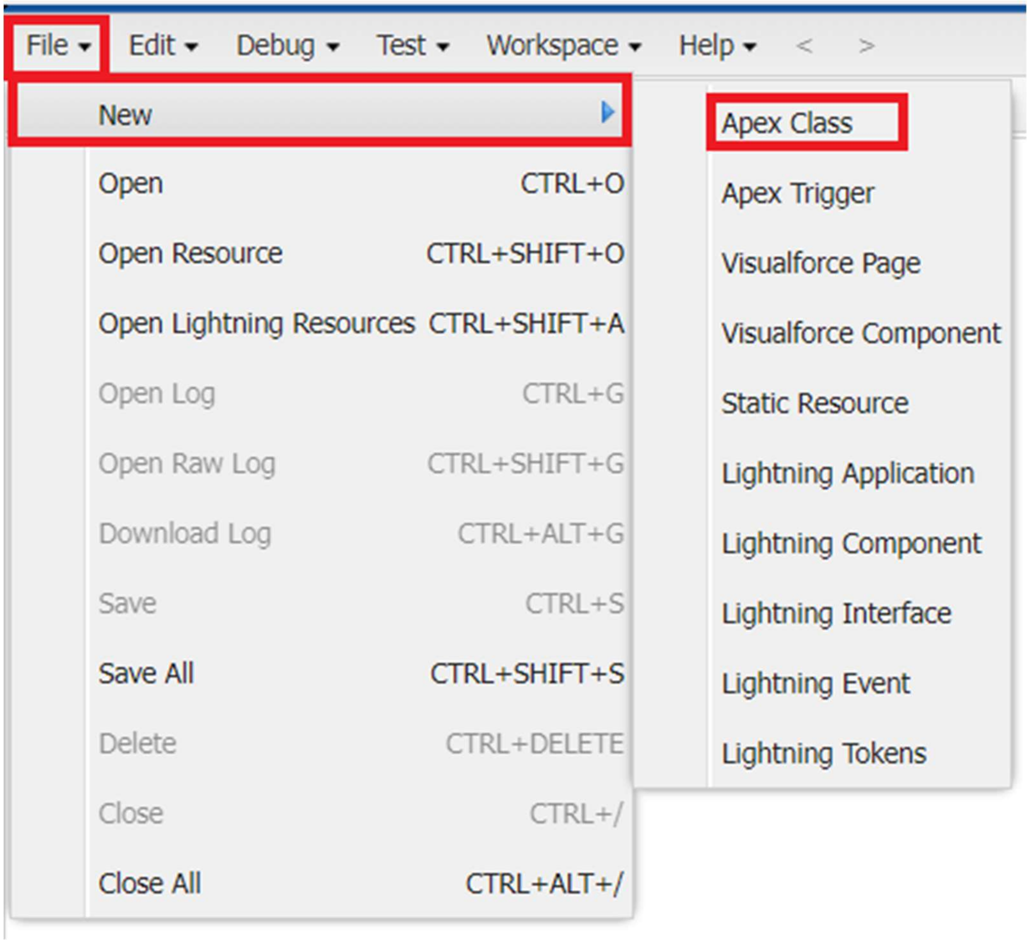
Source Snippet:

```
public class PhnValid_PassengerObj {

public static void valMethod(List<Passenger__c> newPass){

for(Passenger__c p:newPass){

if(p.Phone__c==null){

p.Phone__c.addError('please Enter phone Number');

}

}

}

}
```

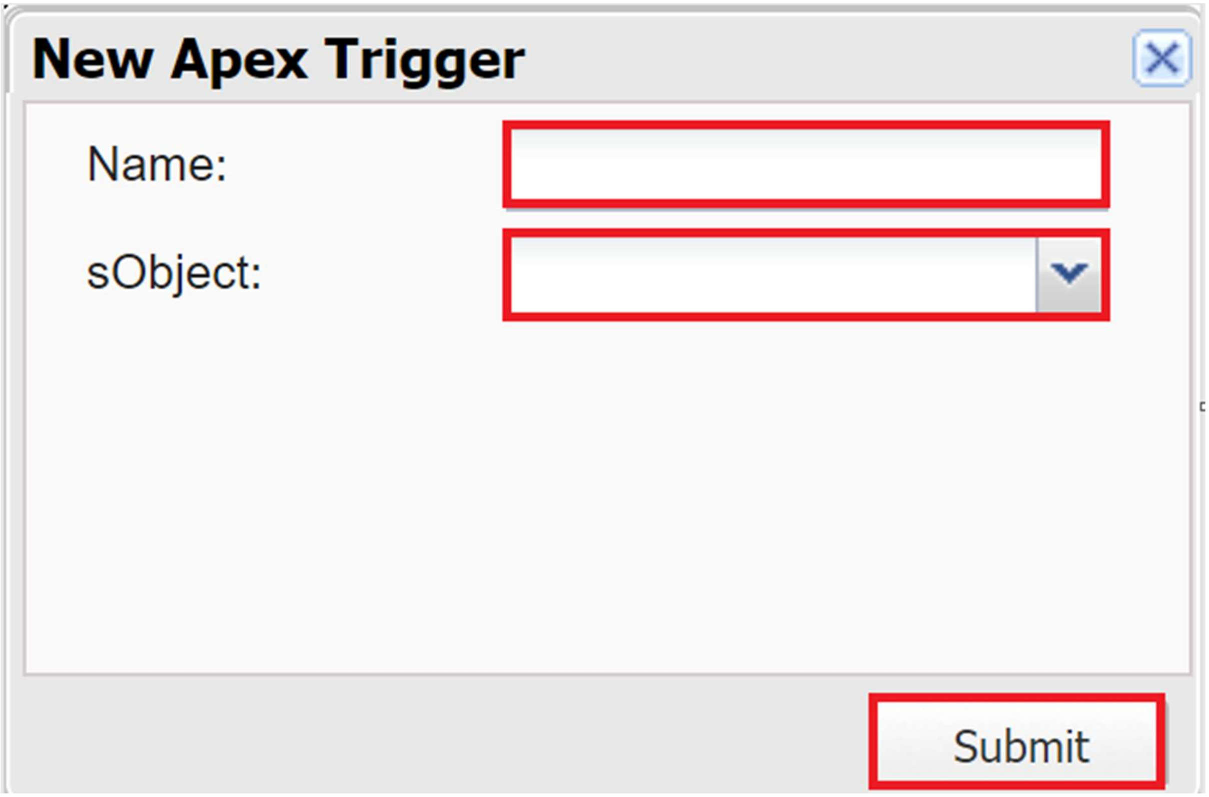
7. Save the code.(click on file --> Save)

Activity 2 : Create an Apex Trigger

1. To create a new Apex Class follow the below steps:
   Click on the file --> New --> Apex Class.

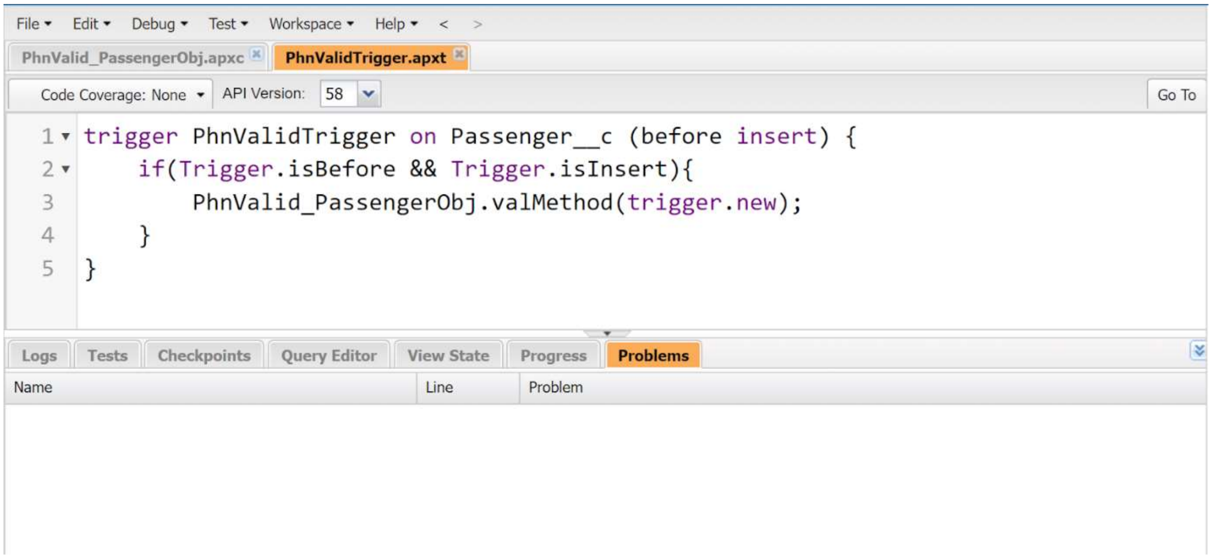


2. Give the Apex Trigger name as “PhnValidTrigger”, and select “Passenger__c” from the dropdown for sObject.



3. Click Submit.

4. Now write the code logic here



```
trigger PhnValidTrigger on Passenger__c (before insert) {
    if(Trigger.isBefore && Trigger.isInsert){
        PhnValid_PassengerObj.valMethod(trigger.new);
    }
}
```

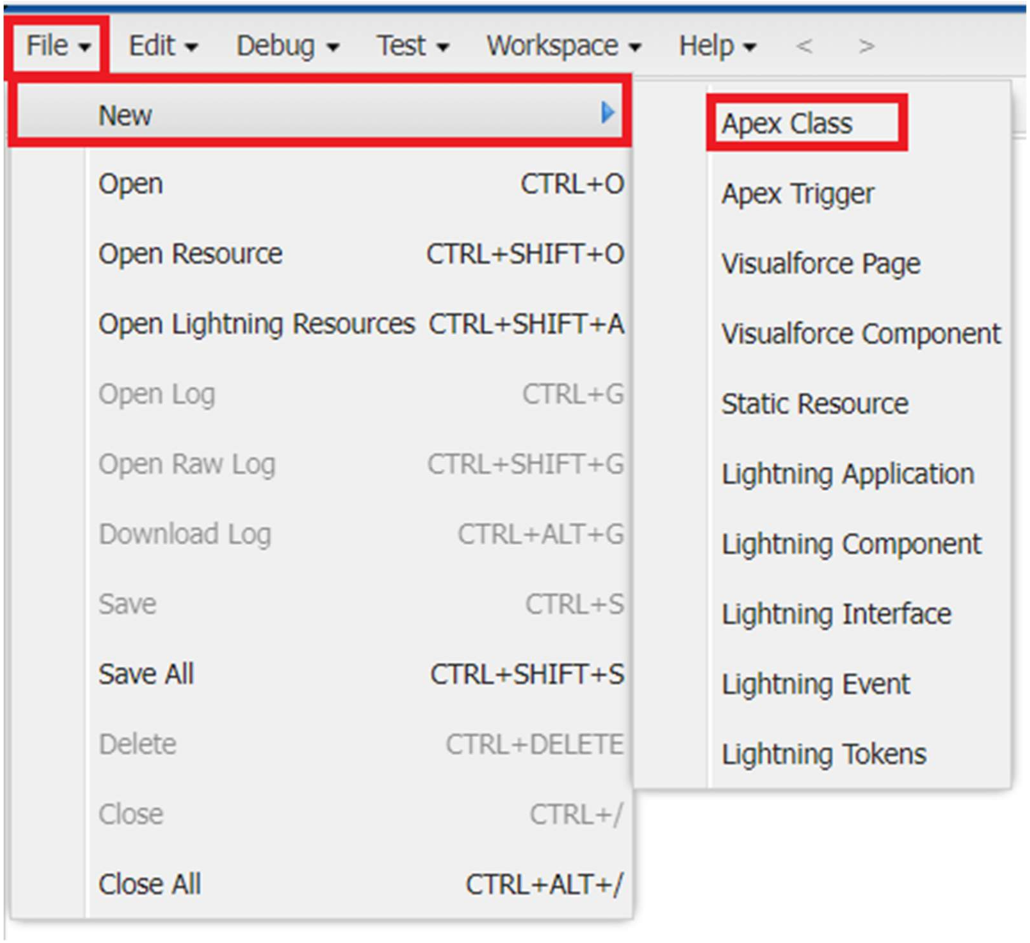
Source Code:

```
trigger PhnValidTrigger on passenger__c (before insert) {

if(trigger.isBefore && trigger.isInsert){

PhnValid_PassengerObj.valMethod(trigger.new);

}

}
```

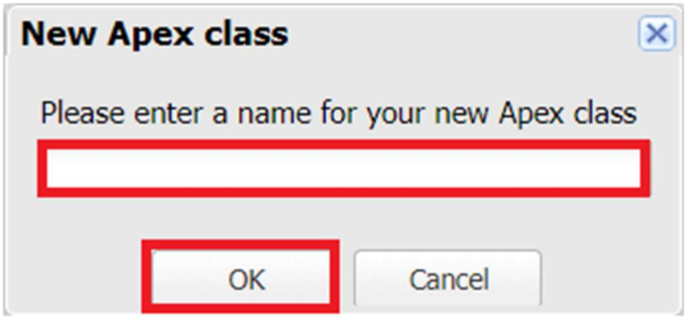
5. Save the code.(click on file --> Save)

Activity 3 : Create an Apex Test Class

1. To create a new Apex Class follow the below steps:
   Click on the file --> New --> Apex Class.



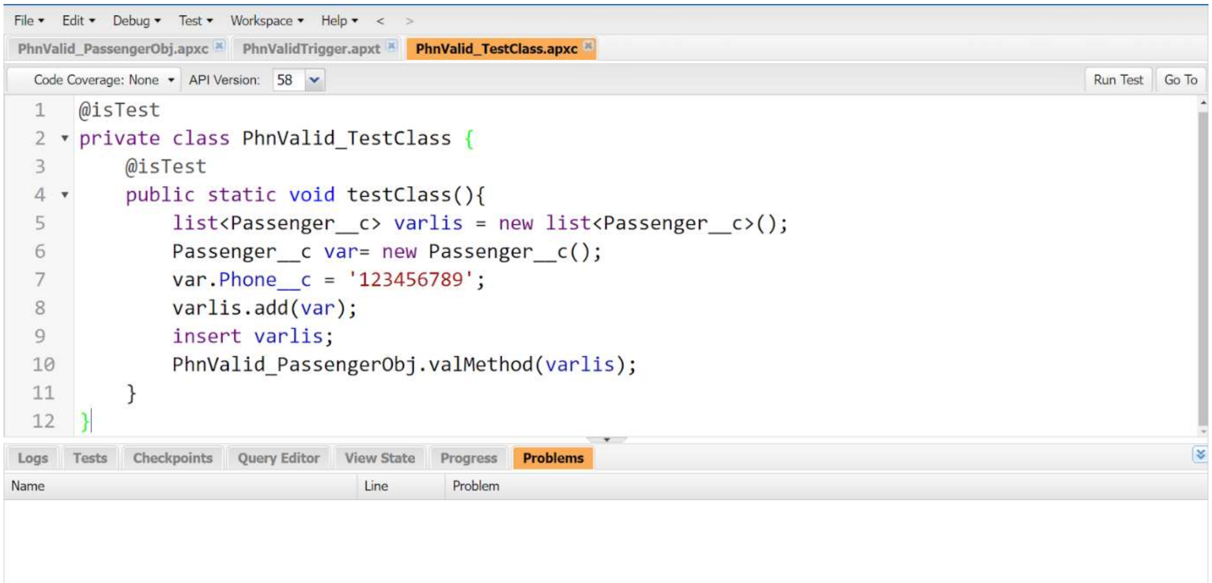
2. Give the Apex Class name as “PhnValid_TestClass”.



Please enter a name for your new Apex class

3. Click ok.

4. Now write the code logic here



Code Coverage: None API Version: 58

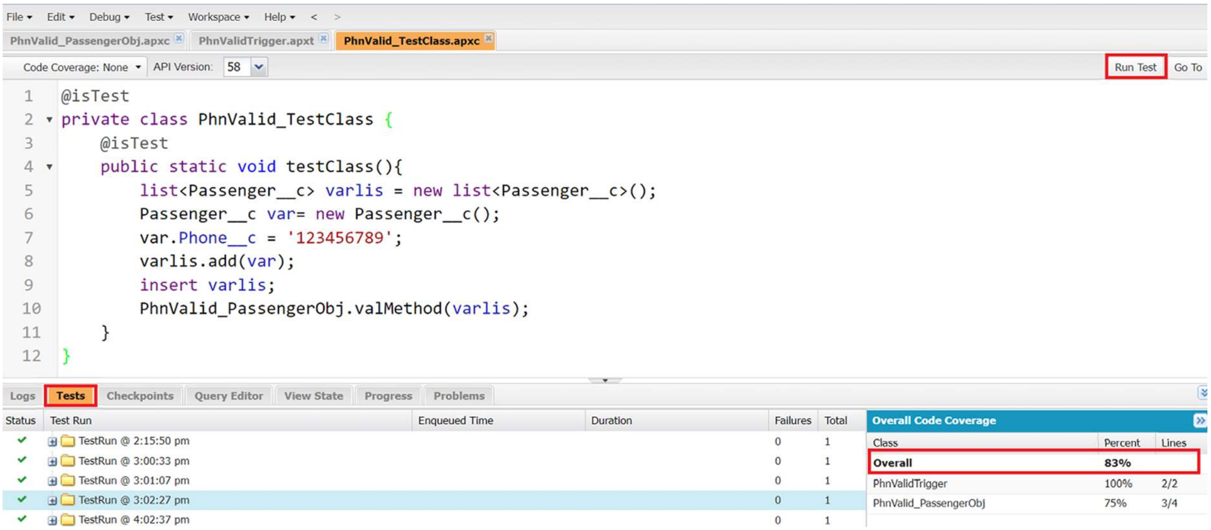
```
@isTest
private class PhnValid_TestClass {
    @isTest
    public static void testClass(){
        list<Passenger__c> varlis = new list<Passenger__c>();
        Passenger__c var= new Passenger__c();
        var.Phone__c = '123456789';
        varlis.add(var);
        insert varlis;
        PhnValid_PassengerObj.valMethod(varlis);
    }
}
```

Source Code: @isTest

```
public class PhnValid_TestClass {

  @isTest

  public static void testClass(){
    list <Passenger__c> varlis = new list<Passenger__c>();
    Passenger__c var = new Passenger__c();
    var.Phone__c = null;
    varlis.add(var);
    insert varlis;
    PhnValid_PassengerObj.valMethod(varlis);
  }

}
```

5. Save the code.(click on file --> Save).

6. Click on “Run Test” and then click on Test under the terminal section and do check that your overall code coverage should be more than 75%.



Code Coverage: None ▾ API Version: 58

```
@isTest
private class PhnValid_TestClass {
    @isTest
    public static void testClass(){
        list<Passenger__c> varlis = new list<Passenger__c>();
        Passenger__c var= new Passenger__c();
        var.Phone__c = '123456789';
        varlis.add(var);
        insert varlis;
        PhnValid_PassengerObj.valMethod(varlis);
    }
}
```

Status	Test Run	Enqueued Time	Duration	Failures	Total
✓	TestRun @ 2:15:50 pm			0	1
✓	TestRun @ 3:00:33 pm			0	1
✓	TestRun @ 3:01:07 pm			0	1
✓	TestRun @ 3:02:27 pm			0	1
✓	TestRun @ 4:02:37 pm			0	1

Class	Percent	Lines
Overall	**83%**	
PhnValidTrigger	100%	2/2
PhnValid_PassengerObj	75%	3/4