**Test Classes:**

* Test Classes are used to check the functionality of the Apex Classes.
* Test classes must cover 75% of tests.

1. Create a Apex Class:

* Define a method to insert record
* Define another method to update Records:

**Example 1: ApexClass:**

public class AccountTestingClass {

public static List<Account> createAccount()

{

List<Account> acc=new List<Account>();

for(Integer i=0;i<100;i++)

{

Account a=new Account();

a.name='Capgemini '+i;

if(Math.Mod(i,2)==0)

{

a.Industry='Apparel';

}

else if(Math.Mod(i,3)==0)

{

a.Industry='Agriculture';

}

else

{

a.Industry='Energy';

}

a.Phone='0123456789';

acc.add(a);

}

insert acc;

return acc;

}

public static List<Account> updateAccount()

{

List<Account> list1 = [select name,phone,industry from Account where name like '%Capgemini%'];

for(Account acc:list1)

{

if(acc.Industry=='Energy')

{

acc.Phone='1234567890';

}

else if(acc.Industry=='Apparel')

{

acc.Phone='1234567891';

}

else

{

acc.Industry='1234567892';

}

}

update list1;

return list1;

}

}

1. **Apex TestClass:**

@isTest

public class AccountTestingTestClass {

@testSetup

public static void testCreate()

{

List<Account> acc=new List<Account>();

for(Integer i=0;i<100;i++)

{

Account a=new Account();

a.name='Capgemini '+i;

if(Math.Mod(i,2)==0)

{

a.Industry='Apparel';

}

else if(Math.Mod(i,3)==0)

{

a.Industry='Agriculture';

}

else

{

a.Industry='Energy';

}

a.Phone='0123456789';

acc.add(a);

}

insert acc;

}

@isTest

public static void testUpdate()

{

//AccountTestingClass.updateAccount();

Test.startTest();

List<Account> createList=AccountTestingClass.createAccount();

List<Account> updateList=AccountTestingClass.updateAccount();

Test.stopTest();

System.assertNotEquals(null,createList[0].id,'Account Id expected not to be null');

System.assertEquals('1234567891', updateList[0].Phone, 'Account Phone Number Updated');

}}

Key Points:

* Test class must have 75% coverage.
* Test Triggers class needs 1% coverage.
* @isTest is mentioned in front of class name and also in front of method signature. , which doesn’t count the org code limit.
* Test.startTest() and Test.stopTest() : To set separate governor limit to the code which is inside the start and stop.

Indicates some action/trigger logic

* System.assert: Used to compare the actual and expected output.

------System.assertNotEquals(expected,actual,msg);

------System.assertEquals(expected,actual,msg);

* Test Method :

------It must be defined . // means must contain some logic.

------It does not use Original database data.

**Example 2:** Custom Object (Student)

Apex Class:

public class StudentClass {

public static List<Student\_\_c> addStudent ()

{

List<Student\_\_c> stud=new List<Student\_\_c>();

Student\_\_c s=new Student\_\_c();

s.Name='Student 2';

s.Class\_\_c='BSC';

s.Marks\_\_c=50;

stud.add(s);

insert stud;

return stud;

}

public static List<Student\_\_c> updateStudent()

{

List<Student\_\_c> stud= [select name,Class\_\_c,Marks\_\_c from Student\_\_c];

for(Student\_\_c s:stud)

{

s.Marks\_\_c=80;

}

update stud;

return stud;

}

}

Apex Test Class:

@isTest

public class StudentTestClass {

@testSetup

public static void testaddStudents()

{

List<Student\_\_c> stud=new List<Student\_\_c>();

Student\_\_c s=new Student\_\_c();

s.Name='Student 2';

s.Class\_\_c='BSC';

s.Marks\_\_c=50;

stud.add(s);

insert stud;

}

@isTest

public static void testupdateStudents()

{

Test.startTest();

List<Student\_\_c> list1=StudentClass.addStudent();

List<Student\_\_c> list2=StudentClass.updateStudent();

System.assertNotEquals(null,list1[0].id,'Student Should have ID');

System.assertEquals(80,list2[0].Marks\_\_c,'MArks updated');

Test.stopTest();

}

}

**Example 3: Opportunity Example**:

Apex Class:

public class OppClass {

public static List<Opportunity> createOpp()

{

List <Opportunity> op1=new List<Opportunity>();

for(integer i=0;i<5;i++)

{

Opportunity op=new Opportunity();

op.Name='Opportunity '+i;

op.Amount=10000;

op.Description='Amount 100';

op.StageName='Closed Won';

op.Discount\_Percent\_\_c=10;

op.CloseDate=System.today();

op1.add(op);

}

insert op1;

return op1;

}

public static List<Opportunity> updateOpp()

{

List<Opportunity> op2=[select name,amount,description from Opportunity where Name Like '%Opportunity%'];

for(Opportunity o:op2)

{

if(o.Amount > 1000)

{

o.Description='Greater Than 1000';

}

else

{

o.Description='Less Than 1000';

}

}

update op2;

return op2;

}}

Apex Test Class:

@isTest

public class OppTestClass {

@testSetup

public static void testCreateOpp()

{

List<Opportunity> op1=new List<Opportunity>();

for(integer i=0;i<5;i++)

{

Opportunity op=new Opportunity();

op.Name='Opportunity '+i;

op.Amount=100;

op.Description='Amount 100';

op.StageName='Closed Won';

op.Discount\_Percent\_\_c=10;

op.CloseDate=System.today();

op1.add(op);

}

insert op1;

}

@isTest

public static void updateOppTest()

{

Test.startTest();

List<Opportunity> createList=OppClass.createOpp();

List<Opportunity> updateList=OppClass.updateOpp();

Test.stopTest();

System.assertNotEquals(null, createList[0].id, 'It should contain Id');

//System.assertEquals('Less Than 1000',updateList[0].Description,'Amount Less than 1000');

System.assertEquals('Greater Than 1000',updateList[0].Description,'Amount greater than 1000'); }}

**Example 4:**

* No insert Method in main class , created the test records in the test class using @testSetup.

Apex Class:

public class ApexClass {

public static void myMethod1(Account acc)

{

System.debug('inside method1');

if(acc.name == 'Test Account 1')

{

acc.Website='www.google.com';

update acc;

}

}

public static void myMethod2(List<Account> accounts){

System.debug('inside method 2');

}

}

Apex TestClass:

@isTest

public class ApexTestClass {

@testSetup static void createRecords()

{

Account acnA=new Account();

acnA.name='Test Account 1';

Account acnB=new Account();

acnB.name='Test Account 2';

acnB.website='www.google.com';

insert acnA;

insert acnB;

}

@isTest static void test\_Method1()

{

Account acc=[select id,name from Account where name='Test Account 1'];

ApexClass.myMethod1(acc);

}

@isTest static void test\_Method2()

{

List<Account> acc=[select id,name,website from Account];

ApexClass.myMethod2(acc);

}}