) What is test class in Salesforce?

.When we are moving code from sandbox to production org we need to write a test class.

.Whatever code we have written we need to ensure that each and every condition is correct

and we can create dummy data for it.

.The code must have 75% coverage to deploy it to production.

Points to remember.

1)Test class does not have access to organization data but if we write @isTest(seeAllData=true),

then we can access data in the test class as well.

2)As every organization have code limit by writing @isTest above test class, the limit will not be considered.

3)Every test class has a method declared with testmethod keyword and is static.

Syntax:

static testmethod void methodname()
{

}

2) What is test.starttest() and test.stoptest()?

.Whatever test we are performing we write the test inside of Test.starttest() and Test.stoptest.

.Basically they help to create a fresh set of governing limit.

.Each test method can call the start test and stop test only once.

3) What is the use of seeAllData=true ?

==>seeAllData=true

If we are defining a test class with @isTest(SeeAllData=true) then we can access data in test class from the database under all method present in the test class. Under this case annotating a method with @isTest(SeeAllData=false) would be ignored and you can access database data inside this method as well.

Let's understand with an example.

@isTest(SeeAllData=true)

public class testclassexample {

static testmethod void testMethodFirst() {

contact obj = [SELECT Id, Name from contact where name='test' LIMIT 1];

//Accessing contact from database

}

@isTest(SeeAllData=false)

static testmethod void testMethodFirst() {

contact obj1 = [SELECT Id, Name from contact where name='test' LIMIT 1];

//Accessing contact from database although @isTest(SeeAllData=false) is applied for method but it will get ignored as test class is having @isTest(SeeAllData=true).

}

}

4) What is the use of seeAllData=false ?

==>@isTest(SeeAllData=false)

If we are defining a test class with @isTest(SeeAllData=false) then we cannot access data in test class from the database under all method present in the test class. Under this case annotating a method with @isTest(SeeAllData=true) would not be ignored and you can access database data inside this method.

Let's understand with an example.

```
@isTest(SeeAllData=false)

public class testclassexample {

static testmethod void testMethodFirst() {

//You cannot access like this now.

contact obj = [SELECT Id, Name from contact where name='test' LIMIT 1];

//You need to create test data

Contact obj=new contact();

obj.name='test';

insert obj;

}

@isTest(SeeAllData=true)

static testmethod void testMethodFirst() {

contact obj1 = [SELECT Id, Name from contact where name='test' LIMIT 1];

//Accessing contact from database although @isTest(SeeAllData=false) is applied for
class but it will get ignored.

}

}
```

5) What are test class best practices?

1) We should always considered doing testing with bulk records.

2) We Should test "test cases" for both positive and negative test scenario.

3) Avoid using (SeeAllData=true) in test class because it might happen that the test class pass in sandbox but fails in production if data is not present in production.

4) Avoid using hardcoded is's in test class.

6) What is @testSetup?

Method that are annotated with @testSetup are used to create test data and we can this test data inside every method of the test class. The testSetup method is executed first in any test class before other methods.

7) Let say i have a test class in which i have created a testSetup method in which i am creating a contact record. Let say i have two test method inside my test class which is accessing this contact record. If i access this contact record in testmethod1() and made a update to the contact record from testSetup method and now i am accessing this contact in testmethod2(), so will i get the updated contact record or the original unmodified contact record in testmethod2()?

testMethod2() will gets access to the original unmodified state of record, this is because changes are rolled back after each test method finishes execution.

8) If the test class is having access to organization data using @isTest(SeeAllData=true) annotation, will the test class be able to support @testSetup method?

@testSetup method are not supported in this case.

9) How many @testSetup method are supported inside a test class?

One test setup method per test class.

10) Why is @TestVisible annotation used in test class?

Using @TestVisible annotation before variable or methods or inner classes inside apex class allows test class to access them. This annotation does not change the visibility of variable or methods or inner classes if accessed by non-test class.

Apex class:

public class apexClassExample{

@TestVisible private static integer rollNo=1;

}

Test class:

@isTest

```
public class apexClassExampleTest{

static testmethod void methodName()

{

    Integer rollNo=apexClassExample.rollNo;

}


}
```

11) What is System.runAs() in test class?

Test class runs in system mode, System.runAs() allow us to run the test in context of current user where user 's record sharing is taken into account. User permission or field level permission are not taken into account.

12) What is the use of test.isrunningtest()?

Sometimes it may happen that in test class you are not able to satisfy certain condition in apex class

so under this case, we can bypass this condition by writing test.isrunningtest() in apex class.

Syntax:

In Apex under condition,

if(a=3 || test.isrunningtest())

{

}