# Week 6 Test

## Luis Eduardo Enriquez Perez

1. **What is unit testing? how do we unit test in android? What library?**

These are small tests that you can run in isolation from production systems. They typically mock every major component and should run very quickly. The library use is JUnit.

1. **What is instrumentation testing? How do we make instrumentation test in Android?**

Instrumentation tests run on a device or an emulator. In the background, your app will be installed and then a testing app will also be installed which will control your app, lunching it and running UI tests as needed.

Instrumentation tests can be used to test none UI logic as well. They are especially useful when you need to test code that has a dependency on a context.

1. **Which library in Android do we use to mock objects/methods in android for testing?**

Mockito.

1. **Where do we create unit test classes?**

In the Test package.

1. **What are the standard annotations we use to create a simple unit test?**

Before, Test and After.

1. **How to mock an object? Write the code to do it in two ways?**

Addition addition;  
Subtraction subtraction;  
  
/\* Before comes from JUnit. \*/  
@Before  
public void setUp()  
{  
 addition = *spy*(Addition.class);  
 multiplication = *mock*(Multiplication.class);  
}

1. **How to mock a method in the mock object?**

After we just mocked the classes that we are going to use in the setup method of the testing class (with the @Before annotation), after, we create a method with the @Test annotation, here we can call the method inside the mocked class.

1. **What is the expected value and actual value in an assert statement?**

The expected value is a value we give to the method, expecting that it will return that value, and the actual value, is the value given after the execution of the method, we use these to verify that the method is working accordingly.

1. **Where do we create the espresso test classes?**

In the androidTest packages.

1. **Do we need the device/emulator to run espresso tests?**

Yes.

1. **Why do we need to create a rule for an espresso test?**

Because Rules allows uses to add or redefine the behaviour of each test method in a test class in a very flexible way.

1. **How to create a custom Content Provider?**

* You need to create a class for the custom content provider, this class will extend the “ContentProvider”, and there we need to implement these methods:
  + onCreate()
  + getType(Uri)
  + Delete(Uri uri, String selection, String[] selectionArgs)
  + Insert(Uri uri, ContentValues values)
  + Query(Uri uri, String[] projection, String selection, String[] selectionArgs, String sortOrder)
  + Update(Uri uri, ContentValues values, String selection, String[] selectionArgs)
* After that, we need to create a contract class, here in this class we will define the methods so the clients can access to our data.
* Then we need to create the UriMatcher definitions. To deal with multiple URIs Android provides the helper class UriMatcher. This class eases the parsing of URIs.
* We call our custom content provider in the onCreate() method.
* We Implement the getType() method, so our content provider returns the right content type for it’s supported URIs.
* We use the Insert() method to add new records.
* We notify the listeners of dataset changes.
* We query the content of the content provider with the SQLiteHelper.
* And finally, we need to declare the contentprovider in the manifest of the application, there we create the read and write permissions, that will be required for each applications that wants to access to our content provider.

1. **Name some operators used in RxJava?**

Just, filter, take, concat, merge, zip.

1. **How do you handle synchronization in Java?**

To make method synchronized, we only need to add the “synchronized” keyword to its declaration:

Public synchronized void increment()

{

C++;

}