Unit Testing

Programming the Android Platform

Android Unit Testing

- Android-specific extensions to Junit
- MoreAsserts
  - Additional result checking classes
- ViewAsserts
  - Asserts about view layout
- TouchUtils
  - Classes for generating touch events
- Instrumentation
  - For monitoring application interaction with system

Android Unit Testing (cont.)

- Steps for testing Activity
  - Create an Android test project
  - Create one or more Android test classes
    - Add code to each Test class
- Test code usually includes test methods for
  - App initialization
  - UI behavior & class functionality
  - Application lifecycle state management

Android Unit Testing (cont.)

- AndroidTestCase
  - Test case with access to Activity Context
  - ActivityUnitTestCase
    - isolated testing of a single activity
- InstrumentationTestCase
  - Test case with access to system instrumentation & events
  - ActivityInstrumentationTestCase2
    - Functional testing of a single activity
- Many other test case types
  - developer.android.com/guide/topics/testing/testing_android.html

Android Unit Testing (cont.)

- Junit Test flow
- setup()
  - Put application in known state
- run()
  - Create additional data
  - Execute method under test
  - Check result with Assert
- tearDown()
  - Return to known state

HelloAndroidTest

```
import android.test.ActivityInstrumentationTestCase2;
import course.examples.HelloWorld.HelloWorld.HelloAndroid;
...
public class HelloAndroidTest extends
        ActivityInstrumentationTestCase2<HelloAndroid> {

 private HelloAndroid mActivity;
 private TextView mView;
 private String resourceString;

 public HelloAndroidTest() {
  super("course.examples.HelloWorld.HelloWorld”,
            HelloAndroid.class);
}
```

HelloAndroidTest (cont.)

```
protected void setUp() throws Exception {
  super.setUp();
  mActivity = this.getActivity();
  mView = (TextView) mActivity.findViewById
          (course.examples.HelloWorld.HelloWorld.R.id.textview);
  resourceString = mActivity.getString
          (course.examples.HelloWorld.HelloWorld.R.string.hello);
}
```

HelloAndroidTest (cont.)

```
public void testPreconditions() {
  assertNotNull(mView);
}

public void testText() {
   assertEquals(resourceString,(String)mView.getText());
}
```

Android Unit Testing (cont.)

- developer.android.com/resources/tutorials/testing/activity_test.html

Layout inspector