

Engineering Design for Embedded Systems: Assignment 6 (version 2)*

Due Date: February 25, 2013

This is an assignment on testing. The first half gives you practice with JUnit tests, while the second half is Android-specific testing.

If I find any cases of plagiarism on this assignment, I will apply the standard penalties and report the case to the Associate Dean, as per Policy 71.

1 JUnit Tests for List (5 points, 1 for each test)

I've included yet another `List` implementation as well as a skeleton JUnit test class, `A6Q1`. We saw in Lecture 16 how to create JUnit tests.

Task. Add the following tests to `A6Q1`.

- test the `clear()` method by creating a list, adding some elements to the list, clearing it, and then calling `isEmpty` and checking its return value.
- test the `append`, `head` and `tail` methods by creating a list, adding an element, and checking that it is both the head and tail of the list.
- test the `append` method by creating a list, appending two elements to the list, then checking that `removeFirstItem` removes them in the proper order.
- test the `prepend` method by creating a list, prepending two elements to the list, then checking that `removeFirstItem` removes them in the proper order.
- test `removeFirstItem` on an empty list. Note that the provided `removeFirstItem` implementation throws an exception on an empty list.

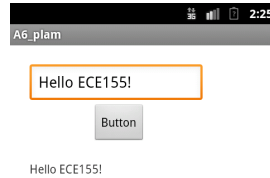
Handin. For this part, hand in your version of `A6Q1.java` with the new tests.

2 Testing Android Activities (5 points)

We previously talked about `onSaveInstanceState()` but never practiced it. We'll therefore practice that along with Android Activity integration testing in this part of the assignment.

*Changes in version 2: `append` → `prepend`; package should be `ca.uwaterloo.ece155.a6`.

Creating a Test Subject (2/5 points). Create an Android Activity which contains an `EditText`, a `Button`, and a `TextView`. Please put the Activity in package `ca.uwaterloo.ece155.a6` (this helps us mark the assignments). When you click on the button, the Activity copies the value from the `EditText` to a field on the Activity. The button click listener must not directly set the `TextView` contents. I assume that you know how to do everything up to here.



(Here's a hint: you can rotate the emulator by hitting Ctrl-F12.)

(2 points) Using a Java `Timer` (not a `Handler/Runnable!`), the `TextView` must poll the field every 100ms and copy the field value to its text. You must also clean up the `Timer` when appropriate.

(0 direct points) Your Activity must preserve the state—the field value—upon orientation change.

Creating Unit Tests (3/5 points). Create an Android Test project and a test class in it. The package should now be `ca.uwaterloo.ece155.a6.test`. Add a `setUp()` method and a set of Android tests (1 point each) which verify that:

- entering text in the `EditText` does not change the text in the `TextView`;
- entering text in the `EditText` and clicking on the button makes the text appear in the `TextView`;
- changing the orientation of the phone preserves the text in the `TextView`.

Tip for `assertEquals()`. When you call `getText()` to retrieve the contents of a `TextView` or an `EditText`, you'd better call `toString()` on the result before passing it to `assertEquals()`.

Tip for Second Test. There's a problem with the second test. The problem is that you have to wait before the change gets propagated from the `EditText` to the `TextView`, because it's running in another thread on a timer. For this assignment only, it's OK to write this:

```
try {
    Thread.sleep(1000);
} catch (InterruptedException e) {}
```

Usually that would be bad practice for tests, as it slows down the test execution. But I can't think of an alternative here.

Tip for Third Test. To cause an orientation change, call the following method on the Activity: `setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_LANDSCAPE);` (or `_PORTRAIT`, as appropriate).

Handin. For this part, hand in your `MainActivity.java` file, your `MainActivityTest.java` file, your `R.java` file for the app, and the `apk` files for your test and app.