

TeenCoder™: Android Programming

First Edition Errata Sheet

Updated May, 2013

This document lists the known typographical or other corrections to the *TeenCoder™: Android Programming* First Edition course.

- Chapter Nine, Lesson Three, page 158:

The discussion explaining how to process click events on the **Gallery** control left out a final step. After implementing the **OnItemClickListener** interface on an object, you need to call the **setOnItemClickListener()** method on the **Gallery** in order for the **Gallery** control to pass click events to that listening interface.

```
Gallery gallery = (Gallery)findViewById(R.id.myGallery);
gallery.setAdapter(ia);
gallery.setOnItemClickListener(this);
```

In this example we are passing in “**this**” as the interface implementation, assuming that the **OnItemClickListener** interface has been implemented directly by the parent **Activity**.

- Chapter Twelve, Lesson Two, pages 212-213:

This task is listed as step 2 under “**Receiving SMS Messages in an Application**”, but no example was given in the textbook:

“2. Register your **BroadcastReceiver** object with the Android system by calling **registerReceiver()** from your **Activity**. Use a filter that will only call our function when a SMS message is received.”

To register your receiver, you will use an **IntentFilter** with the action “android.provider.Telephony.SMS_RECEIVED” like this:

```
registerReceiver(mySmsReceiver,
                new IntentFilter("android.provider.Telephony.SMS_RECEIVED"));
```

- Chapter Twelve, Weather Application Activity

The free World Weather API has changed slightly since the first edition course was printed. Please see updated registration instructions listed on our Errata page to obtain your Weather API key.

In addition, you will need to make a small adjustment to the `getWeatherURL()` function provided in the “Main.java” activity starter. Comment out the old `strWURL` and add the new `strWURL` listed below in bold.

```
private static String getWeatherURL(String location)
{
    // Create and initialize the String value
    String strWURL = null;

    // Build the URL string using the location and APIKey values
    //strWURL = "http://free.worldweatheronline.com/feed/weather.ashx?q=" +
location + "&format=csv&num_of_days=2&key=" + APIKey;
    strWURL = "http://api.worldweatheronline.com/free/v1/weather.ashx?q=" +
location + "&format=csv&num_of_days=2&key=" + APIKey;
}
```

- Chapter Fourteen, Lesson Three, page 262-263, early printings:

The discussion of **Map Overlays** does not mention that the `onTouchEvent()` callback will actually be called for both “down” and “up” events as the user touches and releases a point on the screen. In order to avoid double processing of a single click, your `onTouchEvent()` method should check the action of the event and only process either the up or down event.

The bolded “if” statement below will eliminate skip function calls except for the “down” event:

```
// override the onTouchEvent method
public boolean onTouchEvent(MotionEvent event, MapView mv)
{
    // ignore everything except "Down" presses
    if (event.getAction() != MotionEvent.ACTION_DOWN)
    {
        return false; // return false to pass event through to underlying MapView
    }
}
```