

Lesson 3 Examples

Create a new project in Eclipse, with the following settings:

Project name: **Lesson3_Examples**

Create new project in workspace should be selected.

Build Target: **Android 2.3.3**

Application Name: **Lesson3_Examples**

Package name: **lastname.firstname. Lesson3_Examples** (substitute lastname.firstname with your last name and first name – no spaces, apostrophes or dashes.)

Create Activity should be checked, and **Lesson3Examples** should be the text for it.

Min SDK Version: **10**

Click **Finish**.

Open the main.xml file, found in the **res/layout** folder.

Modify the <TextView> element's XML to look like this:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:id="@+id/txtOut"
    />
</LinearLayout>
```

This will allow us to output to the screen from our code, as we did in previous lessons. Save and close the main.xml file.

Open Lesson3Examples.java file, in your SRC/package/ folder. We'll begin with string comparison.

Add the code highlighted in yellow to the onCreate event of the file.

```

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);

    final TextView tOut = (TextView) findViewById(R.id.txtOut);
    String sOut = "";

    String sText1 = "test"; // Test, I hate taking tests
    String sText2 = "test";

    if (sText1.contains(sText2)) {
        sOut += "sText1 contains \"" + sText2 + "\"\n";
        sOut += "sText2 starts at char. position: " + sText1.indexOf(sText2) + "\n";
    }

    if (sText1.contentEquals(sText2)) {
        sOut += "Strings are equal!\n";
    }

    if (sText1.equalsIgnoreCase(sText2)) {
        sOut += "Strings are equal regardless of case.\n";
    }

    tOut.setText(sOut);
}

```

Replace the values in sText1 (highlighted in red above) with the following text: “Test”, and run the application. You should see “Strings are equal regardless of case” on the screen. Now, replace “Test” with “I hate taking tests”, and re-run your application. You’ll see “sText1 contains sText2” and “sText2 starts at char. position 14” displayed. Now, change it back to “test”. You should see “sText1 contains test”, “sText2 starts at char. position 0”, and “Strings are equal!” on the screen. Review the video for an explanation.

Next, let’s add a ternary condition on the screen. This one will evaluate a string variable, and set a second variable to be true or false, depending on the text encountered.

Add the highlighted code to your onCreate event:

```

    if (sText1.equalsIgnoreCase(sText2)) {
        sOut += "Strings are equal regardless of case.\n";
    }

    String sLight = "red";
    boolean bDrive = (!sLight.equals("red")) ? true : false;

    if (bDrive) {
        sOut += "bDrive = DRIVE!\n";
    } else {
        sOut += "bDrive is false - STOP!\n";
    }

    tOut.setText(sOut);
}

```

Run your application. You should get “bDrive is false – STOP!” to display on the screen. Try changing the value in sLight from “red” to something else and re-run the code. Note the result.

Now, we’re going to combine a switch statement with a for loop. Add the following code to your onCreate event (as usual, highlighted in yellow):

```
        if (bDrive) {
            sOut += "bDrive = DRIVE!\n";
        } else {
            sOut += "bDrive is false - STOP!\n";
        }

        for (int i = 1; i <= 10; i++) {
            switch (i) {
                case 1:
                    sOut += "i is One!\n";
                    break;
                case 2:
                    sOut += "i is Two!\n";
                    break;
                case 3:
                    sOut += "i is Three!\n";
                    break;
                case 4:
                    sOut += "i is Four!\n";
                    break;
                case 5:
                    sOut += "i is Five!\n";
                    break;
                default:
                    sOut += "i is greater than five!\n";
                    break;
            }
            tOut.setText(sOut);
        }
    }
```

The above code will loop 10 times, outputting a text string for each value. Substitute i++ with i+2 (highlighted in red). Note the results. Remove the break statement from the first case. Note the results. Why did “I is Two” repeat twice?

Let’s finish things up with a method call, and a multi-dimensional array.

Add the following code BELOW your onCreate event. This is a method that outputs a string representation of a two-dimensional array, which is passed in as input to the method. It uses the read-only “for-each” loop to process the array.

```

String GridOutput(int[][] imGrid) {
    String sText = "-----\n";
    for (int[] row : imGrid) {
        for (int cell : row) {
            sText += " | " + cell + " | ";
        }
        sText += "\n";
    }
    sText += "-----\n";
    return sText;
}

```

Now, modify your onCreate event, and add the following code, highlighted in yellow:

```

        default:
            sOut += "i is greater than five!\n";
            break;
    }
}

int[][] inputGrid;

inputGrid = new int[3][3];

for (int i = 0; i < 3; i++) {
    for (int j = 0; j < 3; j++) {
        inputGrid[i][j] = i+j;
    }
}

sOut += GridOutput(inputGrid);

tOut.setText(sOut);
}

```

This code will initialize a new two dimensional array, and pass it to the GridOutput method, which will return a string representation of the values stored in each element of the array.

Save your project, and run it to see the results. Be sure to review the video for this lesson.