**MIND STORM SOFTWARE PVT LTD**  
**Android Development Training**

**Hands-On Exercise – ex05 - ListActivityExercise.docx**

**Objective**This exercise demonstrates how to use the Android ListActivity. The ListActivity makes it easy to show a list of items to the user. The user then typically selects one of the item and you can take appropriate action.

Typical scenarios could be showing a list of products, users and then when the user clicks on it, you can then show another activity that has the details on the row touched/selected.

**Assumptions**

* Development Environment for Android (Java SDK, Eclipse, Android SDK) has been setup successfully.
* You are familiar with using Eclipse.
* Android SDK 4.x is available and Android Virtual Devices are already created.
* Start the Android Virtual Device to save time.
  + Click on **Window 🡪 Android SDK and AVD Manager**.
  + Select an **Android 4.x** compatible AVD and click on **Start**
  + Select **Scale display to real size** and provide a **Screen Size (in)** as **5 inches** or any other appropriate size for your development machine.

**Step by Step Instructions**

**Step 1 – Create the Android Project**

1. Create a new project. Click on **File🡪New 🡪 Android Application Project**
2. Enter **Project Name or Application Name** as **ListActivityExercise .**Click on **Next.**
3. Enter **Package Name** as **com.mindstorm.listactivity .** Click on **Next**
4. Deselect (Uncheck) the **Create custom launcher icon.** Click on **Next.**
5. In **Create Activity**, go with the default options i.e. go with Create Activity and Blank Activity as selected. Click on **Next.**
6. On the **New Blank Activity,** let the Activity Name be **MainActivity** and change the Layout Name to **main**
7. Click on **Finish**
8. (Optional): Verify that the Project runs in your Emulator by **Right-click** the **Project** and **Run As 🡪 Android Application**

**Step 2 - Add an array of values**

First we will add some strings to the resources which will be references in the activity. Go to **res/values** folder and modify **strings.xml** to have the following **additional** string-array element.

<string-array name=*"subjects"*>

<item>Maths</item>

<item>Physics</item>

<item>Chemistry</item>

</string-array>

**Step 3 – Code the Main Activity**

1. Modify the **MainActivity.java** code as shown below:

**package com.mindstorm.listactivity;**

**import android.app.ListActivity;**

**import android.os.Bundle;**

**import android.view.View;**

**import android.widget.ArrayAdapter;**

**import android.widget.ListView;**

**import android.widget.Toast;**

**public class MainActivity extends ListActivity {**

**String[] subjects = null;**

**/\*\* Called when the activity is first created. \*/**

**@Override**

**public void onCreate(Bundle savedInstanceState) {**

**super.onCreate(savedInstanceState);**

**//Read the String Array for subjects that is defined in the strings.xml**

**subjects = getResources().getStringArray(R.array.subjects);**

**//Create the Adapter**

**ArrayAdapter<String> statesAdapter = new ArrayAdapter<String>(this, android.R.layout.simple\_list\_item\_1,subjects);**

**//Assign Adapter to the View**

**getListView().setAdapter(statesAdapter);**

**}**

**@Override**

**protected void onListItemClick(ListView l, View v, int position, long id) {**

**showToast("You selected : " + subjects[position]);**

**}**

**private void showToast(String msg) {**

**Toast.makeText(this, msg, Toast.LENGTH\_SHORT).show();**

**}**

**}**

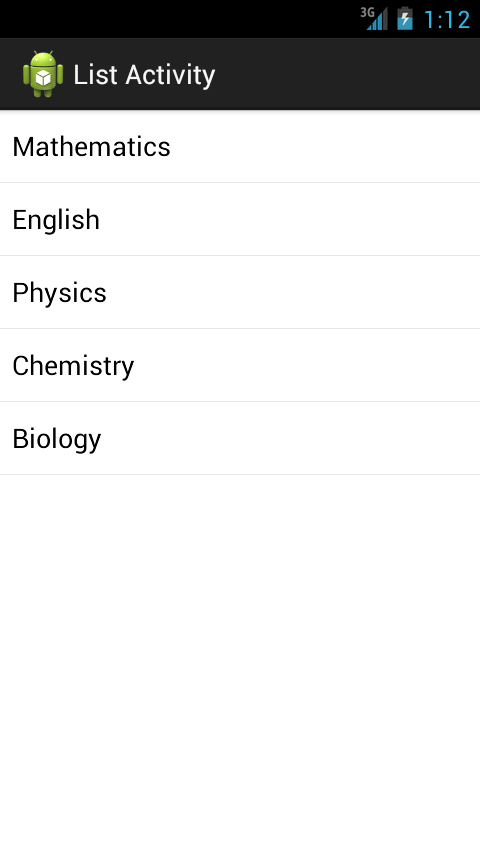
**Observe a few things in the code above:**

1. You extend the **MainActivity** class from **ListActivity** class and not **Activity** class.
2. We do not use any particular layout file (the line is commented) since **ListActivity** has its own layout.
3. We have used an array of strings over here. For simplicity, we have coded the values in the application itself. But it could well have come from the database or any other network service.
4. The Adapter uses the Android **simple\_list\_item\_1** layout.
5. The **ListActivity** exposes a method **onListItemClick** that you can override to determine which item was clicked by the user.

**Step 3 – Run the Example**

1. **Right Click** the **Project** in Eclipse.
2. Select **Run As 🡪 Android** Application
3. (Optional): If you have multiple compatible AVDs running, select the correct AVD. In our case it is the 4.x AVD.

You should see the MainActivity screen come up.



Selecting or clicking on any of the rows (records) will display the item that you clicked.

**Summary**

This hands-on exercise demonstrated how to use the class ListActivity to speed up development of an Activity that presents a list of options to the user to select from. It is one of the most often used activities in most applications.

For more information: <http://developer.android.com/reference/android/app/ListActivity.html>