**Lab Assignment #1 – Developing Android applications that include multiple activities and fragments.**

Due Date: **Sunday of Week 3**, **11:59pm.**

Purpose: The purpose of this lab assignment is to:

1. Use Android Studio IDE
2. Explain and use Android Manifest file
3. Explain and use Intents in Android apps
4. Use resources in Android apps
5. Use fragments in Android apps

References: Textbook, ppt slides, class examples, and Android tutorials ([**http://developer.android.com/training/basics/firstapp/creating-project.html**](http://developer.android.com/training/basics/firstapp/creating-project.html)**, follow the codelab**). This material provides the necessary information that you need to complete the exercises.

Be sure to read the following general instructions carefully:

- This assignment must be completed individually by all the students.

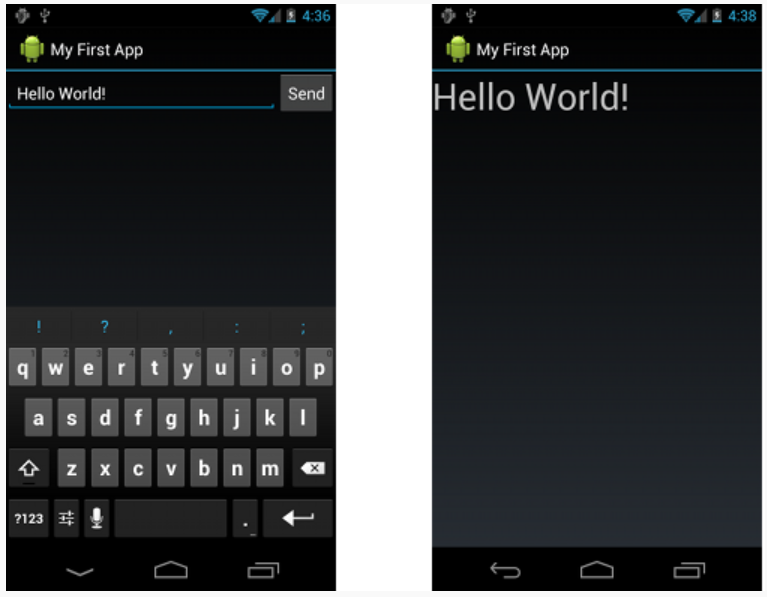
- You will have to **demonstrate your solution in a scheduled lab session** and upload the solution on eCentennial through the **assignment link under Assessments**.

**Exercise 1**

In this exercise, you will build a simple Android application using Android Studio as described here: [**http://developer.android.com/training/basics/firstapp/creating-project.html**](http://developer.android.com/training/basics/firstapp/creating-project.html)**, codelab**.

Make sure you name the project as described on page 2 of this document.

Skip the steps that require the use of command line tools or real devices. This app allows the user to type a message in the text field and click Send button. The message will appear on the second activity as shown below.



**(3 marks)**

**Exercise 2**

Write an Android Application that demonstrates the activity life cycle and fragments lifecycle. The main activity should have two fragments. The top fragment contains a list view control whose items are the names of two other activities, AIActivity and VRActivity. The bottom fragment uses a text view control to display the list of life cycle methods that take place when the main activity starts.

|  |  |
| --- | --- |
|  |  |

Use **Toast** class methods to display a quick message in fragment’s onCreateView and onStart methods.

Alternatively, use Bundle to pass information from the main activity to the bottom fragment:

//use Bundle to pass data to the fragment  
val args = Bundle()  
args.putString("onCreate", "onCreate of MainActivityExecuted")  
//create a fragment reference  
val displayFragment = *supportFragmentManager*.findFragmentById(  
 R.id.*fragment\_container\_view\_bottom*) as BottomFragment  
displayFragment.*arguments* =args

Then, in onCreateView of the bottom fragment, you can use the following code to get the information from Bundle:

// get the data from Bundle  
val info = *arguments*?.getString("onCreate")

The AIActivity and ARActivity should each have a TextView control to display messages when **onCreate**, **onStart**, **onStop**, and **onDestroy** are executed.

Declare the String resources in **strings.xml** file. Feel free to use the code from FragmentsExample app from Week 1 examples.

**(7 marks)**

**Evaluation table:**

|  |  |
| --- | --- |
| **Item** | **Percentage of total mark** |
| **Functionality**:   * **Correct implementation of activities**   + Class code for main activity and two other activities   + UI in XML   + Resources in XML * **Correct implementation of fragments**   + Class code   + UI in XML * **Correct implementation of Event Handlers and life cycle methods**   + onListItemClick event   + life cycle methods for activities and fragments (onCreate, onStart, onCreateView, etc) | 30%  30%  25% |
| **Friendliness**:   * Alignments of UI controls * Friendly I/O | 10% |
| Comments, correct naming of variables, methods, classes, etc. | 5% |
| **Total** | 100% |

**Android Module Naming rules:**

You must name your Android Studio project according to the following rule:

**yourfullname\_COMP304Labnumber\_Exercisenumber**.

Example: **johnsmith\_COMP304Lab1\_Ex1**

**Submission rules:**

Submit your projects as **zip files** that are named according to the following rule:

**yourfullname\_COMP304Labnumber\_Exercisenumber.zip**

Example: **johnsmith\_COMP304Lab1\_Ex1.zip**

**Use Android Studio Export to zip feature to zip your projects.**