**ANDROID**

## Exercise Phone Call

1) Generate a new Android application, give the name as PhoneApp

2) Add the permission of the call into the AndroidManifest.xml file:

*<uses-permission android:name="android.permission.CALL_PHONE"></uses-permission>*

3) Add into the layout of the activity (res/layout/activity_main.xml) EditText a for phone number and a header for it by TextView. Also a Button is needed for a call. Use LinearLayou.

```
<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:text="@string/str_phonenumber"
/>
<EditText
android:id="@+id/phonenumber"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:inputType="number"
/>
<Button
android:id="@+id/buttoncall"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:text="@string/str_call"
/>
</LinearLayout>
```

4) Add into values/strings (res/values/strings.xml) texts:

<string name="str_phonenumber">Phone number:</string>

<string name="str_call">Call</string>

5) Add two variables in the activity class (src/fi/oamk/phoneapp/MainActivity.java)

private Button button;

private EditText number;

ANDROID

6) Add into the activity class in onCreate method a listener for a button click. Add also the call of the call function into the onClick callback method (uses Intent).

```java
public class MainActivity extends Activity {
    private Button button;
    private EditText number;

    @Override

    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        button = (Button) findViewById(R.id.buttoncall);
        number = (EditText)findViewById(R.id.phonenumber);
        // add button listener
        button.setOnClickListener(new Button.OnClickListener() {
            public void onClick(View arg0) {
                String phoneNumber = number.getText().toString();
                String callData = new String("tel:") + phoneNumber;

                call(callData);
            }
        });
    }

    protected void call(String callData) {
        Intent callIntent = new Intent(Intent.ACTION_DIAL); //dial
        callIntent.setData(Uri.parse(callData));
        //Intent in=new Intent(Intent.ACTION_CALL,Uri.parse("0401415086"));
        try{
            startActivity(callIntent);
        }
        catch (android.content.ActivityNotFoundException ex){
            Toast.makeText(getApplicationContext(),"yourActivity is not founded", Toast.LENGTH_SHORT).show();
        }
    }
```
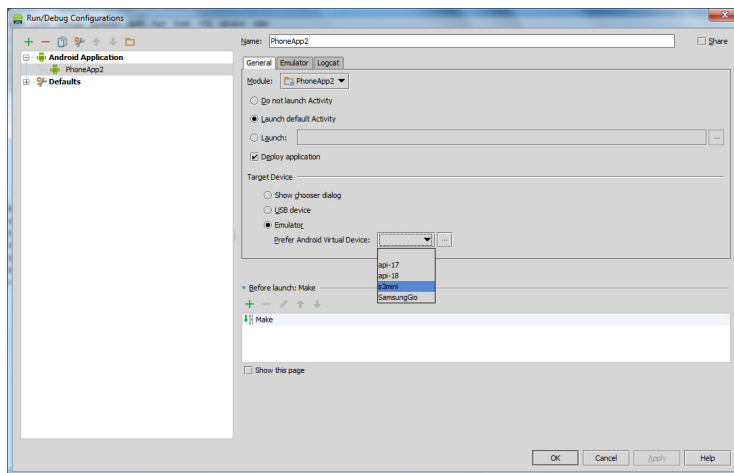
7) Test the application by Android phone as well as emulator.

8) Do similar application but add in it status handling of a call as follows:
   http://www.mkyong.com/android/how-to-make-a-phone-call-in-android/
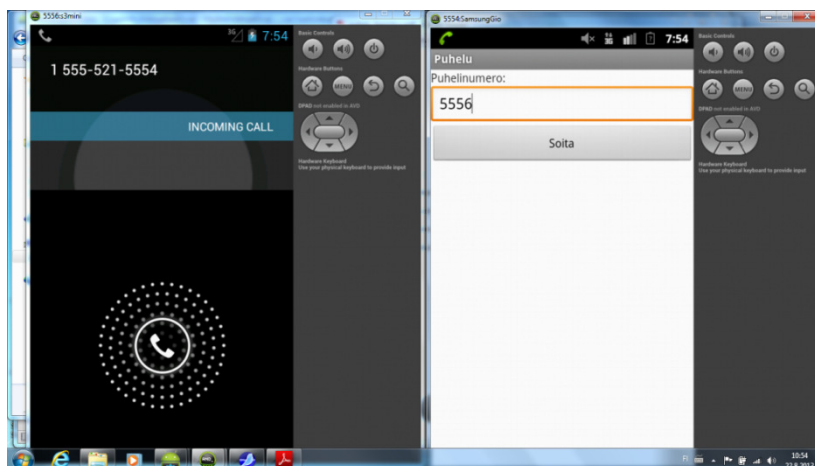
# ANDROID

9) Run emulators using different AVDs (if using only one Studio).



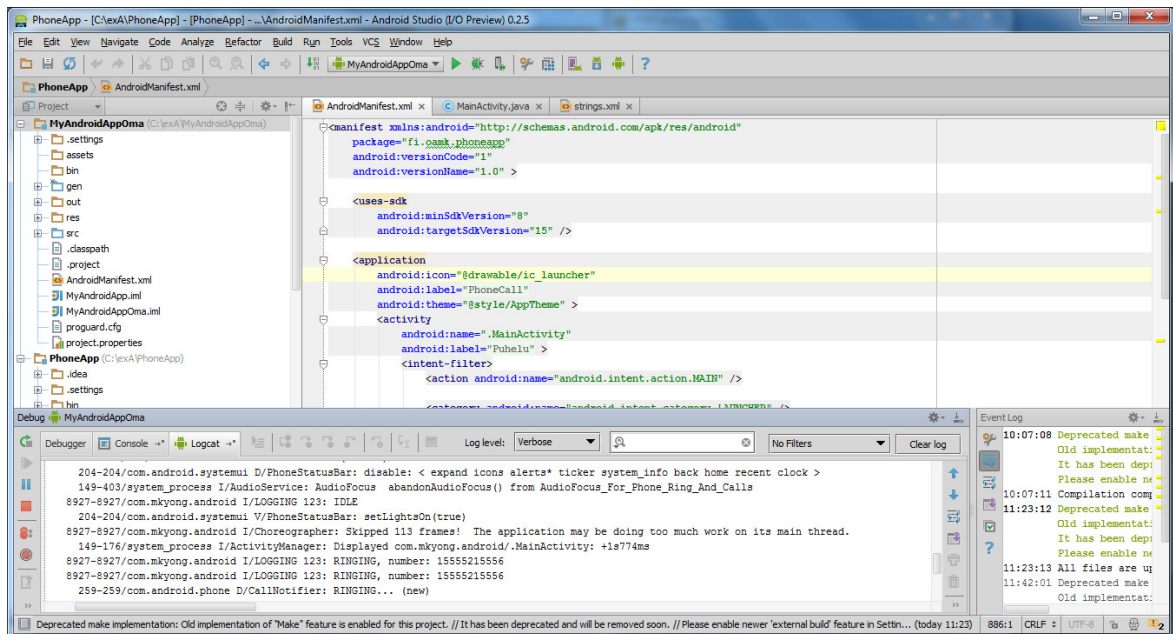Remark: .idea folder is possible to delete if same name of an application is already in use.



10) Use port numbers of emulators as call numbers, like 5556, and click phone button ..

ANDROID

Log includes all the states (Logcat) on a window when running on Debug mode:



EditText example: http://www.tutorialspoint.com/android/android_edittext_control.htm

Phone Call: http://www.tutorialspoint.com/android/android_phone_calls.htm