

ANDROID APPS.

by-
Salman A. Khan, Ishendra Agarwal, Prateek Sahu, Manas
Bhatnagar_____

Installation of the required softwares for android_____

Followings are the softwares and plugins need to be installed step by step before working on android applications :-

1. Java JDK

It is required to be able to write and compile Java applications. Follow these steps to install it.

- Go to www.oracle.com/technetwork/java/javase/downloads/index.html
- Download the latest version of Java JDK

2. Eclipse IDE

It is the Integrated Development Environment (a software used to edit, compile and run the apps) which we will use to write Android applications. Here are the steps:

- Go to www.eclipse.org/downloads/
- Download the latest version on Eclipse Classic

3. Android SDK

It is the software package which contains Android libraries, emulators and all the related stuff that one needs to make Android applications:

Go to <http://developer.android.com/sdk/index.html>

Download the latest version of Windows Installer (.exe file)

Set your browser at no proxy (since the Android SDK does not have the option for proxy authentication)

Run the above downloaded .exe file and let it download the required components

4. ADT Plugin

It is required to link Android SDK with Eclipse IDE. Go to <http://developer.android.com/sdk/eclipse-adt.html> and follow the instructions to install ADT plugin.

* Common queries :-

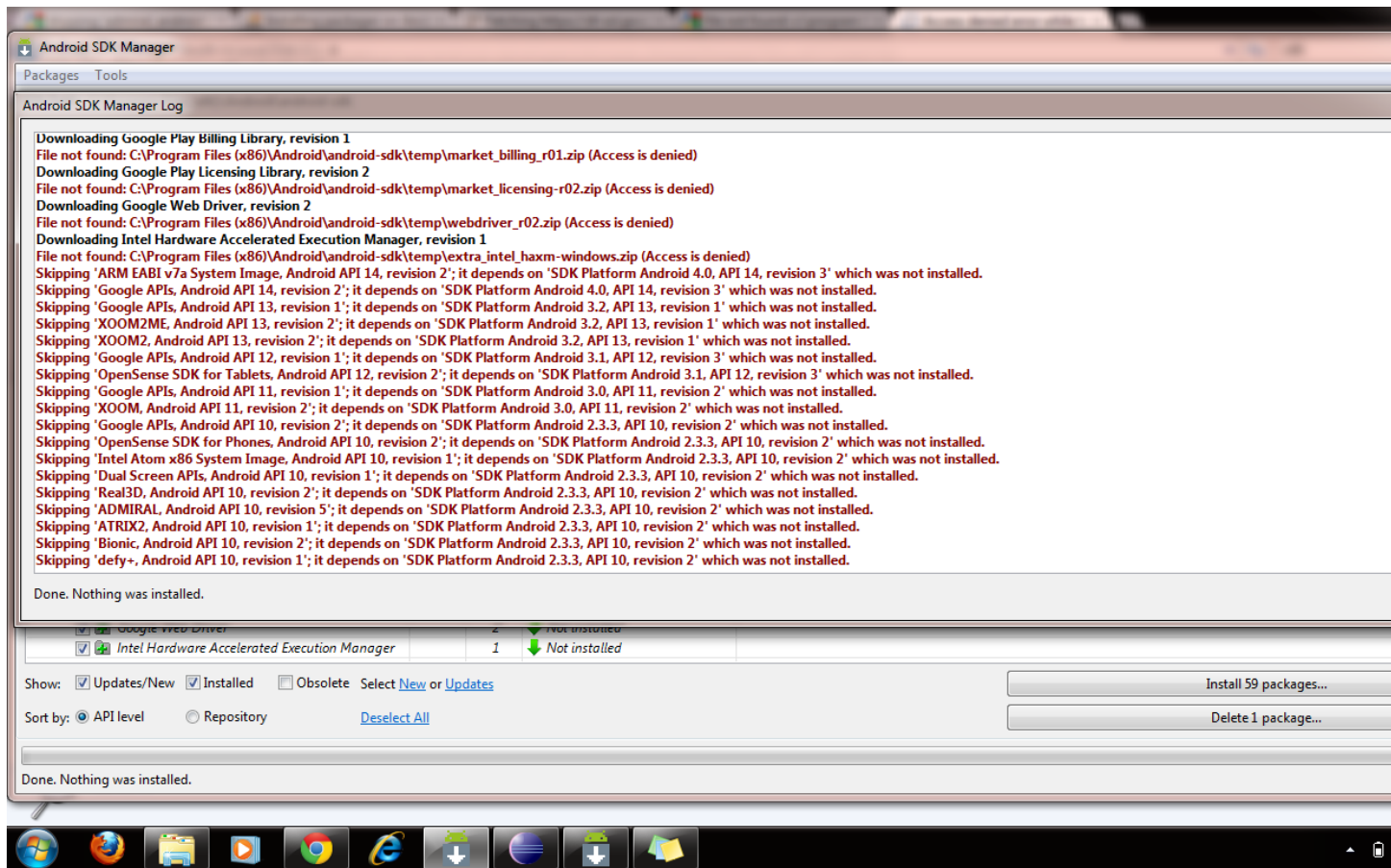
1. While installing Java JDK , go for Java platform (JDK) 7u4.
2. Install Eclipse Classic 3.7.2 when installing Eclipse IDE.
3. Don't forget to keep your browser to no proxy while running .exe file of android sdk manager.

* Problems faced by us while installing following softwares and

plugins

1. One of us firstly installed Android 4.0.3 from the SDK manager then later he was trying to install other versions of it but was unable to do so due to errors like :

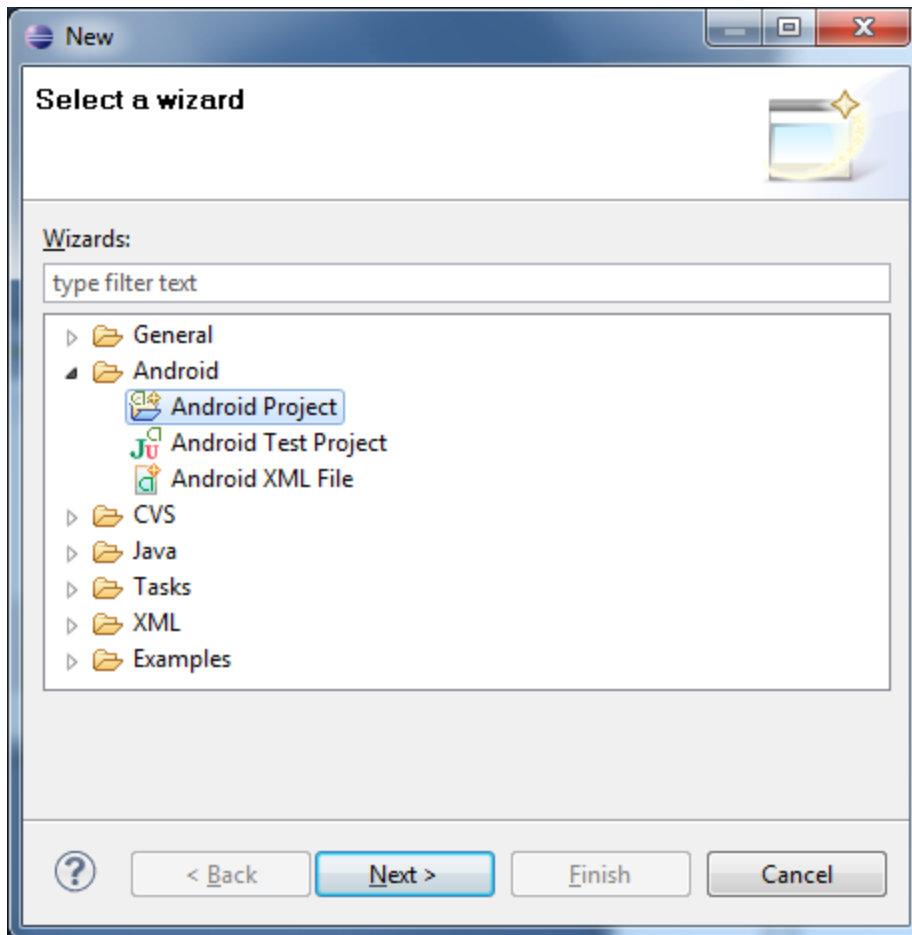
Skipping 'Google APIs, Android API 8, revision 2'; it depends on 'SDK Platform Android 2.2, API 8, revision 3' which was not installed.
Skipping 'Dual Screen APIs, Android API 8, revision 1'; it depends on 'SDK Platform Android 2.2, API 8, revision 3' which was not installed.
Skipping 'Real3D, Android API 8, revision 1'; it depends on 'SDK Platform Android 2.2, API 8, revision 3' which was not installed.
Skipping 'GALAXY Tab, Android API 8, revision 1'; it depends on 'SDK Platform Android 2.2, API 8, revision 3' which was not installed.
Done. Nothing was installed.
Downloading Samples for SDK API 13, revision 1
File not found: C:\Program Files (x86)\Android\android-sdk\temp\samples-3.2_r01-linux.zip (Access is denied)
Downloading Samples for SDK API 12, revision 1
File not found: C:\Program Files (x86)\Android\android-sdk\temp\samples-3.1_r01-linux.zip (Access is denied)
Downloading Samples for SDK API 11, revision 1
File not found: C:\Program Files (x86)\Android\android-sdk\temp\samples-3.0_r01-linux.zip (Access is denied)
Downloading Samples for SDK API 10, revision 1
File not found: C:\Program Files (x86)\Android\android-sdk\temp\samples-2.3.3_r01-linux.zip (Access is denied)
Downloading Samples for SDK API 8, revision 1
File not found: C:\Program Files (x86)\Android\android-sdk\temp\samples-2.2_r01-linux.zip (Access is denied)
Skipping 'XOOM2, Android API 13, revision 2'; it depends on 'SDK Platform Android 3.2, API 13, revision 1' which was not installed.
Skipping 'Google APIs, Android API 12, revision 1'; it depends on 'SDK Platform Android 3.1, API 12, revision 3' which was not installed.
Skipping 'OpenSense SDK for Tablets, Android API 12, revision 2'; it depends on 'SDK Platform Android 3.1, API 12, revision 3' which was not installed.
Skipping 'Google APIs, Android API 11, revision 1'; it depends on 'SDK Platform Android 3.0, API 11, revision 2' which was not installed.



The problem finally was corrected by right clicking on the Android SDK.exe file and then run it as an administrator and then again start downloading again keeping the browser in no proxy .

After finally when eclipse is installed and AVD is created , we started a new project in eclipse.The process is as follows :-

1. In eclipse , select **file>new>project** then select **android project** and then click **next** .



2.Fill the project details as:

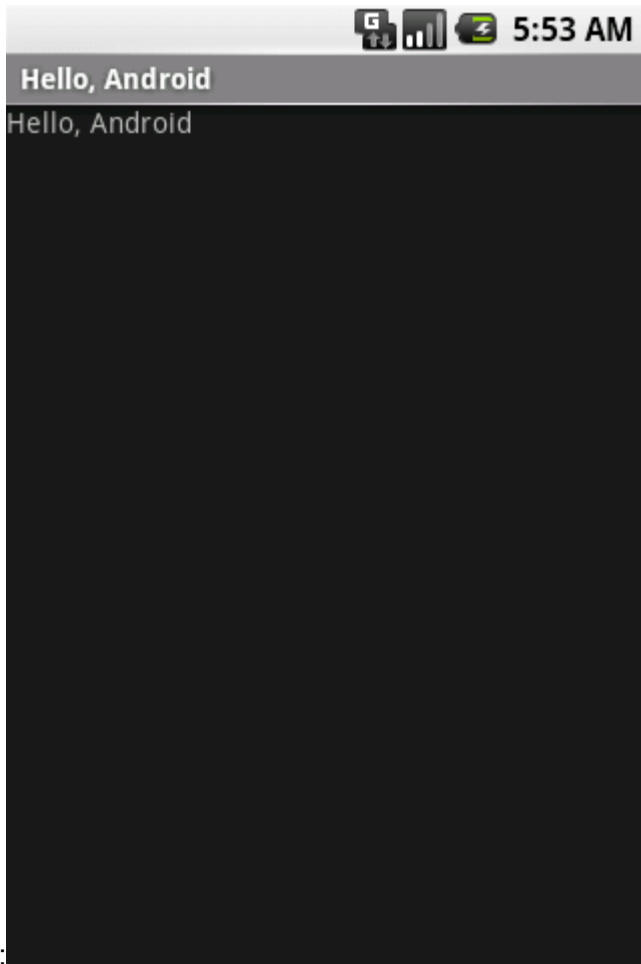
- *Project name:* HelloAndroid
- *Build Target:* Select a platform version
- *Application name:* Hello, Android
- *Package name:* com.example.helloandroid (or your own any private namespace)
- *Create Activity:* HelloAndroid

and then click on finish.

An Android project is created with a name *HelloAndroid* is created . The project will contain many

files like src,gen,Google API,Android Dependencies,assets,bin,res and some of the xml files.

The layout can be altered or adjusted by going to **res>layout>mail.xml** and the java code can be executed by going to **src>com.example.helloandroid>srcActivity.java** and can be changed.Now run the application by selecting **run>run>AndroidApplication** . The eclipse will create the new run configuration and will create the Android Emulator automatically which may take some time to run .Finally the default activity is launched and we can see something like



Finally , the first app named *hello world* is created . This was the way we proceeded to make this tutorial app .

Creating our first calculator

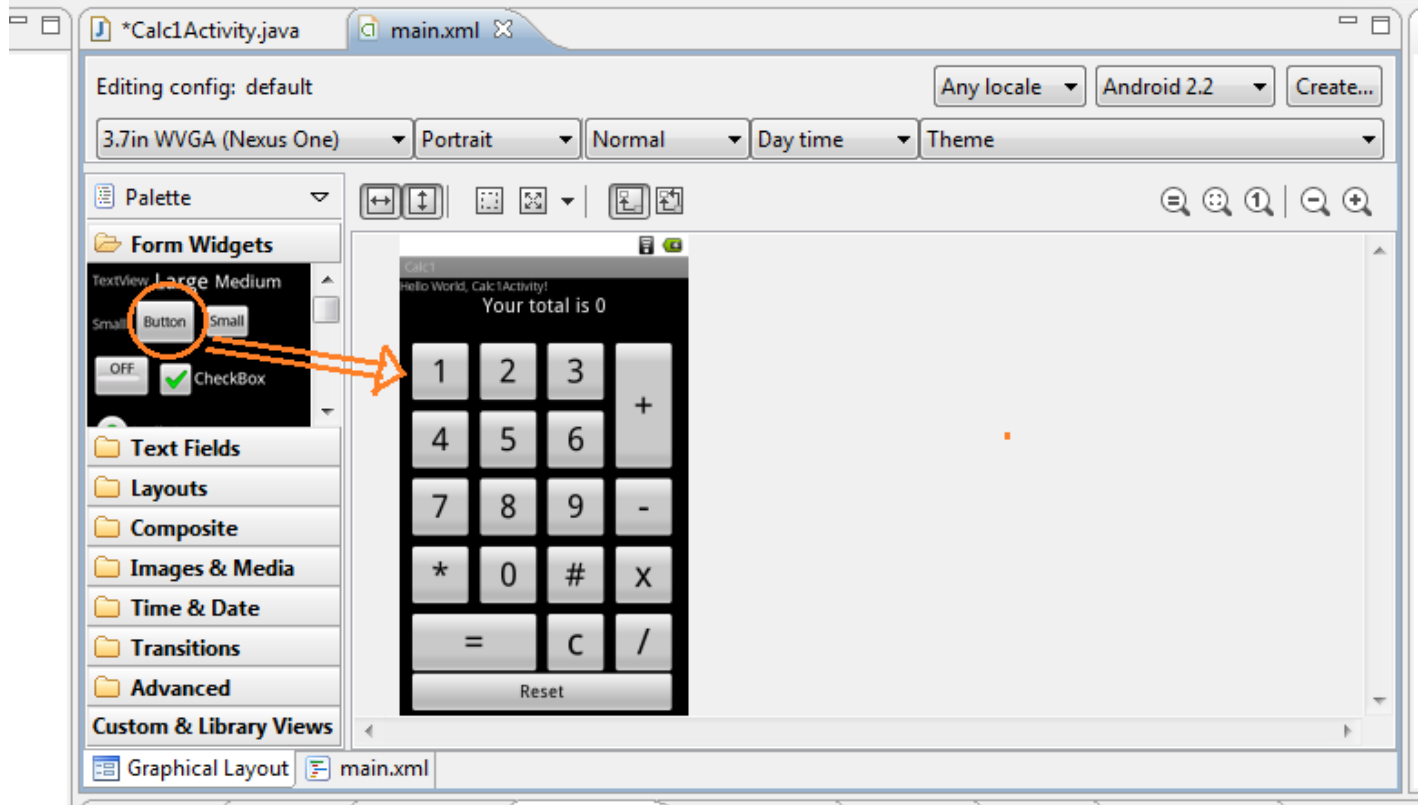
Calculator

Basic app that works as a normal calculator.

The mail.xml file has all 0-9 buttons and operations such as + , - , * & / .

The buttons used are placed in relative layout format.

Buttons can be place don the screen through the mail.xml file.



main.xml file can be found in the layout folder in res file.

I have used co-ordinates of buttons for the layout such as

```

        android:textAppearance="?android:attr/text

<Button
    android:id="@+id/b1"
    android:layout_width="70dp"
    android:layout_height="70dp"
    android:layout_marginLeft="10dp"
    android:layout_marginTop="70dp"
    android:text="@string/button1"
    android:textSize="35dp" />

<Button
    android:id="@+id/b2"
    android:layout_width="70dp"
    android:layout_height="70dp"
    android:layout_marginLeft="85dp"
    android:layout_marginTop="70dp"
    android:text="@string/button2"
    android:textSize="35dp" />

```

Or we can use normal relative layout using ids of buttons.

In the .java file,I have created

Buttons have been given variables as follows

```

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
    counter = 0;
    b1 = (Button) findViewById(R.id.b1);
    b2 = (Button) findViewById(R.id.b2);
    b3 = (Button) findViewById(R.id.b3);
    b4 = (Button) findViewById(R.id.b4);
    b5 = (Button) findViewById(R.id.b5);
    b6 = (Button) findViewById(R.id.b6);
    b7 = (Button) findViewById(R.id.b7);
    b8 = (Button) findViewById(R.id.b8);
    b9 = (Button) findViewById(R.id.b9);
    b0 = (Button) findViewById(R.id.b0);
    plus = (Button) findViewById(R.id.bplus);
    minus = (Button) findViewById(R.id.bminus);
    mult = (Button) findViewById(R.id.bmult);
    div = (Button) findViewById(R.id.bdiv);
    equal = (Button) findViewById(R.id.bequal);
    clear = (Button) findViewById(R.id.bc);
    reset = (Button) findViewById(R.id.b_reset);
    display = (TextView) findViewById(R.id.txt_res);
}

```

And every button is assigned an id as defined in the main.xml file
 Onsetclick listener for all numerical buttons are defines as follows

```

b1.setOnClickListener(new View.OnClickListener() {

    public void onClick(View v) {
        // TODO Auto-generated method stub
        if(temp2=='=')
        {
            counter=0;
            temp2='0';
        }
        if(temp1== '+' || temp1=='-' || temp1=='x' || temp1=='=' || temp1=='/')
        {
            temp=1;
            temp1='0';
        }
        else
            temp=temp*10+1;
        display.setText("Your total is " + temp);
    }
});

```

And for operations:

```

        plus.setOnClickListener(new View.OnClickListener() {
public void onClick(View v) {
    // TODO Auto-generated method stub

    if (temp2=='+'||temp2=='0')
        counter=counter+temp;
    else if (temp2=='-')
        counter=counter-temp;
    else if (temp2=='x')
        counter=counter*temp;
    else if (temp2=='/')
        counter=counter/temp;
    temp2=temp1+'+';
    display.setText("Your total is " + counter + " + ");
}
});

```

Now coming to the coding part,

Whenever we start the application or click reset,our counter(for result) and all temporary variables goes to 0(or '0' in character data type)

Now on clicking a number we save the number in temp. after that if another number is typed (i.e operations are not performed) then

```

else
    temp=temp*10+9;
display.setText("Your total is " + temp);

```

Ensures that the number is saved as the first digit being tens place and second digit being in ones place,and so on.

Now as we use any operator symbol our value in temp is stored into counter and is displayed on the screen. again a character which is same as the operator is stored into temp1 and temp2 variables.

Now as we type a number our 1st condition occurs i.e.

```

if(temp1== '+' || temp1=='-' || temp1=='x' || temp1=='=' || temp1=='/' )
{
    temp=9;
    temp1='0';
}

```

Hence our temp1 goes into default value(such that there is no problem while typing other digits again) and temp stores the value corresponding to the button clicked.

Now on the operators onclick code,we perform the corresponding operation of the resultant value with the next input value.this happens due to the if else code in each operator button


```
plus.setOnClickListener(new View.OnClickListener() {  
public void onClick(View v) {  
    // TODO Auto-generated method stub  
  
    if (temp2=='+'||temp2=='0')  
        counter=counter+temp;  
    else if (temp2=='-')  
        counter=counter-temp;  
    else if (temp2=='x')  
        counter=counter*temp;  
    else if (temp2=='/')  
        counter=counter/temp;  
    temp2=temp1+'+';  
    display.setText("Your total is "+ counter + " + ");  
    }  
});
```

And hence the output in the screen.