Android Application that makes use of RSS Feed

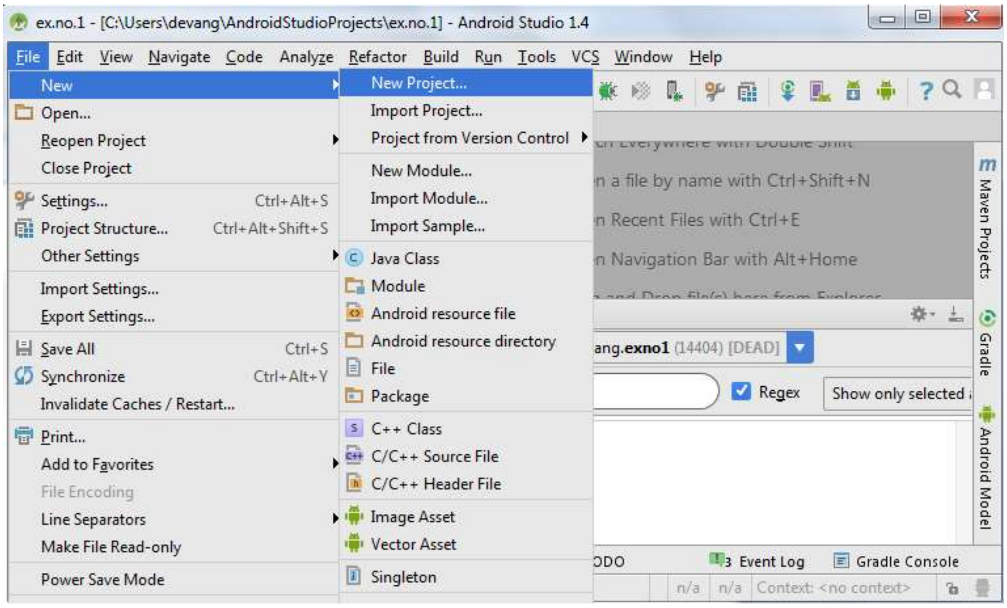
Aim:

To develop a Android Application that makes use of RSS Feed.

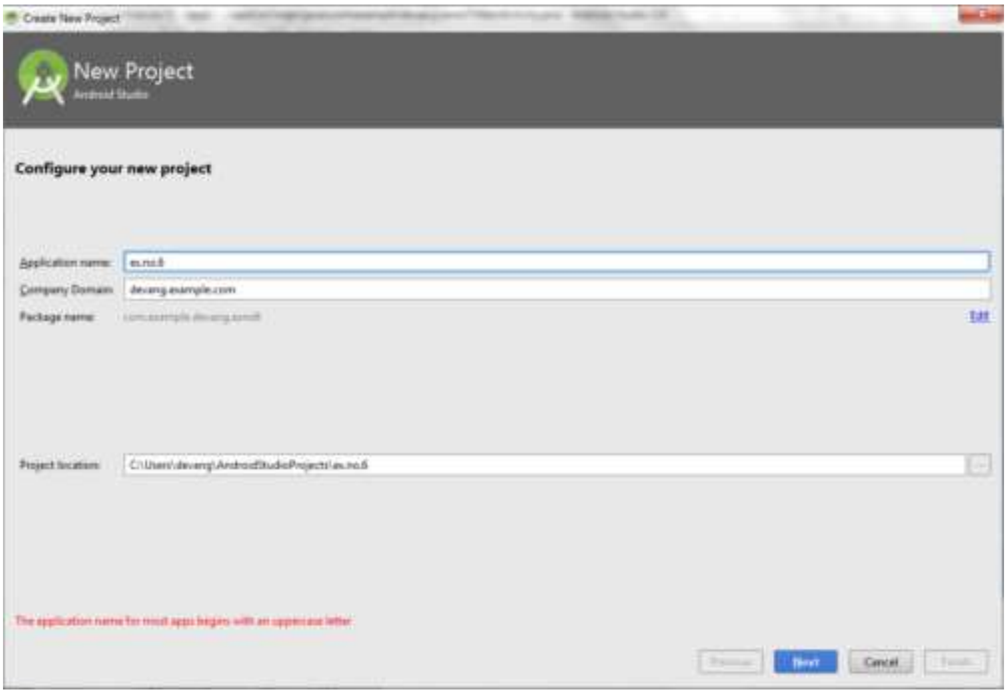
Procedure:

Creating a New project:

- Open Android Studio and then click on **File -> New -> New project.**



- Then type the Application name as “**ex.no.6″** and click **Next.**



Configure your new project

Application name: es.no.6

Company Domain: devang.example.com

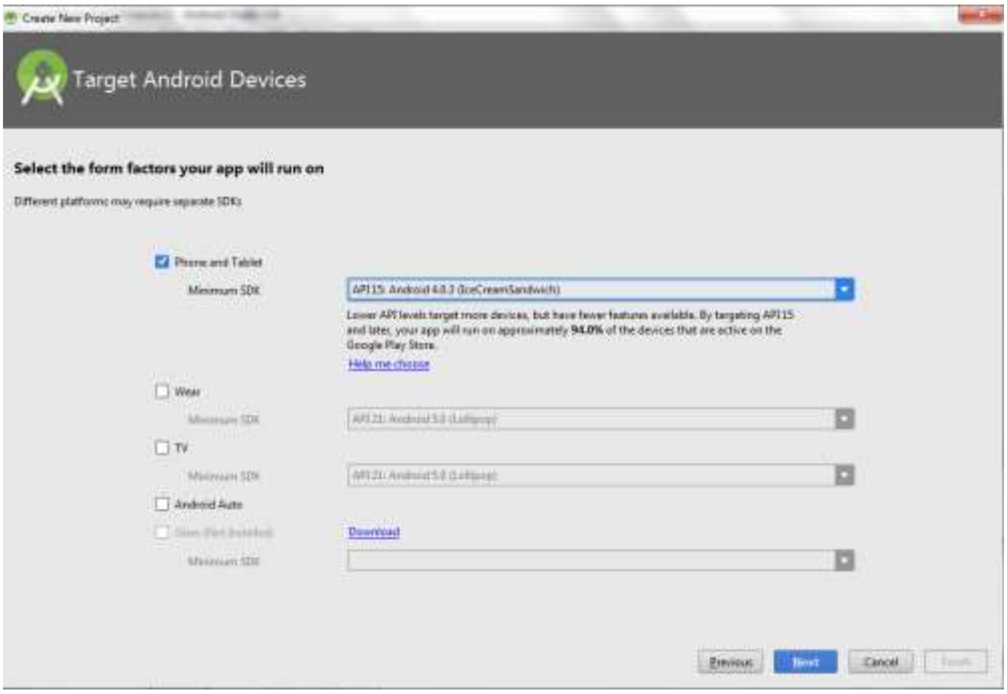
Package name: com.example.devang.esno6

Edit

Project location: C:\Users\devang\AndroidStudioProjects\es.no.6

The application name for most apps begins with an uppercase letter

- Then select the **Minimum SDK** as shown below and click **Next**.



Select the form factors your app will run on

Different platforms may require separate SDKs

☑ Phone and Tablet

Minimum SDK: API15: Android 4.0.3 (IceCreamSandwich)

Lower API levels target more devices, but have fewer features available. By targeting API15 and later, your app will run on approximately **94.0%** of the devices that are active on the Google Play Store.

Help me choose

□ Wear

Minimum SDK: API 21: Android 5.0 (Lollipop)

□ TV

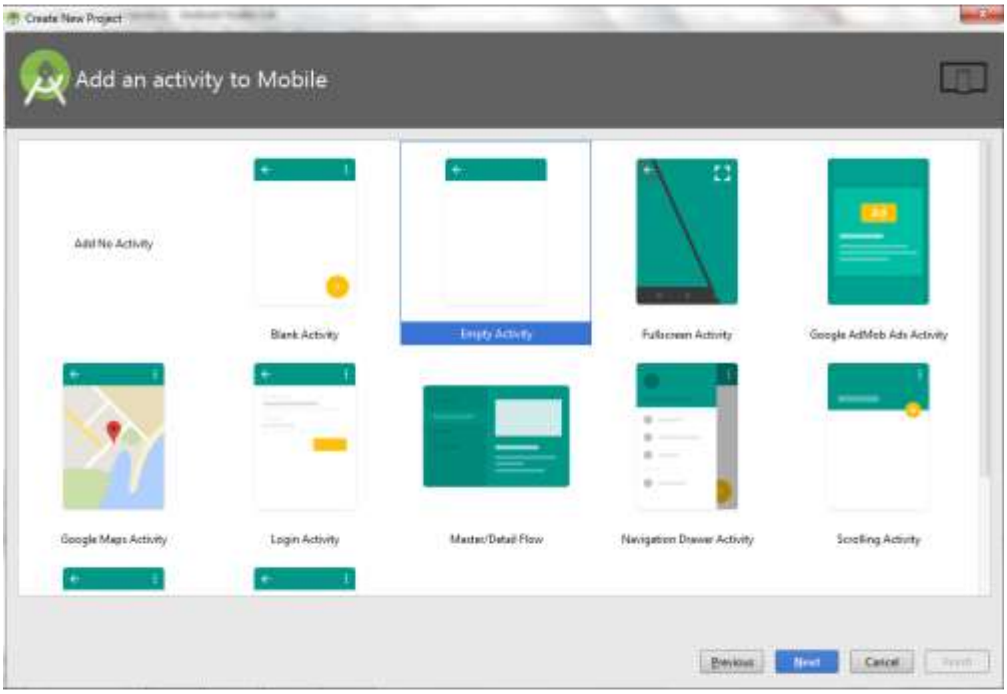
Minimum SDK: API 21: Android 5.0 (Lollipop)

□ Android Auto

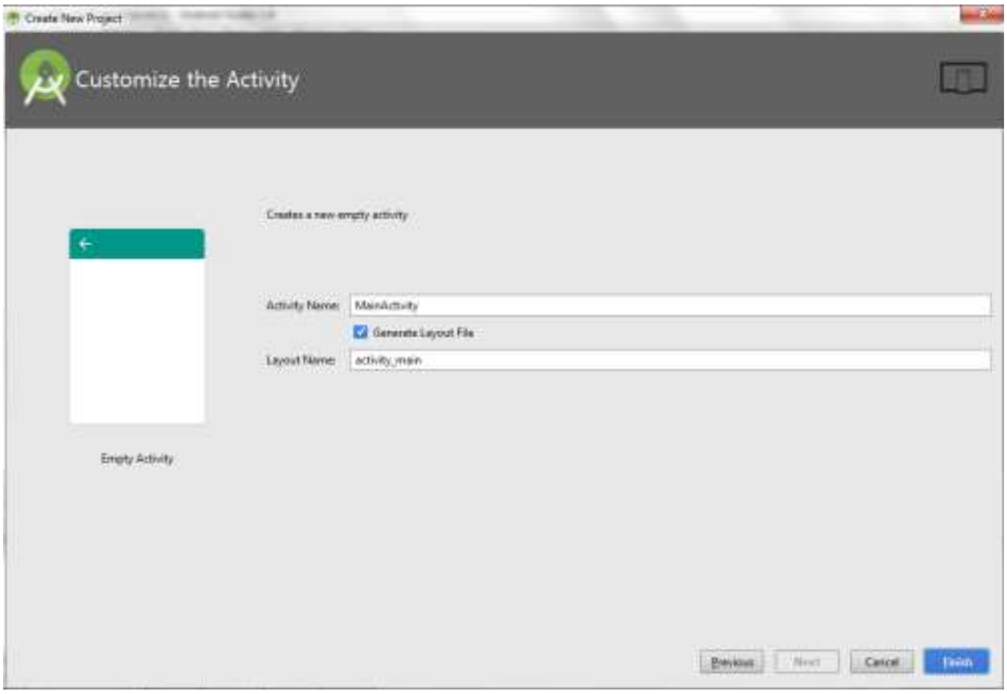
□ Glass (Not Installed) Download

Minimum SDK

- Then select the **Empty Activity** and click **Next.**



- Finally click **Finish**.



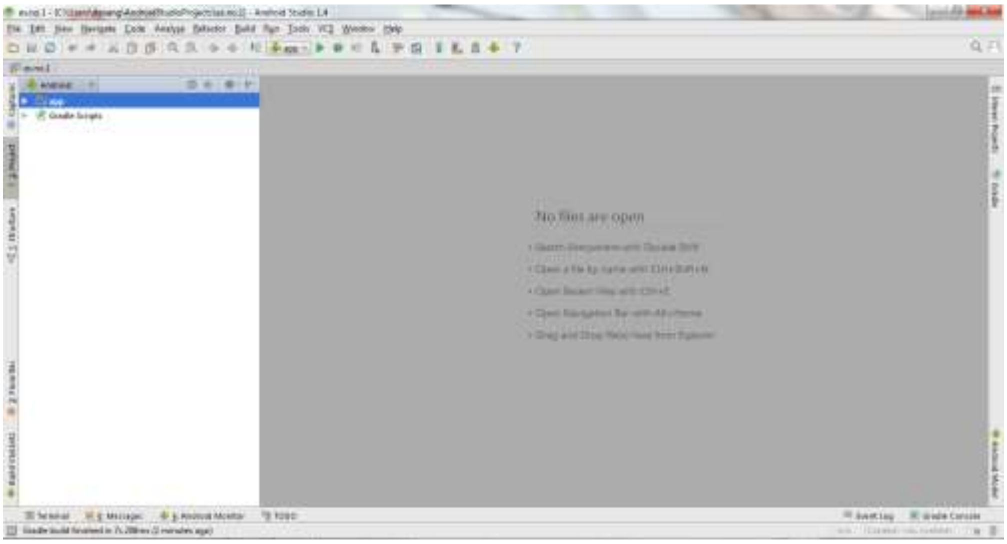
Creates a new empty activity

Activity Name: MainActivity

☑ Generate Layout File

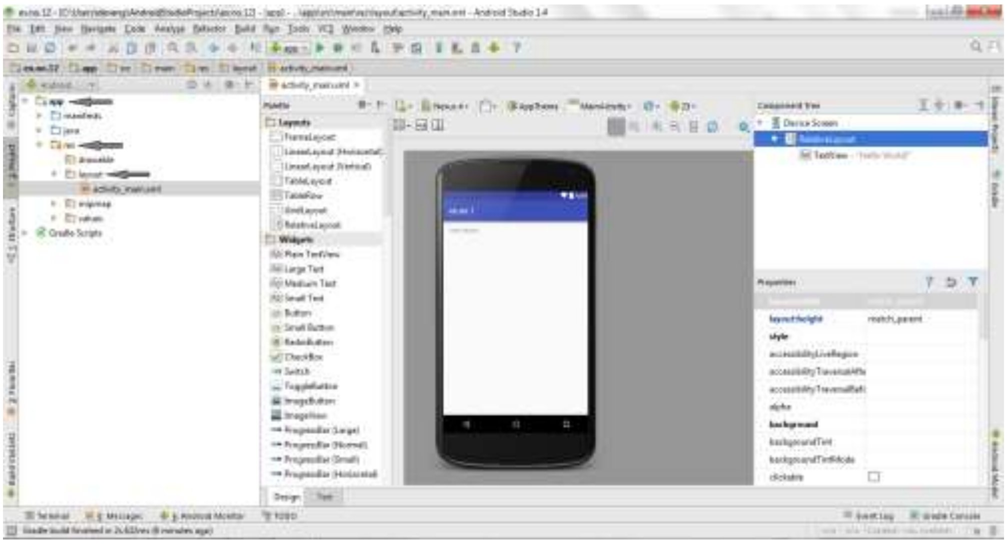
Layout Name: activity_main

- It will take some time to build and load the project.
- After completion it will look as given below.

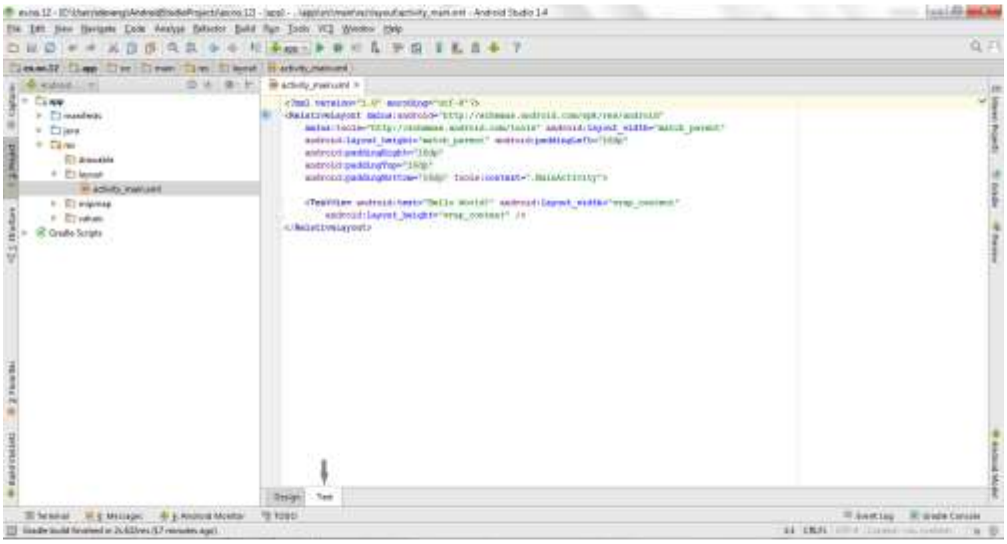


Designing layout for the Android Application:

- Click on **app -> res -> layout -> activity_main.xml**



- Now click on **Text** as shown below.



```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
    android:layout_height="match_parent" android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    android:paddingBottom="16dp" tools:context=".MainActivity">

    <TextView android:text="Hello World!" android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</RelativeLayout>
```

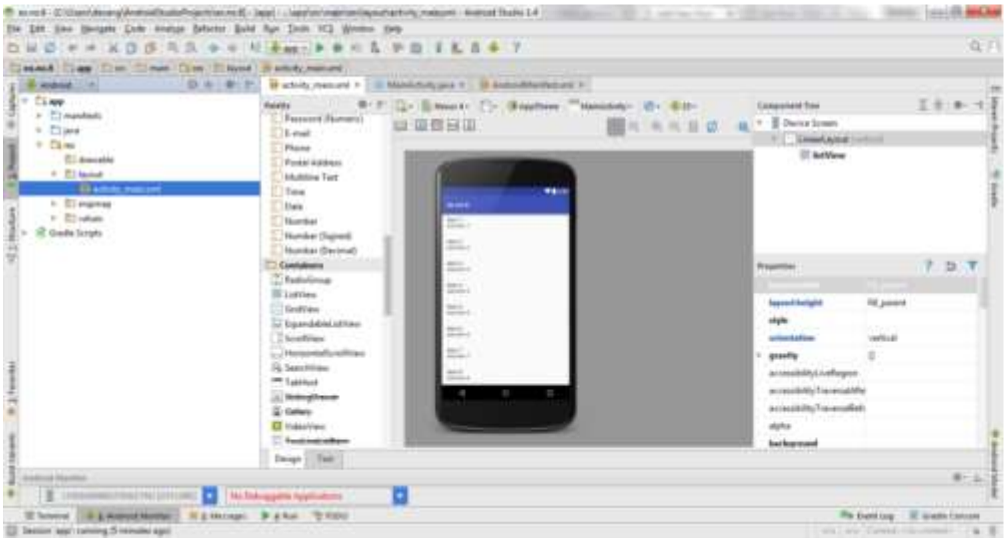
- Then delete the code which is there and type the code as given below.

Code for Activity_main.xml:

?

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

    <ListView
        android:id="@+id/listView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

</LinearLayout>
```

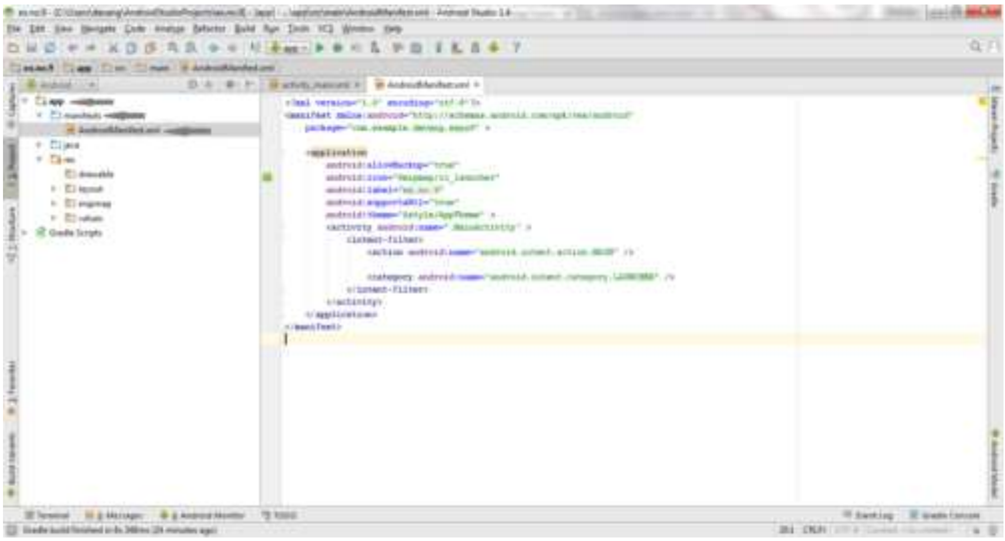
- Now click on **Design** and your application will look as given below.



- So now the designing part is completed.

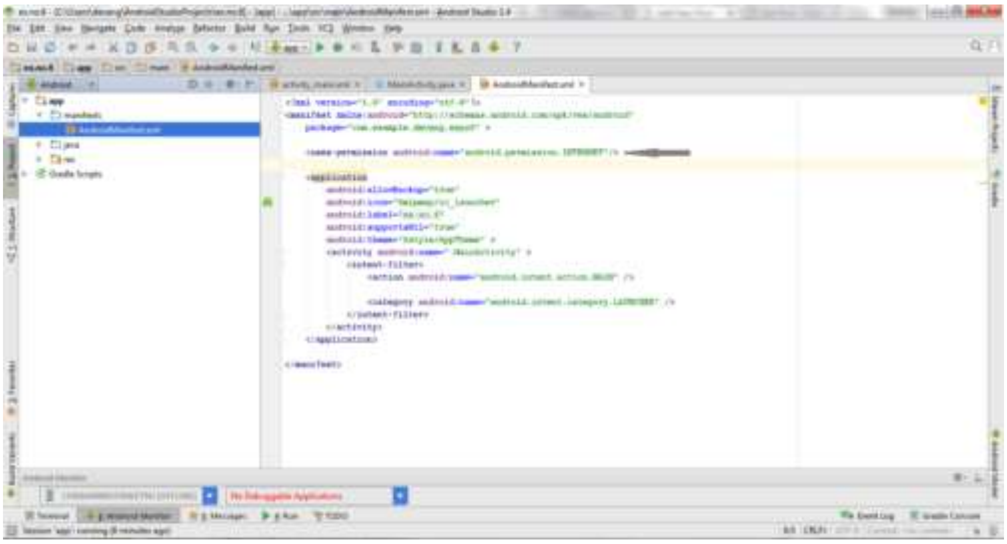
Adding permissions in Manifest for the Android Application:

- Click on **app -> manifests -> AndroidManifest.xml**



```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="[illegible]" >

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="[illegible]"
        android:supportsRtl="true"
        android:theme="@style/AppTheme" >
        <activity android:name=".MainActivity" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

- Now include the **INTERNET** permissions in the AndroidManifest.xml file as shown below



```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="[illegible]" >

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="[illegible]"
        android:supportsRtl="true"
        android:theme="@style/AppTheme" >
        <activity android:name=".MainActivity" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

Code for AndroidManifest.xml:

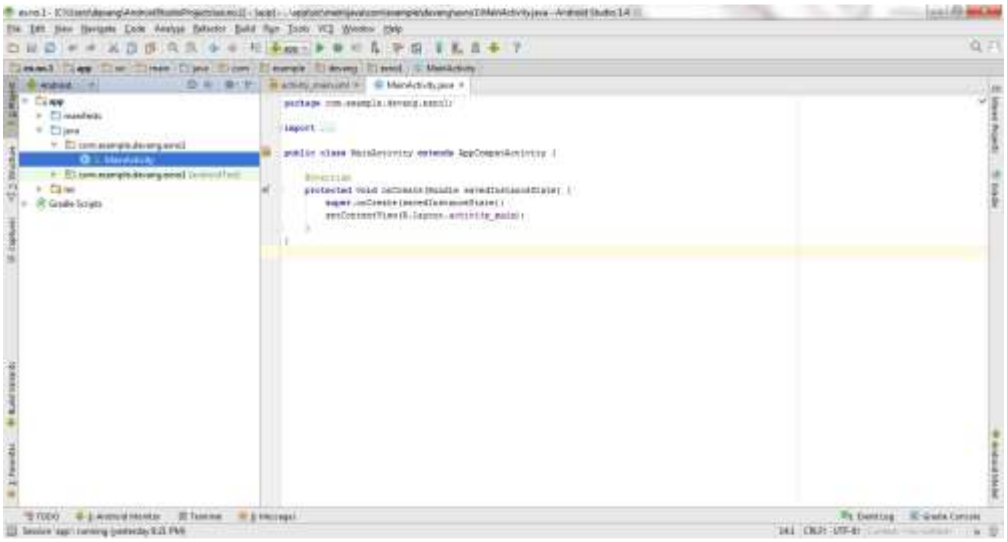
?

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.exno6" >

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/AppTheme" >
        <activity android:name=".MainActivity" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

- So now the Permissions are added in the Manifest.

Java Coding for the Android Application:

- Click on **app -> java -> com.example.exno6 -> MainActivity.**



```
package com.example.devang.exno11;

import ...

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

- Then delete the code which is there and type the code as given below.

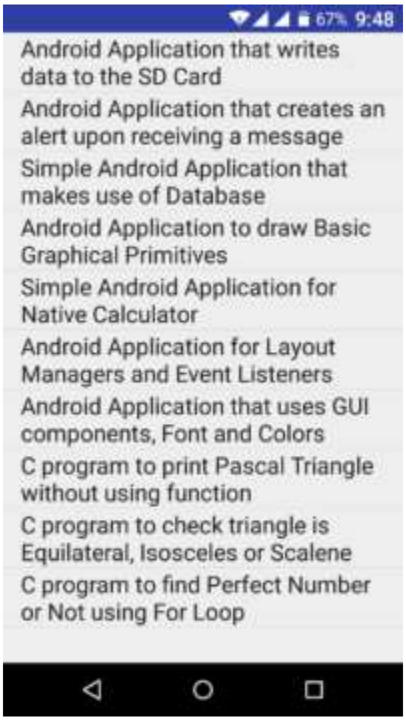
Code for MainActivity.java:

?

```
package com.example.exno6;

import android.app.ListActivity;
import android.content.Intent;
import android.net.Uri;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import org.xmlpull.v1.XmlPullParser;
import org.xmlpull.v1.XmlPullParserException;
import org.xmlpull.v1.XmlPullParserFactory;
import java.io.IOException;
import java.io.InputStream;
import java.net.MalformedURLException;
import java.net.URL;
import java.util.ArrayList;
import java.util.List;

public class MainActivity extends ListActivity
{
    List headlines;
    List links;

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        new MyAsyncTask().execute();
    }

    class MyAsyncTask extends AsyncTask<Object,Void,ArrayAdapter>
    {
        @Override
        protected ArrayAdapter doInBackground(Object[] params)
        {
            headlines = new ArrayList();
            links = new ArrayList();
            try
            {
                URL url = new URL("https://codingconnect.net/feed");
                XmlPullParserFactory factory =
XmlPullParserFactory.newInstance();
                factory.setNamespaceAware(false);
                XmlPullParser xpp = factory.newPullParser();

                // We will get the XML from an input stream
                xpp.setInput(getInputStream(url), "UTF_8");
                boolean insideItem = false;

                // Returns the type of current event: START_TAG,
END_TAG, etc..
                int eventType = xpp.getEventType();
                while (eventType != XmlPullParser.END_DOCUMENT)
                {
                    if (eventType == XmlPullParser.START_TAG)
                    {
                        if (xpp.getName().equalsIgnoreCase("item"))
                        {
                            insideItem = true;
                        }
                        else if (xpp.getName().equalsIgnoreCase("title"))
                        {
                            if (insideItem)
                                headlines.add(xpp.nextText()); //extract
the headline
                        }
                        else if (xpp.getName().equalsIgnoreCase("link"))
                        {
                            if (insideItem)
                                links.add(xpp.nextText()); //extract the
link of article
                        }
                    }
                    else if(eventType==XmlPullParser.END_TAG &&
xpp.getName().equalsIgnoreCase("item"))
                    {
                        insideItem=false;
                    }
                    eventType = xpp.next(); //move to next element
                }

            }
            catch (MalformedURLException e)
            {
                e.printStackTrace();
            }
            catch (XmlPullParserException e)
            {
                e.printStackTrace();
            }
            catch (IOException e)
            {
                e.printStackTrace();
            }
            return null;
        }
        protected void onPostExecute(ArrayAdapter adapter)
        {
            adapter = new ArrayAdapter(MainActivity.this,
android.R.layout.simple_list_item_1, headlines);
            setListAdapter(adapter);
        }
    }

    @Override
    protected void onListItemClick(ListView l, View v, int position, long
id)
    {
        Uri uri = Uri.parse((links.get(position)).toString());
        Intent intent = new Intent(Intent.ACTION_VIEW, uri);
        startActivity(intent);
    }

    public InputStream getInputStream(URL url)
    {
        try
        {
            return url.openConnection().getInputStream();
        }
        catch (IOException e)
        {
            return null;
        }
    }
}
```

- So now the Coding part is also completed.
- Now run the application to see the output.

Output:



Android Application that writes data to the SD Card

Android Application that creates an alert upon receiving a message

Simple Android Application that makes use of Database

Android Application to draw Basic Graphical Primitives

Simple Android Application for Native Calculator

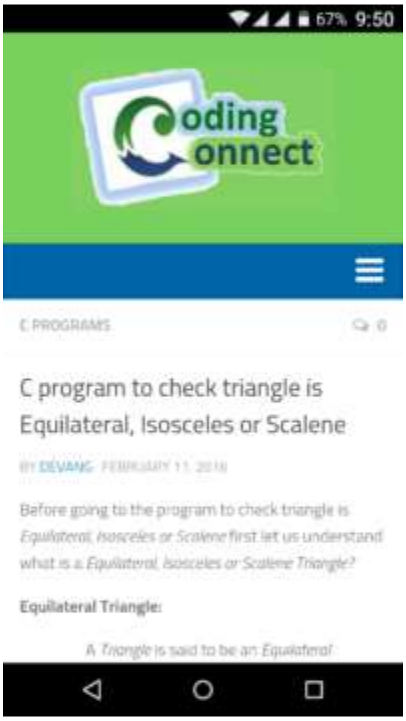
Android Application for Layout Managers and Event Listeners

Android Application that uses GUI components, Font and Colors

C program to print Pascal Triangle without using function

C program to check triangle is Equilateral, Isosceles or Scalene

C program to find Perfect Number or Not using For Loop



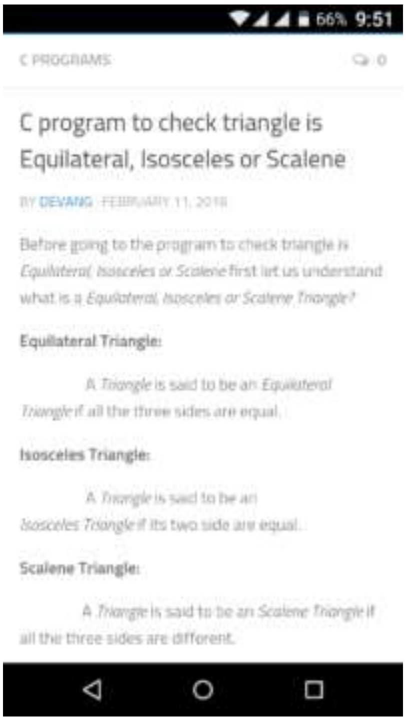
C program to check triangle is Equilateral, Isosceles or Scalene

BY DEVANG · FEBRUARY 11, 2018

Before going to the program to check triangle is *Equilateral, Isosceles or Scalene* first let us understand what is a *Equilateral, Isosceles or Scalene Triangle?*

Equilateral Triangle:

A *Triangle* is said to be an *Equilateral*



C program to check triangle is Equilateral, Isosceles or Scalene

BY DEVANG · FEBRUARY 11, 2018

Before going to the program to check triangle is *Equilateral, Isosceles or Scalene* first let us understand what is a *Equilateral, Isosceles or Scalene Triangle?*

Equilateral Triangle:

A *Triangle* is said to be an *Equilateral Triangle* if all the three sides are equal.

Isosceles Triangle:

A *Triangle* is said to be an *Isosceles Triangle* if its two side are equal.

Scalene Triangle:

A *Triangle* is said to be an *Scalene Triangle* if all the three sides are different.

Result:

Thus Android Application that makes use of RSS Feed is developed and executed successfully.