

MOBILE COMPUTING PRACTICAL

EX NO : 01

LIFECYCLE ACTIVITY

AIM

To demonstrate android application for Life Cycle activity (Application Life Cycle)

Procedure

1. Open eclipse or android studio and select new android project
2. Give project name and select next
3. Choose the android version. Choose the lowest android version(Android 2.2) and select next
4. Enter the package name. Package name must be two word separated by comma and click finish
5. Go to package explorer in the left hand side. Select our project.
6. Go to res folder and select layout. Double click the main.xml file. Don't change anything in layout. Leave as default.
7. Now select mainactivity.java file and to create java class file to type based on android application for Life Cycle activity
8. Run the application to launch Android emulator and verify the result of the changes done in the application.

lifecyleActivity.java

```
package your.welcome.namespace;
import android.app.Activity;
import android.os.Bundle;
import android.widget.Toast;
public class WelcomeActivity extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        Toast.makeText(WelcomeActivity.this, "ON CREATE",
Toast.LENGTH_SHORT).show();
    }
    @Override
    protected void onStart() {

        super.onStart();
        Toast.makeText(WelcomeActivity.this, "ON START",
Toast.LENGTH_SHORT).show();
    }
    @Override
    protected void onResume() {
```

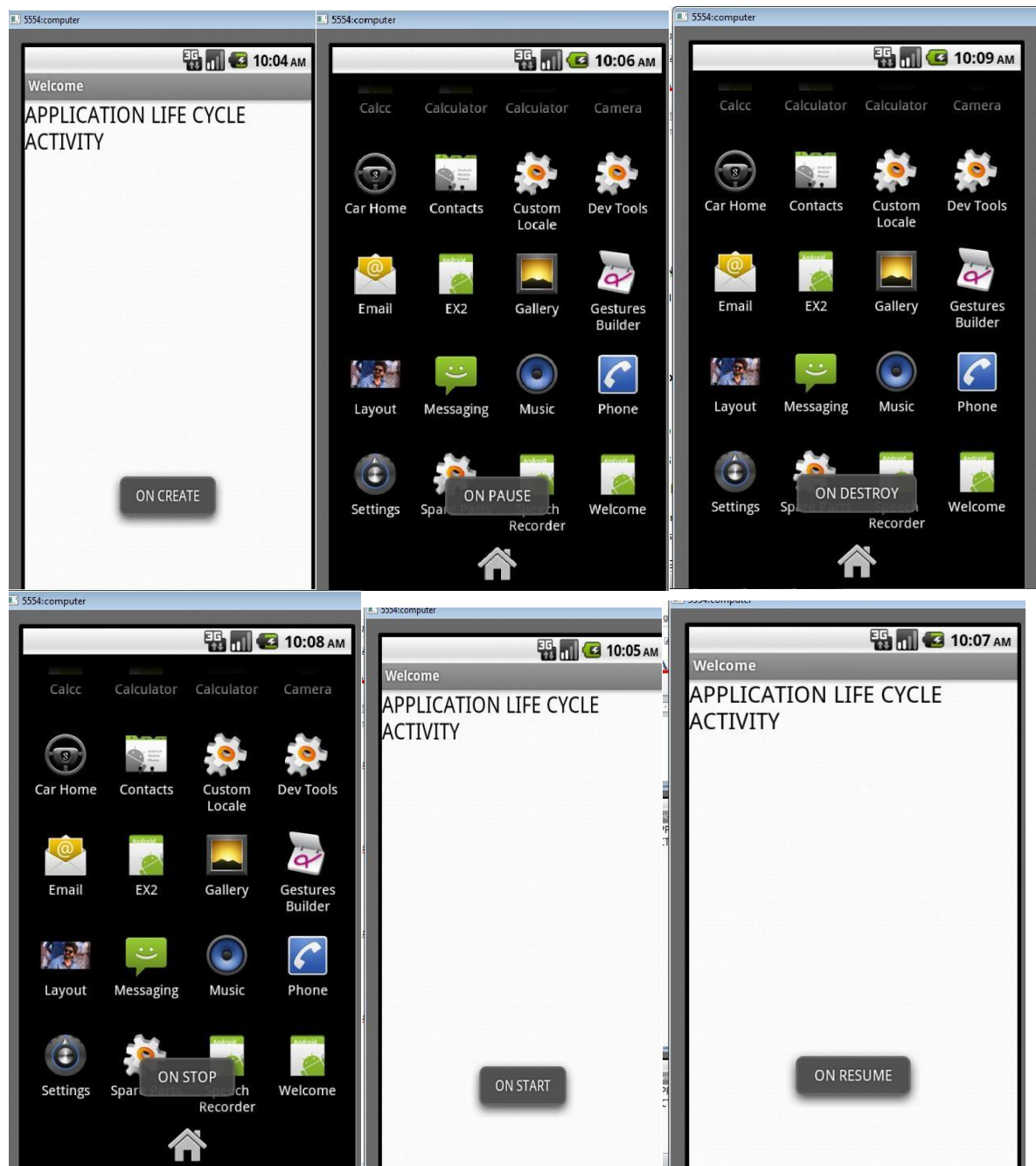
```
        super.onResume();
        Toast.makeText(WelcomeActivity.this, "ON RESUME",
Toast.LENGTH_SHORT).show();
    }
    @Override
    protected void onPause() {

        super.onPause();
        Toast.makeText(WelcomeActivity.this, "ON PAUSE",
Toast.LENGTH_SHORT).show();
    }
    @Override
    protected void onRestart() {

        super.onRestart();
        Toast.makeText(WelcomeActivity.this, "ON RESTART",
Toast.LENGTH_SHORT).show();
    }
    @Override
    protected void onStop() {

        super.onStop();
        Toast.makeText(WelcomeActivity.this, "ON STOP",
Toast.LENGTH_SHORT).show();
    }
    @Override
    protected void onDestroy() {

        super.onDestroy();
        Toast.makeText(WelcomeActivity.this, "ON DESTROY",
Toast.LENGTH_SHORT).show();
    }
}
```



Result

Thus successfully demonstrate android application for Life Cycle activity (Application Life Cycle)

EX NO : 02

DIFFERENT TYPES OF

LAYOUT AIM

To demonstrate android application for different types of layouts

Procedure

1. Open eclipse or android studio and select new android project
2. Give project name and select next
3. Choose the android version. Choose the lowest android version(Android 2.2) and select next
4. Enter the package name. Package name must be two word separated by comma and click finish
5. Go to package explorer in the left hand side. Select our project.
6. Go to res folder and select layout. Double click the main.xml file. To design different types of layouts in single page.
7. Now select mainactivity.java file and to create java class file and Don't change anything. Leave as default.
8. Run the application to launch Android emulator and verify the result of the changes done in the application.

Layoutactivity

```
package your.layout.namespace;
```

```
import  
android.app.Activity;  
import  
android.os.Bundle;
```

```
public class LayoutActivity extends Activity {  
    /** Called when the activity is first  
    created. */ @Override  
    public void onCreate(Bundle  
        savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.main);  
    }  
}
```

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:background="#FF0000"
android:orientation="vertical" >

<TextView
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:text="LINEARLAYOUT"
    android:textAppearance="?android:attr/textAppearanceLarge" />

<RelativeLayout
    android:id="@+id/relativeLayout1"
    android:layout_width="match_parent"
    android:layout_height="266dp"
    android:background="#0000FF" >

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:layout_marginLeft="30dp"
        android:layout_marginTop="47dp"
        android:text="USERNAME"
        android:textAppearance="?android:attr/textAppearanceLarge" />

    <TextView
        android:id="@+id/textView3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/textView2"
        android:layout_below="@+id/textView2"
        android:layout_marginTop="28dp"
        android:text="PASSWORD"
        android:textAppearance="?android:attr/textAppearanceLarge" />

    <EditText
        android:id="@+id/editText1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentRight="true"
        android:layout_alignTop="@+id/textView2"
        android:layout_marginLeft="26dp"
        android:layout_toRightOf="@+id/textView2" >

        <requestFocus />
    </EditText>

    <EditText
        android:id="@+id/editText2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/editText1"
        android:layout_alignParentRight="true"
        android:layout_alignTop="@+id/textView3"
        android:inputType="textPassword" />

```

```

<Button
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/editText2"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="22dp"
    android:text="LOGIN" />

<TextView
    android:id="@+id/textView4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
    android:text="RELATIVELAYOUT"
    android:textAppearance="?android:attr/textAppearanceLarge" />

</RelativeLayout>

<TableLayout
    android:id="@+id/tableLayout1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#00FF00" >

    <TableRow
        android:id="@+id/tableRow1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" >

        <Button
            android:id="@+id/button2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="1" />

        <Button
            android:id="@+id/button4"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="2" />

        <Button
            android:id="@+id/button5"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="3" />

        <Button
            android:id="@+id/button6"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="4" />

        <Button
            android:id="@+id/button7"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="5" />

```

```

</TableRow>

<TableRow
    android:id="@+id/tableRow2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" >

    <Button
        android:id="@+id/button3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="6" />

    <Button
        android:id="@+id/button8"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="7" />

    <Button
        android:id="@+id/button9"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="8" />

    <Button
        android:id="@+id/button10"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="9" />

    <Button
        android:id="@+id/button11"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="10" />

    <TextView
        android:id="@+id/textView5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="TABLE LAYOUT"
        android:textColor="#800000"
        android:textAppearance="?android:attr/textAppearanceLarge" />

</TableRow>

<TableRow
    android:id="@+id/tableRow3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" >

    <Button
        android:id="@+id/button12"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="+" />

    <Button
        android:id="@+id/button13"
        android:layout_width="wrap_content"

```



```

        android:layout_height="wrap_content"
        android:text="-" />

<Button
    android:id="@+id/button14"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="*" />

<Button
    android:id="@+id/button15"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="/" />

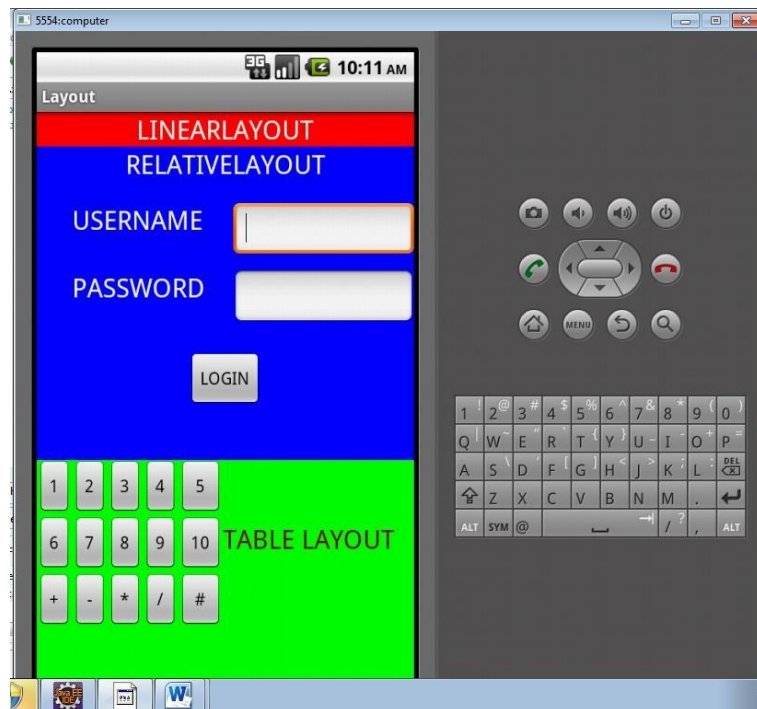
<Button
    android:id="@+id/button16"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="#" />

</TableRow>

</TableLayout>

</LinearLayout>

```



RESULT

Thus successfully demonstrate android application for different types of layouts using single page.

EX NO : 03

SIMPLE CALCULATOR

AIM

Write a program to implement simple calculator using text view, edit view and button

Procedure

1. Open eclipse or android studio and select new android project
2. Give project name and select next
3. Choose the android version. Choose the lowest android version(Android 2.2) and select next
4. Enter the package name. Package name must be two and click finish
5. Go to package explorer in the left hand side. Select our project.
6. Go to res folder and select layout. Double click the main.xml file. Using RelativeLayout design simple calculator layout.
7. Now select MainActivity.java file and to create java class file to type simple calculator using text view, edit view and button
8. Run the application to launch Android emulator and verify the result of the changes done in the application.

Calculator Activity

```
package your.calculator.namespace;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;

public class CalculatorActivity extends Activity {
    /** Called when the activity is first created. */
    EditText e1,e2,e3;
    int a,b,c;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
    public void Add(View v)
    {
        e1=(EditText) findViewById(R.id.editText1);
        e2=(EditText) findViewById(R.id.editText2);
        e3=(EditText) findViewById(R.id.editText3);
```

```

        String s1=e1.getText().toString();
        String s2=e2.getText().toString();

        a=Integer.parseInt(s1);
        b=Integer.parseInt(s2);

        c=a+b;

        String s3=Integer.toString(c);

        e3.setText(s3);

    }
    public void sub(View v)
    {
        e1=(EditText) findViewById(R.id.editText1);
        e2=(EditText) findViewById(R.id.editText2);
        e3=(EditText) findViewById(R.id.editText3);

        String s1=e1.getText().toString();
        String s2=e2.getText().toString();

        a=Integer.parseInt(s1);
        b=Integer.parseInt(s2);

        c=a-b;

        String s3=Integer.toString(c);

        e3.setText(s3);

    }
    public void Mul(View v)
    {
        e1=(EditText) findViewById(R.id.editText1);
        e2=(EditText) findViewById(R.id.editText2);
        e3=(EditText) findViewById(R.id.editText3);

        String s1=e1.getText().toString();
        String s2=e2.getText().toString();

        a=Integer.parseInt(s1);
        b=Integer.parseInt(s2);

        c=a*b;

        String s3=Integer.toString(c);

        e3.setText(s3);

    }
    public void Div(View v)
    {
        e1=(EditText) findViewById(R.id.editText1);
        e2=(EditText) findViewById(R.id.editText2);
        e3=(EditText) findViewById(R.id.editText3);

```

```

        String s1=e1.getText().toString();
        String s2=e2.getText().toString();

        a=Integer.parseInt(s1);
        b=Integer.parseInt(s2);

        c=a/b;

        String s3=Integer.toString(c);

        e3.setText(s3);

    }
}

```

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"
    android:background="#0000FF">

    <RelativeLayout
        android:id="@+id/relativeLayout1"
        android:layout_width="match_parent"
        android:layout_height="match_parent" >

        <TextView
            android:id="@+id/textView1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentTop="true"
            android:layout_marginTop="34dp"
            android:text="ENTER 1st NUMBER"
            android:textAppearance="?android:attr/textAppearanceLarge" />

        <EditText
            android:id="@+id/editText1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentRight="true"
            android:layout_alignTop="@+id/textView1"
            android:layout_marginRight="16dp"
            android:layout_toRightOf="@+id/textView1"
            android:inputType="number" />

        <TextView
            android:id="@+id/textView2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentLeft="true"
            android:layout_below="@+id/editText1"
            android:layout_marginTop="35dp"
            android:text="ENTER 2nd NUMBER"
            android:textAppearance="?android:attr/textAppearanceLarge" />
    </RelativeLayout>
</LinearLayout>

```

```

<EditText
    android:id="@+id/editText2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignBaseline="@+id/textView2"
    android:layout_alignBottom="@+id/textView2"
    android:layout_alignRight="@+id/editText1"
    android:layout_toRightOf="@+id/textView2"
    android:inputType="number" >

    <requestFocus />
</EditText>

<TextView
    android:id="@+id/textView3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignRight="@+id/textView1"
    android:layout_below="@+id/editText2"
    android:layout_marginTop="42dp"
    android:text="RESULT"
    android:textAppearance="?android:attr/textAppearanceLarge" />

<EditText
    android:id="@+id/editText3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignBaseline="@+id/textView3"
    android:layout_alignBottom="@+id/textView3"
    android:layout_alignRight="@+id/editText2"
    android:layout_toRightOf="@+id/textView2"
    android:inputType="number" />

<Button
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/editText3"
    android:layout_marginRight="21dp"
    android:layout_marginTop="35dp"
    android:layout_toLeftOf="@+id/textView3"
    android:text="ADD"
    android:onClick="Add" />

<Button
    android:id="@+id/button2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignBaseline="@+id/button1"
    android:layout_alignBottom="@+id/button1"
    android:layout_toRightOf="@+id/textView2"
    android:text="SUB"
    android:onClick="Sub" />

<Button
    android:id="@+id/button3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignRight="@+id/button1"
    android:layout_below="@+id/button1"
    android:layout_marginTop="47dp"

```

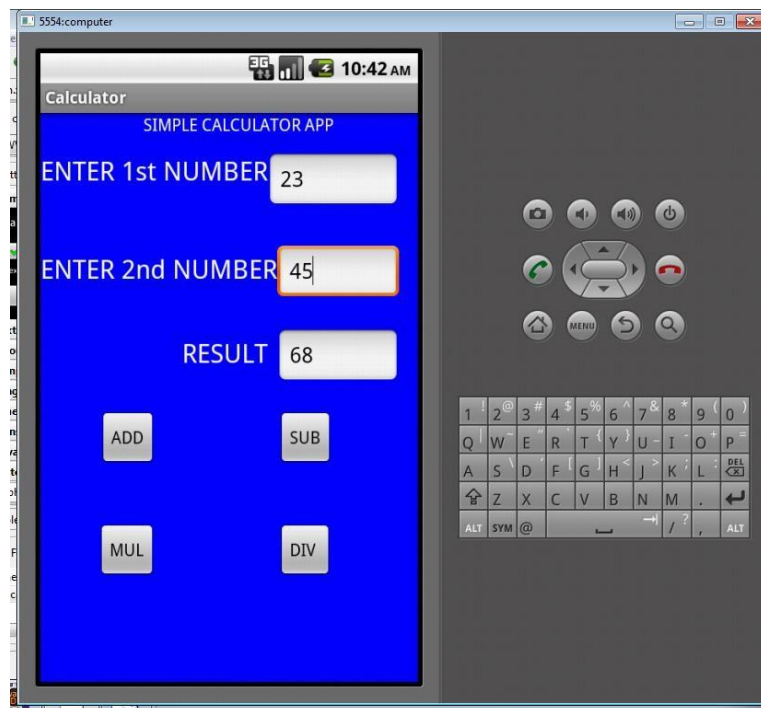
```
android:text="MUL"  
android:onClick="Mul" />
```

```
<Button  
    android:id="@+id/button4"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_alignBaseline="@+id/button3"  
    android:layout_alignBottom="@+id/button3"  
    android:layout_toRightOf="@+id/textView2"  
    android:text="DIV "  
    android:onClick="Div" />
```

```
<TextView  
    android:id="@+id/textView4"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_alignParentTop="true"  
    android:layout_alignRight="@+id/button2"  
    android:text="SIMPLE CALCULATOR APP"  
    android:textColor="#FFFFFF" />
```

```
</RelativeLayout>
```

```
</LinearLayout>
```



Result

Thus successfully running mobile application for simple calculator using text view, edit view and button .

EX NO : 04

AUTO COMPLETE TEXT

AIM

To demonstrate android application for auto complete text view

Procedure

1. Open eclipse or android studio and select new android project
2. Give project name and select next
3. Choose the android version. Choose the lowest android version(Android 2.2) and click finish
4. Go to package explorer in the left hand side. Select our project.
5. Go to res folder and select layout. Double click the main.xml file. Go to text field pick auto complete text view then paste on layouts.
6. Now select mainactivity.java file and to create java class file write program for autocomplete text view.
7. Run the application to launch Android emulator and verify the result of the changes done in the application.

AutocompletetextActivity.java

```
package your.autocompletetext.namespace;

import android.app.Activity;
import android.os.Bundle;
import android.graphics.Color;
import android.widget.ArrayAdapter;
import android.widget.AutoCompleteTextView;

public class AutocompletetextActivity extends Activity {
    /** Called when the activity is first created. */
    String[] language
    ={"C", "C++", "C#", "Java", ".NET", "iPhone", "Android", "ASP.NET", "PHP"};
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        ArrayAdapter<String> adapter = new ArrayAdapter<String>
        (this, android.R.layout.select_dialog_item, language);

        AutoCompleteTextView actv =
        (AutoCompleteTextView) findViewById(R.id.autoCompleteTextView1);
        actv.setThreshold(1);
        actv.setAdapter(adapter);
        actv.setTextColor(Color.RED);
    }
}
```

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"
    android:background="#F0000F" >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="AUTOCOMPLETE TEXT VIEW"
        android:textStyle="bold"
        android:textAppearance="?android:attr/textAppearanceLarge" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="favourite computer programming language"
        android:textAppearance="?android:attr/textAppearanceMedium" />

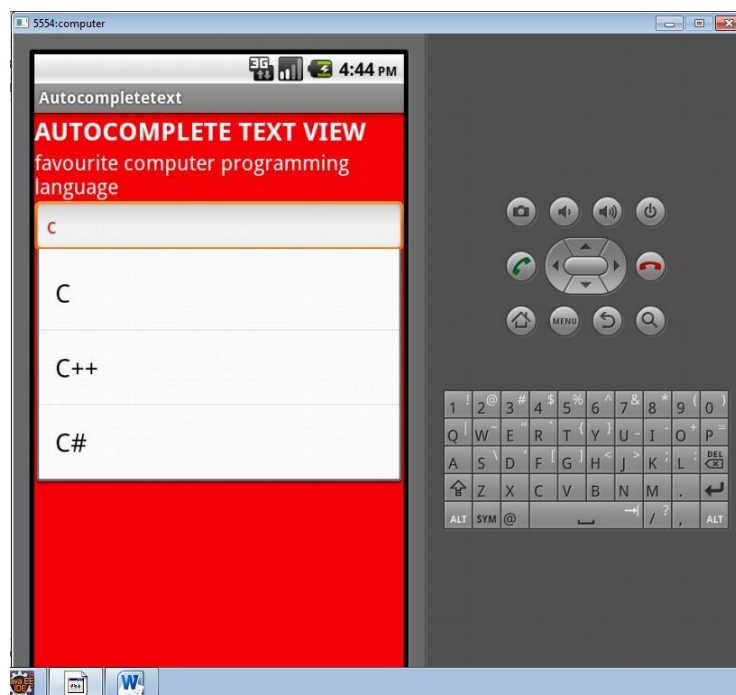
    <AutoCompleteTextView
        android:id="@+id/autoCompleteTextView1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="" >

        <requestFocus />

    </AutoCompleteTextView>

</LinearLayout>

```



Result

Thus successfully demonstrate android application for auto complete text view.

EX NO : 05

LISTVIEW

AIM

To demonstrate android application for list view .

Procedure

1. Open eclipse or android studio and select new android project
2. Give project name and select next
3. Choose the android version. Choose the lowest android version(Android 2.2) and click finish
4. Go to package explorer in the left hand side. Select our project.
5. Go to res folder and select layout. Double click the main.xml file. Go to composite choose list view label then paste on layouts. And create another xml file like listview.xml. go to listview.xml file choose text view paste on layout.
6. Now select mainactivity.java file and to create java class file write program for list view.
7. Run the application to launch Android emulator and verify the result of the changes done in the application.

listviewActivity.java

```
package your.listview.namespace;

import android.app.Activity;
import android.os.Bundle;
import android.view.Menu;
import android.widget.AdapterView;
import android.widget.ListView;

public class ListviewActivity extends Activity {
    String[] mobileArray =
{"Android", "iPhone", "WindowsMobile", "Blackberry",
    "WebOS", "Ubuntu", "Windows7", "Max OS X"};

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        ArrayAdapter adapter = new ArrayAdapter<String>(this,
            R.layout.listview, mobileArray);
```

```

        ListView listView = (ListView) findViewById(R.id.mobilelist);
        listView.setAdapter(adapter);
    }
}
}

Main.xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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:orientation="vertical" >

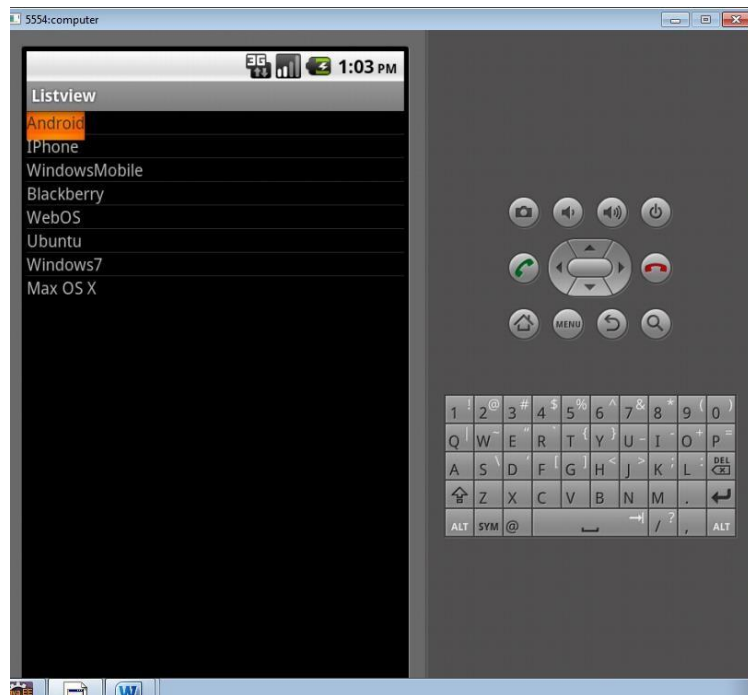
    <ListView
        android:id="@+id/mobilelist"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" >

    </ListView>

</LinearLayout>

Listview.xml
<?xml version="1.0" encoding="utf-8"?>
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="TextView" />

```



Result

Thus successfully demonstrate android application for list view.

EX NO : 06

ALERT DIALOG BOX

AIM

Write a program to demonstrate android application for alert dialog box

Procedure

1. Open eclipse or android studio and select new android project
2. Give project name and select next
3. Choose the android version. Choose the lowest android version(Android 2.2) and click finish
4. Go to package explorer in the left hand side. Select our project.
5. Go to res folder and select layout. Double click the main.xml file. Go to widgets choose button then paste on layouts.
6. Now select mainactivity.java file and to create java class file write program for alert dialog box
 - . To make an object of AlertDialogBuilder which an inner class of AlertDialog. set the positive (yes) or negative (no) button using the object of the AlertDialogBuilder class.
7. Run the application to launch Android emulator and verify the result of the changes done in the application.

Alertdialogboxactivity.Java

```
package your.Alertdialogbox.namespace;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.os.Bundle;
import android.view.View;
import android.widget.Toast;

public class AlertdialogboxActivity extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }

    public void open(View view){
        AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder(this);
        alertDialogBuilder.setMessage("Are you sure,You wanted to make decision");
        alertDialogBuilder.setMessage("YOU WANT EXIT THIS APP");
        alertDialogBuilder.setNegativeButton("YES",new DialogInterface.OnClickListener() {

            public void onClick(DialogInterface dialog, int which) {
```

```

        finish();
    }
});

AlertDialog alertDialog = alertDialogBuilder.create();
alertDialog.show();
}
}

```

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"
    android:background="@drawable/whatsapp" >

    <RelativeLayout
        android:id="@+id/relativeLayout1"
        android:layout_width="match_parent"
        android:layout_height="match_parent" >

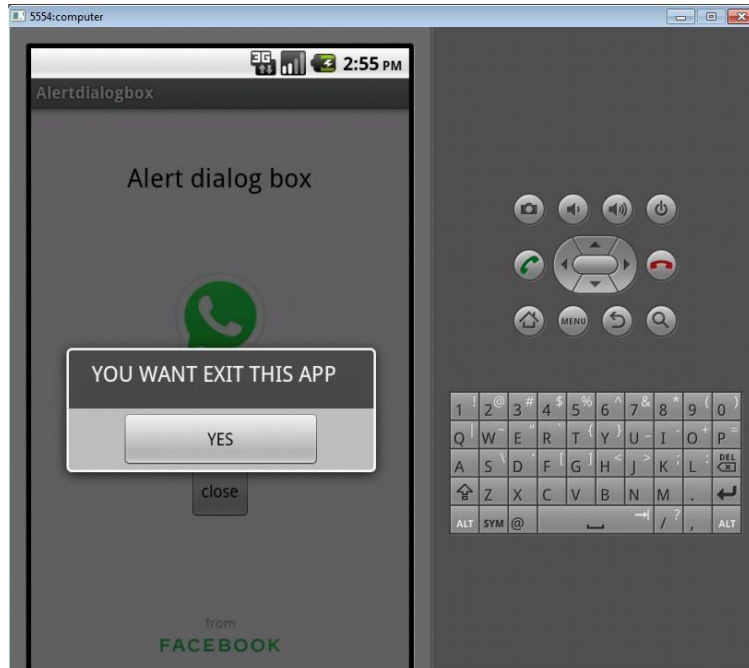
        <Button
            android:id="@+id/button1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentBottom="true"
            android:layout_centerHorizontal="true"
            android:layout_marginBottom="132dp"
            android:onClick="open"
            android:text="close" />

        <TextView
            android:id="@+id/textView1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentTop="true"
            android:layout_centerHorizontal="true"
            android:layout_marginTop="44dp"
            android:text="Alert dialog box"
            android:textColor="#000000"
            android:textAppearance="?android:attr/textAppearanceLarge" />

    </RelativeLayout>

</LinearLayout>

```



Result

Thus successfully demonstrate android application for alert dialog box .

EX NO : 07

PHOTO GALLERY

AIM

Write a program to demonstrate android application for Photo Gallery.

Procedure

1. Open eclipse or android studio and select new android project
2. Give project name and select next
3. Choose the android version. Choose the lowest android version(Android 2.2)and comma and click finish
4. Go to package explorer in the left hand side. Select our project.
5. Go to res folder and select layout. Double click the main.xml file. Go to Image&media choose gallery view and image view then paste on layouts.
6. Go to res folder and select drawable folder .save some picture to drawable folder
7. Now select mainactivity.java file and to create java class file write program for gallery view .
8. Run the application to launch Android emulator and verify the result of the changes done in the application.

Gallery viewActivity.java

```
package your.gallery.namespace;

import android.app.Activity;
import android.content.Context;
import android.os.Bundle;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.BaseAdapter;
import android.widget.Gallery;
import android.widget.ImageView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ImageView.ScaleType;

public class GalleryActivity extends Activity {

    Gallery Imagegallery;
    ImageView imgGalleryImage;
    Integer[] GalleryImagesList =
    {
        R.drawable.one,
        R.drawable.two,
        R.drawable.three,
        R.drawable.four,
        R.drawable.five,
        R.drawable.six,
        R.drawable.seven
    };

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

        imgGalleryImage = (ImageView)findViewById(R.id.imageView1);
        imgGalleryImage.setImageResource(R.drawable.one);

        Imagegallery = (Gallery)findViewById(R.id.gallery);
        Imagegallery.setAdapter(new ImageAdapter(this));
        Imagegallery.setOnItemClickListener(new OnItemClickListener()
        {
            @Override
            public void onItemClick(AdapterView<?> parent, View view,
            int position, long id)
            {
                imgGalleryImage.setImageResource(GalleryImagesList[position]);
            }
        });
    }

    private class ImageAdapter extends BaseAdapter
    {
        Context context;
        public ImageAdapter(Context context)
        {
            this.context = context;
        }
    }
}
```

```

@Override
public int getCount()
{
    return GalleryImagesList.length;
}

@Override
public Object getItem(int position)
{
    return GalleryImagesList[position];
}

@Override
public long getItemId(int position)
{
    return position;
}

@Override
public View getView(int position, View convertView, ViewGroup parent)
{
    ImageView imageView = new ImageView(this.context);
    imageView.setImageResource(GalleryImagesList[position]);
    imageView.setLayoutParams(new Gallery.LayoutParams(150, 200));
    imageView.setScaleType(ImageView.ScaleType.FIT_XY);

    return imageView;
}
}
}

```

Main.xml

```

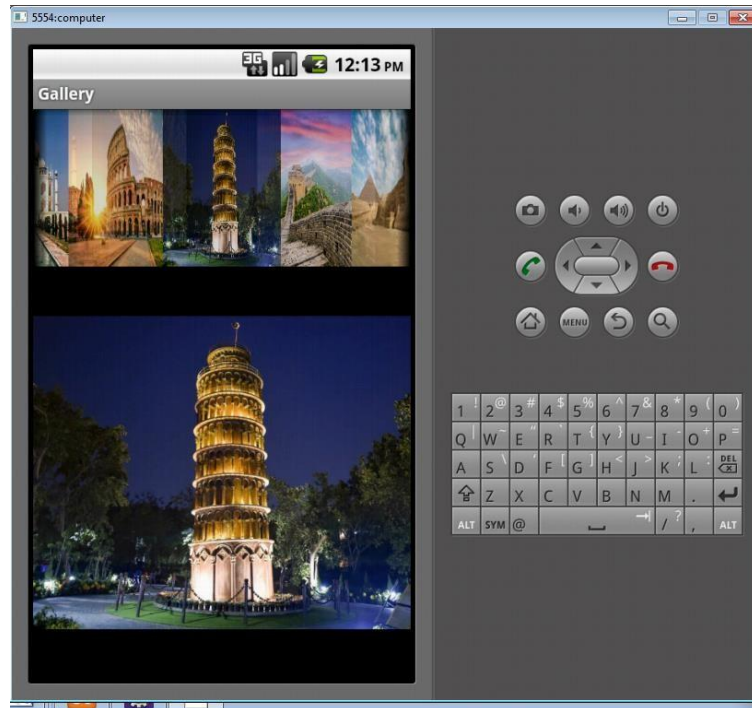
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <Gallery
        android:id="@+id/gallery"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

    <ImageView
        android:id="@+id/imageView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/ic_launcher" />

</LinearLayout>

```



Result

Thus successfully demonstrate for android gallery application.

EX NO : 08

AIM

Write a program to demonstrate android application for date and time picker.

Procedure

1. Open eclipse or android studio and select new android project
2. Give project name and select next
3. Choose the android version. Choose the lowest android version(Android 2.2) and comma and click finish
4. Go to package explorer in the left hand side. Select our project.
5. Go to res folder and select layout. Double click the main.xml file. Go to date & time choose date and time then paste on layouts.
6. Now select mainactivity.java file and to create java class file write program for date & time picker .

7. Run the application to launch Android emulator and verify the result of the changes done in the application.

datetimepickerActivity.java

```
package your.datetimepicker.namespace;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.DatePicker;
import android.widget.TextView;
import android.widget.TimePicker;

public class DatetimepickerActivity extends Activity {
    DatePicker picker;
    Button displayDate;
    TextView textview1;
    TimePicker timepicker;
    TextView textview2;
    Button changetime;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        textview1=(TextView) findViewById(R.id.textView1);
        textview2=(TextView) findViewById(R.id.textView2);
        picker=(DatePicker) findViewById(R.id.datePicker1);
        displayDate=(Button) findViewById(R.id.button1);
        changetime=(Button) findViewById(R.id.button2);

        textview1.setText("Current Date: "+getCurrentDate());

        displayDate.setOnClickListener(new View.OnClickListener(){
            @Override
            public void onClick(View view) {

                textview1.setText("Change Date: "+getCurrentDate());
            }

        });
        timepicker=(TimePicker) findViewById(R.id.timePicker1);
        timepicker.setIs24HourView(true);

        textview2.setText(getCurrentTime());
        changetime.setOnClickListener(new View.OnClickListener(){
            @Override
            public void onClick(View view) {
                textview2.setText(getCurrentTime());
            }
        });
    }

    public String getCurrentDate(){
        StringBuilder builder=new StringBuilder();
        builder.append((picker.getMonth() + 1)+"//month is 0 based");
        builder.append(picker.getDayOfMonth()+"//");
        builder.append(picker.getYear());
    }
}
```

```

        return builder.toString();
    }
    public String getCurrentTime(){
        String currentTime="Current Time:
"+timepicker.getCurrentHour()+":"+timepicker.getCurrentMinute();
        return currentTime;
    }
}

```

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"
    android:background="#F0000F">

    <DatePicker
        android:id="@+id/datePicker1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center" />

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="pick current date"
        android:textAppearance="?android:attr/textAppearanceLarge" />

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="pickdate" />

    <TimePicker
        android:id="@+id/timePicker1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center" />

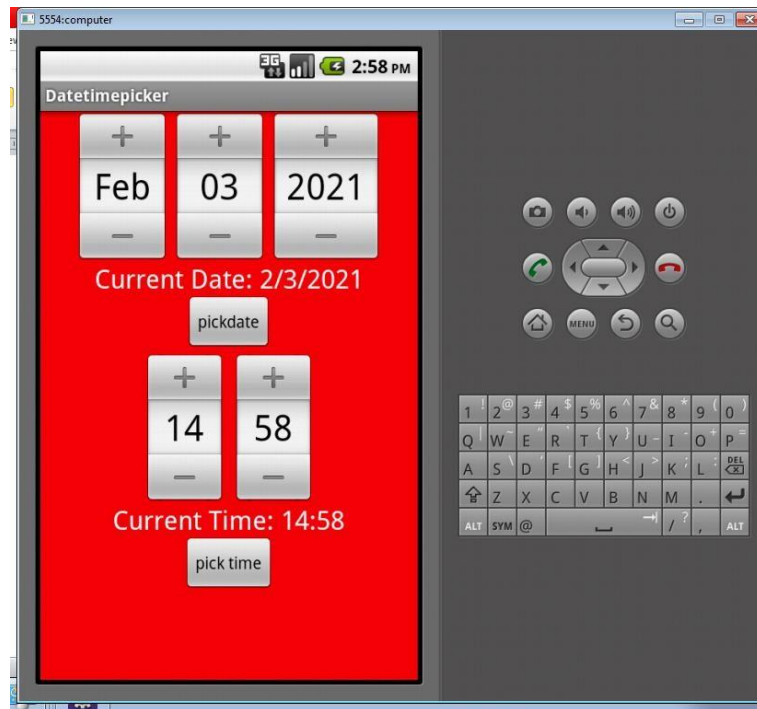
    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:text="pick current time"
        android:textAppearance="?android:attr/textAppearanceLarge" />

    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"

```

```
android:layout_height="wrap_content"  
android:layout_gravity="center"  
android:text="pick time" />
```

```
</LinearLayout>
```



Result

Thus successfully demonstrate for android date and time.

EX NO : 09

OPTIONS MENU CONTEXT MENU

AIM

Write a program to demonstrate android application for. Options Menu
Context Menu

Procedure

1. Open eclipse or android studio and select new android project
2. Give project name and select next
3. Choose the android version. Choose the lowest android version(Android 2.2)and comma and click finish
4. Go to package explorer in the left hand side. Select our project.
5. Go to res folder and select layout. Double click the main.xml file. To design menus layouts.

6 . Go to res folder and create menu folder and menu xml file layout. Double click the menu.xml file. To adding list of items.

7. Now select mainactivity.java file and to create java class file write program for option menu and context menu .

8. Run the application to launch Android emulator and verify the result of the changes done in the application.

MENUACTIVITY.JAVA

```
package
```

```
your.menu.namespace;
```

```
import android.app.Activity;
```

```
import android.os.Bundle;
```

```
import
```

```
android.view.ContextMenu;
```

```
import android.view.Menu;
```

```
import
```

```
android.view.MenuItem;
```

```
import android.view.View;
```

```
import android.widget.Button;
```

```
import android.widget.Toast;
```

```
public class MenuActivity extends Activity {
```

```
    protected void onCreate(Bundle
```

```
        savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.main);
```

```
        Button btn = (Button) findViewById(R.id.button1);
```

```
        registerForContextMenu(btn);
```

```
    }
```

```
    @Override
```

```
    public void onCreateContextMenu(ContextMenu menu,
```

```
View v, ContextMenu.ContextMenuInfo menuInfo) {  
    super.onCreateContextMenu(menu, v,  
        menuInfo); menu.setTitle("Context  
Menu");
```

```

        menu.add(0, v.getId(), 0, "Upload");
        menu.add(0, v.getId(), 0, "Search");
        menu.add(0, v.getId(), 0, "Share");
        menu.add(0, v.getId(), 0, "Bookmark");
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        Toast.makeText(this, "Selected Item: " +item.getTitle(),
            Toast.LENGTH_SHORT).show(); return true;
    }

    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.option_menu,
            menu); return true;
    }

    public boolean onOptionsItemSelected(MenuItem item) {
        Toast.makeText(this, "Selected Item: " +item.getTitle(),
            Toast.LENGTH_SHORT).show(); switch (item.getItemId()) {
            case R.id.share_item:
                // do your
                code return
                true;
            case R.id.upload_item:
                // do your
                code return
                true;
            case R.id.search_item:
                // do your
                code return

```

true;

default:

```
        return super.onOptionsItemSelected(item);
    }
}
}
```

Option menu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:compat="http://schemas.android.com/apk/res-auto">
    <item android:id="@+id/mail"
        android:icon="@drawable/mail"
        android:title="@string/mail" />
    <item android:id="@+id/upload"
        android:icon="@drawable/upload"
        android:title="@string/upload"
        android:showAsAction="ifRoom" />
    <item android:id="@+id/share"
        android:icon="@drawable/share"
        android:title="@string/share" />
</menu>
```

Main.xml

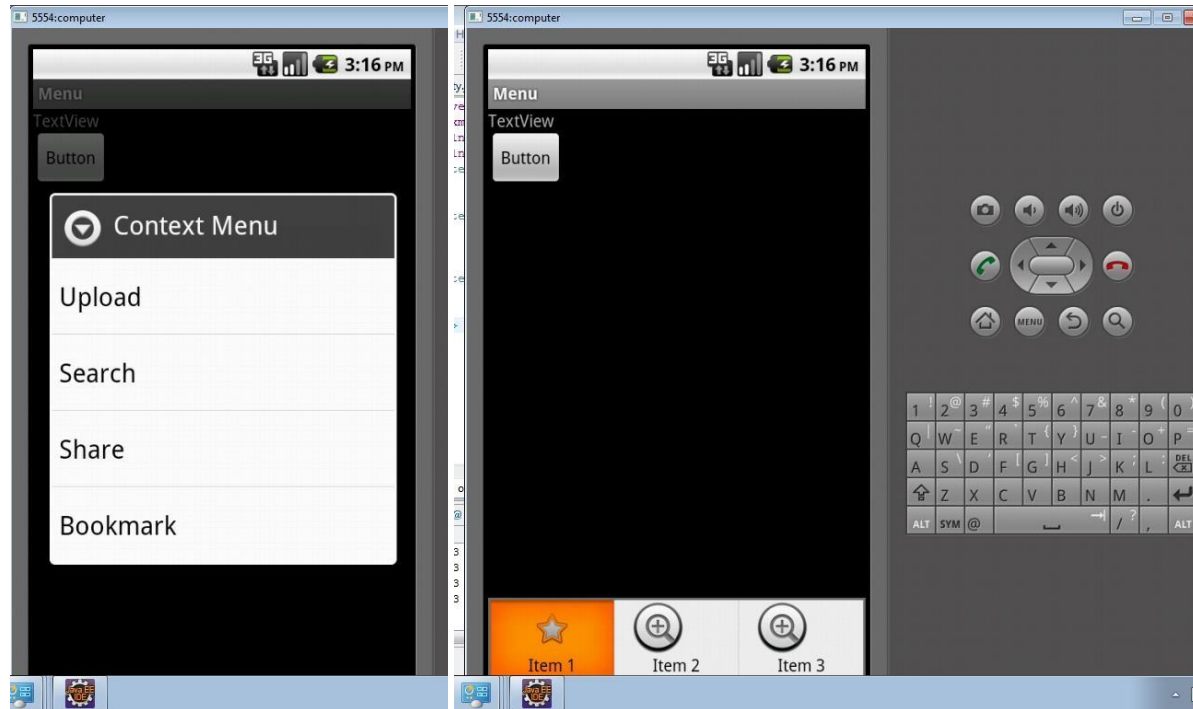
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:compat="http://schemas.android.com/apk/res-auto"
    tools:context=".menu"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"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

    <RelativeLayout
        android:id="@+id/relativeLayout1"
        android:layout_width="match_parent"
        android:layout_height="match_parent" >

        <Button
            android:id="@+id/button1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentTop="true"
            android:layout_centerHorizontal="true"
            android:layout_marginTop="197dp"
            android:text="Context Menu" />

    </RelativeLayout>

</LinearLayout>
```

Result

Thus successfully demonstrate for android application for options menu context menu .

EX NO : 10

FIXED DIAL CALL

AIM

Develop an android application for fixed dialing call

Procedure

1. Open eclipse or android studio and select new android project
2. Give project name and select next
3. Choose the android version. Choose the lowest android version(Android 2.2)and comma and click finish
4. Go to package explorer in the left hand side. Select our project.

5. Go to res folder and select layout. Double click the main.xml file. To pick single button only on layouts.

6. Now select mainactivity.java file and to create java class file write program fixed dial call and call phone dialer to make a call.

7. Go to androidmanifest.xml file and add uses permission **for <uses-permission android:name="android.permission.CALL_PHONE"/>**

8. Run the application to launch Android emulator and verify the result of the changes done in the application.

fixeddailcallActivity.java

```
package your.fixeddailcall.namespace;

import android.app.Activity;
import android.os.Bundle;
import android.content.Intent;
import android.net.Uri;
import android.view.View;
import android.widget.Button;

public class FixeddailcallActivity extends Activity {
    private Button button;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        button = (Button) findViewById(R.id.button1);

        button.setOnClickListener(new View.OnClickListener() {
            public void onClick(View arg0) {
                Intent callIntent = new Intent(Intent.ACTION_CALL);
                callIntent.setData(Uri.parse("tel:108"));

                startActivity(callIntent);
            }
        });
    }
}
```

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"
    android:background="#0FF000">

    <RelativeLayout
        android:id="@+id/relativeLayout1"
        android:layout_width="match_parent"
        android:layout_height="match_parent">
```

```

        <Button
            android:id="@+id/button1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentTop="true"
            android:layout_centerHorizontal="true"
            android:layout_marginTop="199dp"
            android:text="HELP LINE" />

    </RelativeLayout>

</LinearLayout>

```

AndroidManifest.xml

```

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

```

Result

Thus successfully demonstrate for android application fixed dial call and call phone dialer to make a call .

EX NO : 11

sendSMS

AIM

Develop an android application for fixed dialing call

Procedure

1. Open eclipse or android studio and select new android project
2. Give project name and select next
3. Choose the android version. Choose the lowest android version(Android 2.2) and click finish
4. Go to package explorer in the left hand side. Select our project.
5. Go to res folder and select layout. Double click the main.xml file. To pick single edittext and button paste on layouts.
6. Now select mainactivity.java file and to create java class file write program send sms.
7. Go to androidmanifest.xml file and add uses permission **for <uses-permission android:name="android.permission.SEND_SMS"/>**
8. Run the application to launch Android emulator and verify the result of the changes done in the application.

sendsmsActivity.java

```

package your.sendsms.namespace;

import android.app.Activity;
import android.os.Bundle;
import android.app.PendingIntent;
import android.content.Intent;
import android.telephony.SmsManager;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class SendsmsActivity extends Activity {

    EditText mobileno,message;
    Button sendsms;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        mobileno=(EditText)findViewById(R.id.editText1);
        message=(EditText)findViewById(R.id.editText2);
        sendsms=(Button)findViewById(R.id.button1);

        //Performing action on button click
        sendsms.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View arg0) {
                String no=mobileno.getText().toString();
                String msg=message.getText().toString();

                //Getting intent and PendingIntent instance
                Intent intent=new
Intent(getApplicationContext(),SendsmsActivity.class);
                PendingIntent
pi=PendingIntent.getActivity(getApplicationContext(), 0, intent,0);

                //Get the SmsManager instance and call the sendTextMessage
method to send message
                SmsManager sms=SmsManager.getDefault();
                sms.sendTextMessage(no, null, msg, pi,null);

                Toast.makeText(getApplicationContext(), "Message Sent
successfully!",
                    Toast.LENGTH_LONG).show();
            }
        });
    }
}

```

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"
    android:background="#FF3333" >

```

```

<RelativeLayout
    android:id="@+id/relativeLayout1"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:layout_marginTop="17dp"
        android:text="To"
        android:textAppearance="?android:attr/textAppearanceLarge" />

    <EditText
        android:id="@+id/editText1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentRight="true"
        android:layout_alignTop="@+id/textView1"
        android:layout_marginLeft="41dp"
        android:layout_toRightOf="@+id/textView1"
        android:inputType="phone" >

        <requestFocus />
    </EditText>

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_below="@+id/editText1"
        android:layout_marginTop="26dp"
        android:text="Message"
        android:textAppearance="?android:attr/textAppearanceLarge" />

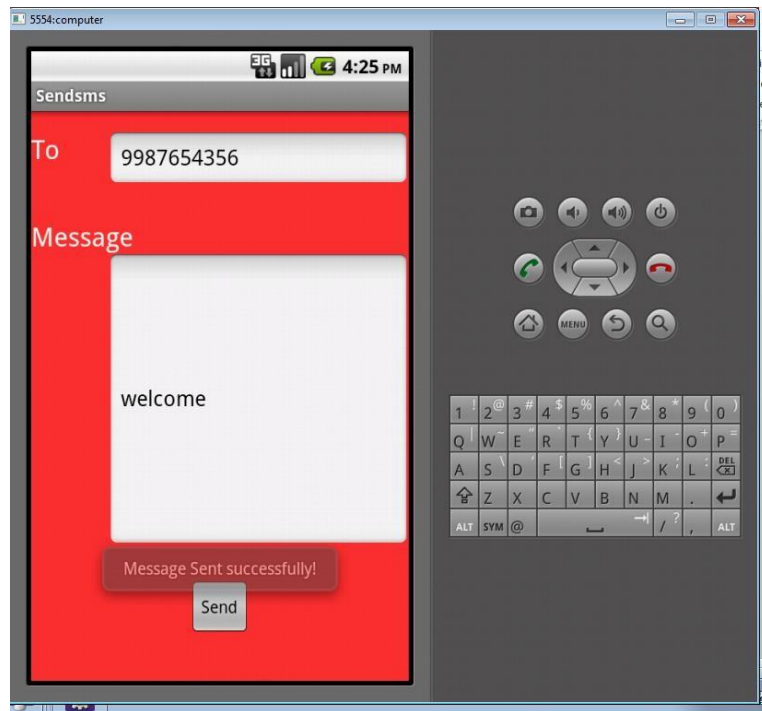
    <EditText
        android:id="@+id/editText2"
        android:layout_width="wrap_content"
        android:layout_height="250dp"
        android:layout_alignLeft="@+id/editText1"
        android:layout_alignParentRight="true"
        android:layout_below="@+id/textView2"
        android:inputType="textMultiLine" />

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/editText2"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="27dp"
        android:text="Send" />

</RelativeLayout>

```

</LinearLayout>



Result

Thus successfully develop for android application sending sms.

EX NO : 12

view and edit contact

AIM

Develop an android application for fixed dialing call

Procedure

1. Open eclipse or android studio and select new android project
2. Give project name and select next
3. Choose the android version. Choose the lowest android version(Android 2.2) and click finish
4. Go to package explorer in the left hand side. Select our project.
5. Go to res folder and select layout. Double click the main.xml file. To pick single edittext and button paste on layouts.
6. Now select mainactivity.java file and to create java class file write program send sms.

7. Go to androidmanifest.xml file and add uses permission for `<uses-permission android:name="android.permission.WRITE_CONTACTS"/>`

8. Run the application to launch Android emulator and verify the result of the changes done in the application.

Viewandeditcontact.java

```
package your.viewandeditcontact.namespace;

import android.app.Activity;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.provider.ContactsContract;
import android.view.Menu;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

public class VieweditcontactActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        Button insert = (Button) findViewById(R.id.button1);
        insert.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View v) {
                // TODO Auto-generated method stub
                insert();
            }
        });

        public void insert() {
            Intent intent = new Intent(
                ContactsContract.Intents.SHOW_OR_CREATE_CONTACT,
                ContactsContract.Contacts.CONTENT_URI);
            intent.setData(Uri.parse("tel:011-98943043")); //specify your number
            here
            intent.putExtra(ContactsContract.Intents.Insert.COMPANY, "Google");
            intent.putExtra(ContactsContract.Intents.Insert.POSTAL,
                "House Address, Street Name, State/Country");
            startActivity(intent);
            Toast.makeText(this, "Record insert", Toast.LENGTH_SHORT).show();
        }
    }
}
```

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" >

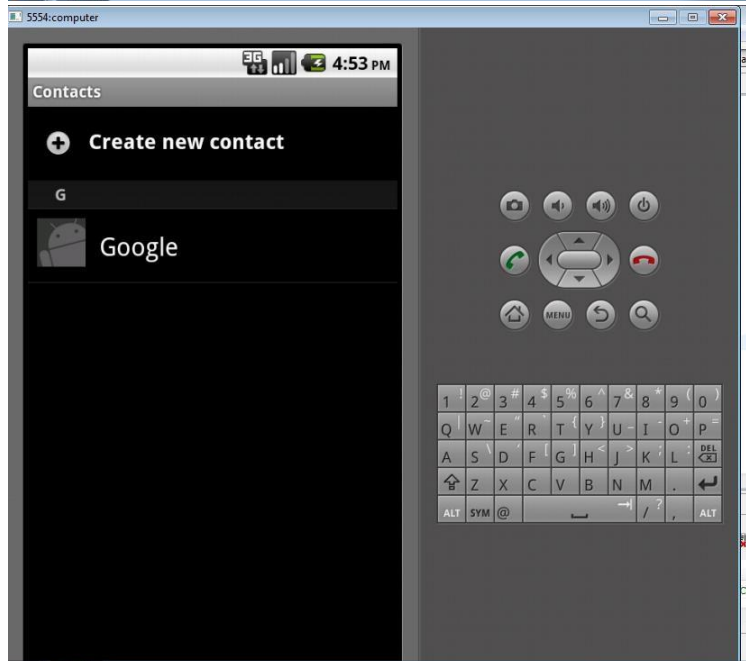
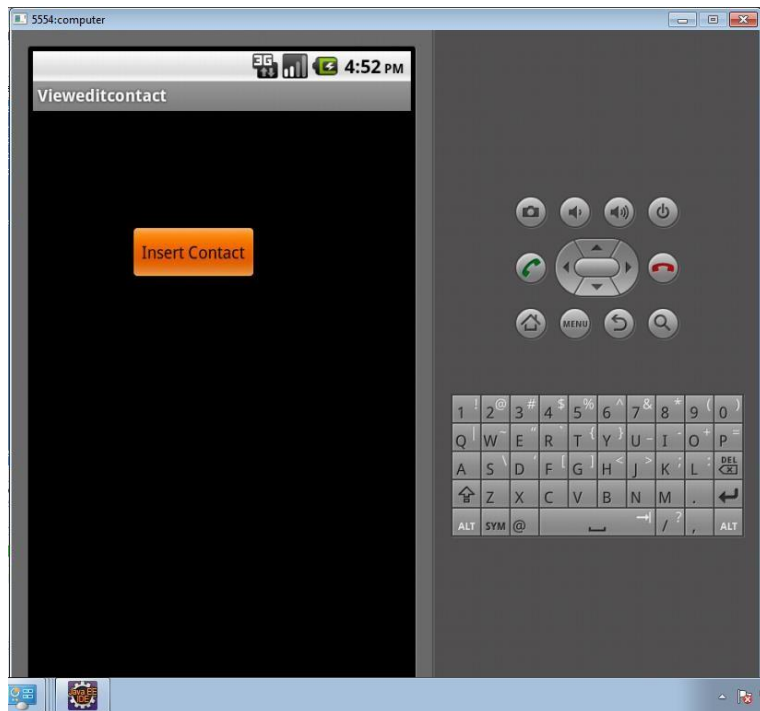
    <RelativeLayout
```

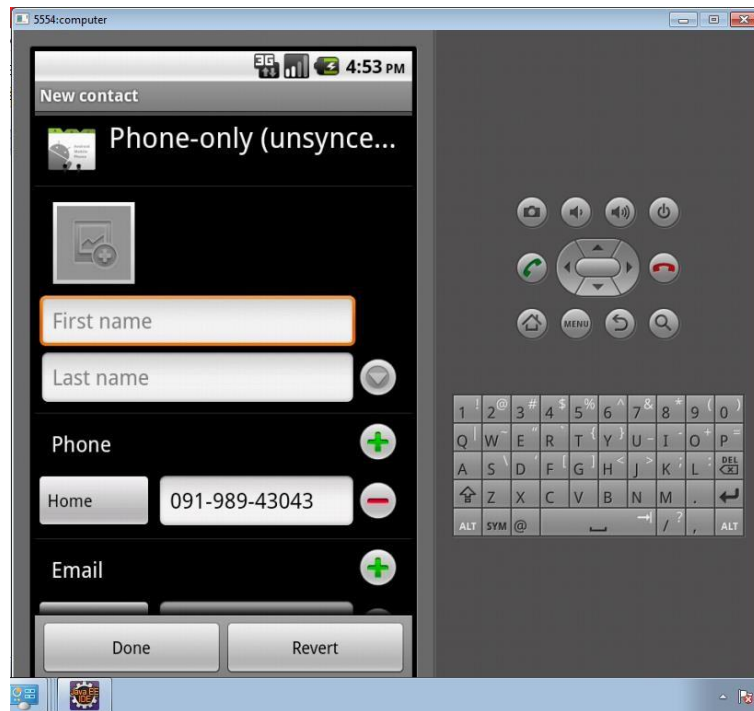
```
android:id="@+id/relativeLayout1"  
android:layout_width="match_parent"  
android:layout_height="match_parent" >
```

```
<Button  
    android:id="@+id/button1"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_alignParentLeft="true"  
    android:layout_alignParentTop="true"  
    android:layout_marginLeft="81dp"  
    android:layout_marginTop="98dp"  
    android:text="Insert Contact" />
```

```
</RelativeLayout>
```

```
</LinearLayout>
```



Result

Thus successfully running for android application view and edit contact .

EX NO : 13

E-MAIL APP

AIM

Write a program for an android application e-mail

Procedure

1. Open eclipse or android studio and select new android project
2. Give project name and select next
3. Choose the android version. Choose the lowest android version(Android 2.2)and comma and click finish
4. Go to package explorer in the left hand side. Select our project.
5. Go to res folder and select layout. Double click the main.xml file. To pick edittextbox and button paste on layouts.
6. Now select mainactivity.java file and to create java class file write program send EMAIL.

7. Go to androidmanifest.xml file and add uses permission for `<uses-permission android:name="android.permission.SEND_EMAIL"/>`

8. Run the application to launch Android emulator and verify the result of the changes done in the application.

sendemailActivity.java

```
package your.sendemail.namespace;

import android.app.Activity;
import android.os.Bundle;
import android.content.Intent;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;

public class SendemailActivity extends Activity {

    Button buttonSend;
    EditText textTo;
    EditText textSubject;
    EditText textMessage;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        buttonSend = (Button) findViewById(R.id.button1);
        textTo = (EditText) findViewById(R.id.editText1);
        textSubject = (EditText) findViewById(R.id.editText2);
        textMessage = (EditText) findViewById(R.id.editText3);

        buttonSend.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View v) {

                String to = textTo.getText().toString();
                String subject = textSubject.getText().toString();
                String message = textMessage.getText().toString();

                Intent email = new Intent(Intent.ACTION_SEND);
                email.putExtra(Intent.EXTRA_EMAIL, new String[]{ to});
                email.putExtra(Intent.EXTRA_SUBJECT, subject);
                email.putExtra(Intent.EXTRA_TEXT, message);
                email.setType("message/rfc822");
                startActivity(Intent.createChooser(email, "Choose an Email client
:"));
            }
        });
    }
}
```

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:background="#FF33E6"
android:orientation="vertical" >
```

```
<RelativeLayout
```

```
    android:id="@+id/relativeLayout1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_weight="1.11" >
```

```
<TextView
```

```
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:text="To"
    android:textAppearance="?android:attr/textAppearanceMedium" />
```

```
<EditText
```

```
    android:id="@+id/editText1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentRight="true"
    android:layout_below="@+id/textView1"
    android:layout_toRightOf="@+id/textView1"
    android:inputType="textEmailAddress" >
```

```
<requestFocus />
```

```
</EditText>
```

```
<TextView
```

```
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_below="@+id/editText1"
    android:text="Subject"
    android:textAppearance="?android:attr/textAppearanceMedium" />
```

```
<EditText
```

```
    android:id="@+id/editText2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentRight="true"
    android:layout_below="@+id/textView2"
    android:layout_toRightOf="@+id/textView1" />
```

```
<TextView
```

```
    android:id="@+id/textView3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_below="@+id/editText2"
    android:text="Message"
    android:textAppearance="?android:attr/textAppearanceMedium" />
```

```
<EditText
```

```
    android:id="@+id/editText3"
    android:layout_width="wrap_content"
```

```

        android:layout_height="250dp"
        android:layout_alignLeft="@+id/editText2"
        android:layout_alignParentRight="true"
        android:layout_below="@+id/textView3"
        android:layout_marginTop="16dp"
        android:inputType="textMultiLine" />

```

```

<Button
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_centerHorizontal="true"
    android:text="send"/>

```

```

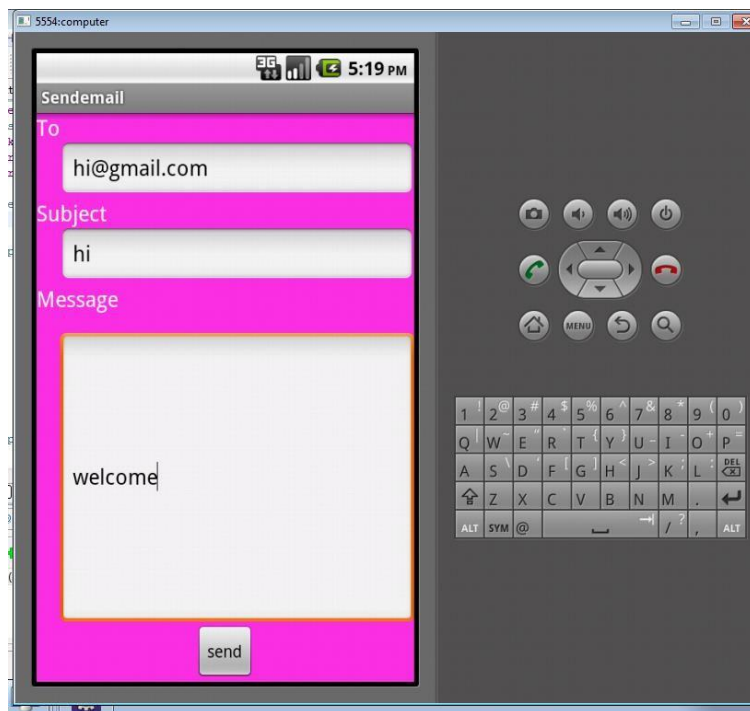
</RelativeLayout>

```

```

</LinearLayout>

```



RESULT

Thus successfully running for android application send E-Mail ..

EX NO : 14

SERVICES

AIM

Write a program for an android application e-mail

Procedure

1. Open eclipse or android studio and select new android project
2. Give project name and select next
3. Choose the android version. Choose the lowest android version(Android 2.2) and click finish
4. Go to package explorer in the left hand side. Select our project.
5. Go to res folder and select layout. Double click the main.xml file. To pick two button paste on layouts.
6. create raw folder on res and paste some mp3 music on raw folder .
7. Now select ServiceActivity.java file and to create java class file write program for service. A create another java class file for MyService.java.
8. Go to androidmanifest.xml file and add service on music player activity like


```
<service android:name=".MyService"
  android:enabled="true" />
```
9. Run the application to launch Android emulator and verify the result of the changes done in the application.

ServiceActivity.java

```
package your.service.namespace;

import android.app.Activity;
import android.os.Bundle;
import android.content.Intent;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
public class ServiceActivity extends Activity implements OnClickListener {
    Button buttonStart, buttonStop, buttonNext;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        buttonStart = (Button) findViewById(R.id.button1);
        buttonStop = (Button) findViewById(R.id.button2);

        buttonStart.setOnClickListener(this);
        buttonStop.setOnClickListener(this);
    }
    public void onClick(View src) {
        switch (src.getId()) {
            case R.id.button1:
                startService(new Intent(this, MyService.class));
                break;
            case R.id.button2:
                stopService(new Intent(this, MyService.class));
        }
    }
}
```

```

        break;
    }
}

```

Myservice.java

```

package your.servicee.namespace;

import android.app.Service;
import android.content.Intent;
import android.media.MediaPlayer;
import android.os.IBinder;
import android.widget.Toast;
public class MyService extends Service {
    MediaPlayer myPlayer;
    @Override
    public IBinder onBind(Intent intent) {
        return null;
    }
    @Override
    public void onCreate() {
        Toast.makeText(this, "Service Created", Toast.LENGTH_LONG).show();

        myPlayer = MediaPlayer.create(this, R.raw.aaa);
        myPlayer.setLooping(false); // Set looping
    }
    @Override
    public void onStart(Intent intent, int startid) {
        Toast.makeText(this, "Service Started", Toast.LENGTH_LONG).show();
        myPlayer.start();
    }
    @Override
    public void onDestroy() {
        Toast.makeText(this, "Service Stopped", Toast.LENGTH_LONG).show();
        myPlayer.stop();
    }
}

```

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"
    android:background="#ACFA58">

    <RelativeLayout
        android:id="@+id/relativeLayout1"
        android:layout_width="match_parent"
        android:layout_height="match_parent" >

        <Button
            android:id="@+id/button1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentTop="true"
            android:layout_centerHorizontal="true"
            android:layout_marginTop="141dp"
            android:text="service start" />

```

```

<Button
    android:id="@+id/button2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/button1"
    android:layout_below="@+id/button1"
    android:layout_marginTop="50dp"
    android:text="service stop" />

```

```

</RelativeLayout>

```

```

</LinearLayout>

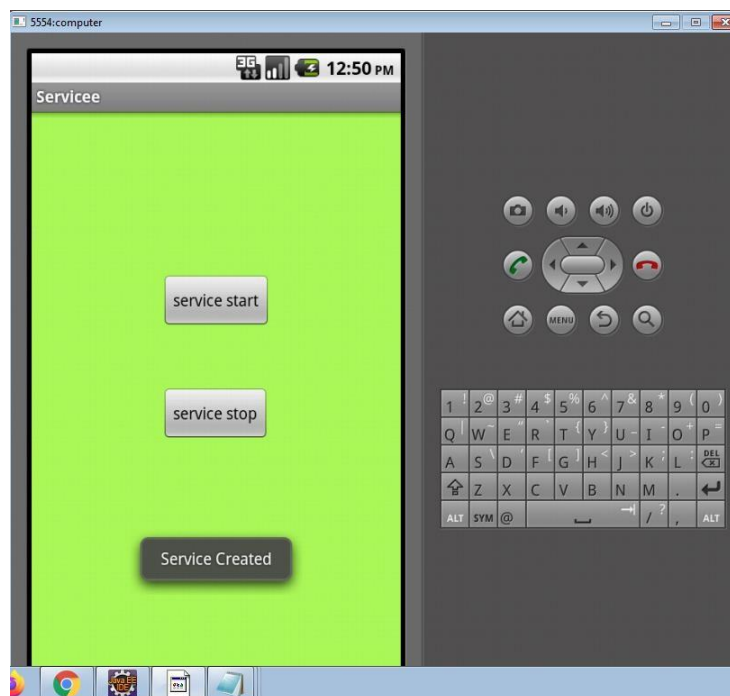
```

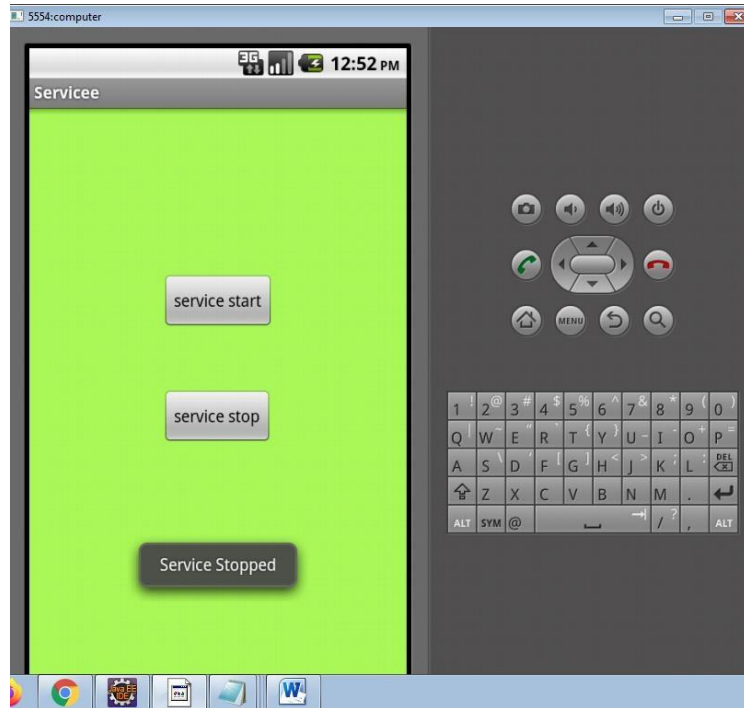
Androidmanifest.xml

```

<service
    android:name=".MyService"
    android:enabled="true" />

```





Result

Thus successfully demonstrate for android application service.

EX NO : 15

webview

AIM

Write a program to demonstrate web view to display web site an android

Procedure

1. Open eclipse or android studio and select new android project
2. Give project name and select next
3. Choose the android version. Choose the lowest android version(Android 2.2) and comma and click finish
4. Go to package explorer in the left hand side. Select our project.
5. Go to res folder and select layout. Double click the main.xml file. To pick two button paste on layouts.
6. Now select ServicessActivity.java file and to create java class file write program for service. A create another java class file for Myservice.java.

7. Go to androidmanifest.xml file and add uses permission for `<uses-permission android:name="android.permission.INTERNET"/>`

8. Run the application to launch Android emulator and verify the result of the changes done in the application.

webViewActivity.java

```
package your.webview.namespace;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.webkit.WebSettings;
import android.webkit.WebView;
import android.webkit.WebViewClient;
public class WebViewActivity extends Activity {
    /** Called when the activity is first created. */

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        WebView mainWebView = (WebView) findViewById(R.id.webView1);

        WebSettings webSettings = mainWebView.getSettings();
        webSettings.setJavaScriptEnabled(true);
        mainWebView.setWebViewClient(new WebViewClient());

        mainWebView.setScrollBarStyle(View.SCROLLBARS_INSIDE_OVERLAY);
        mainWebView.loadUrl("https://www.facebook.com");
    }
}
```

Main.xml

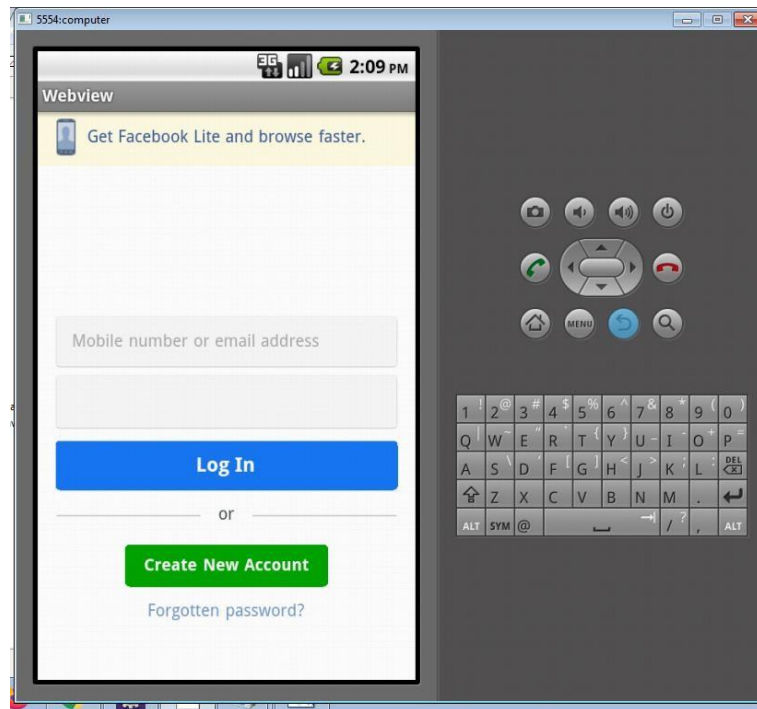
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <WebView
        android:id="@+id/webView1"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

</LinearLayout>
```

Androidmanifest.xml

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



Result

Thus successfully demonstrate for web view to display web site an android.

EX NO : 16

intent class

AIM

Write a program to demonstrate intent class activity

Procedure

1. Open eclipse or android studio and select new android project
2. Give project name and select next
3. Choose the android version. Choose the lowest android version(Android 2.2) and click finish
4. Go to package explorer in the left hand side. Select our project.
5. Go to res folder and select layout. Double click the main.xml file. To pick one button paste on layouts.
6. Go to res folder and create new android xml file name as second.xml. Double click the second.xml file. To pick one button paste on layouts

7. Now select ServicessActivity.java file and to create java class file write program for intent. A create another java class file for second.java.

8.Go to androidmanifest.xml file and add another class activity like as

```
<activity android:name=".second"
    android:label="@string/app_name" />
```

9. Run the application to launch Android emulator and verify the result of the changes done in the application.

IntentActivity.java

```
package your.intent.namespace;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class IntentActivity extends Activity {
    Button nextactivity;
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        {

            nextactivity= (Button)findViewById(R.id.button1);

            nextactivity.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    Intent intent=new Intent(IntentActivity.this,second.class);
                    startActivity(intent);
                }
            });
        }
    }
}
```

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"
    android:background="#FF0F0F">

    <RelativeLayout
        android:id="@+id/relativeLayout1"
        android:layout_width="match_parent"
        android:layout_height="match_parent" >

        <Button
```

```

        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:layout_marginBottom="156dp"
        android:text="go to second activity" />

```

```

    </RelativeLayout>

```

```

</LinearLayout>

```

Second.java

```

package your.intent.namespace;

```

```

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

```

```

public class second extends Activity {
    Button firstactivity;
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.second);
        {
            firstactivity= (Button)findViewById(R.id.button1);

            firstactivity.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                    Intent intent=new Intent(second.this, IntentActivity.class);
                    startActivity(intent);
                }
            });
        }
    }
}

```

Second.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:background="#00FF00" >

    <RelativeLayout
        android:id="@+id/relativeLayout1"
        android:layout_width="match_parent"
        android:layout_height="match_parent" >

        <Button
            android:id="@+id/button1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"

```

```

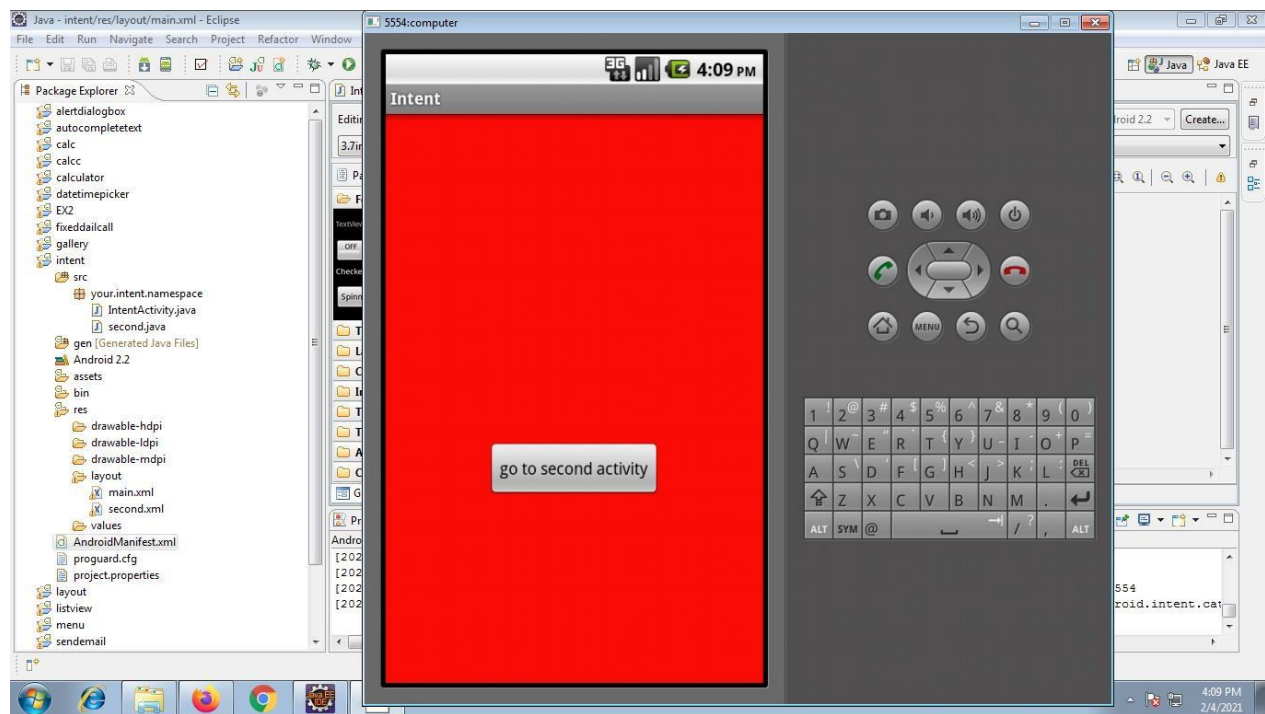
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:layout_marginBottom="153dp"
        android:text="go to FIRST activity" />

</RelativeLayout>

</LinearLayout>

Androidmanifest.xml
<activity
    android:name=".second"
    android:label="@string/app_name" />

```



Result

Thus successfully demonstrate for intent class using one activity to another activity.

EX:17

CREATE TEXT FILE IN A EXTERNAL MEMORY

AIM

Write a program to create text file in external memory

1. Open eclipse or android studio and select new android project
2. Give project name and select next
3. Choose the android version.Choose the lowest android version(Android 2.2) and select next
4. Enter the package name.package name must be two word seprated by comma and click finish
5. Go to package explorer in the left hand side.select our project.
6. Go to res folder and select layout.Double click the main.xml file.Don't change anything in layout.Leave as default.
7. Now select mainactivity.java file and to create java class file to type based on create text file in a external storage application coding.
8. Modify the default content of res/layout/main.xml file to include design for layout create text file in a external storage
9. Go to res/AndroidManifest.xml to allow uses permission for “<Uses permission android:name=”android:permission WRITE_INTERNAL_EXTERNAL”/>”.
10. Run the application to launch Android emulator and verify the result of the changes done in the application.

externalActivity.java

```
package your. externaltext.namespace;
```

```
import android.app.Activity;
import android.os.Bundle;
import android.os.Environment;
import android.text.format.DateFormat; import android.view.View;
import android.widget.Button;
```

```

import android.widget.EditText;
import android.widget.TextView;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
public class externaltextActivity extends Activity
{
    EditText editText;
    Button button;
    TextView textView;
    String h;
    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        editText = (EditText) findViewById(R.id.editText1);
        textView= (TextView) findViewById(R.id.textView1);
        button = (Button) findViewById(R.id.button1);
        button.setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View v) { try {
                h = DateFormat.format("MM-dd-yyyyy-h-mmssaa",
                System.currentTimeMillis()).toString();

```

```

File root = new File(Environment.getExternalStorageDirectory(),
"Notes");

```

```

if (!root.exists()) {
    root.mkdirs(); // this will create folder.

```

```

    }
    File filepath = new File(root, h + ".txt"); // file path to save
    FileWriter writer = new FileWriter(filepath);

```



```

writer.append(editText.getText().toString());
writer.flush(); writer.close();
String m = "File generated with name " + h + ".txt"; textView.setText(m);
} catch (IOException e) { e.printStackTrace();
textView.setText(e.getMessage().toString());
}
}
});
}
}
}

```

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"
    android:background="#FF00F0">
    <RelativeLayout
        android:id="@+id/relativeLayout1"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="center" >
    <EditText
        android:id="@+id/editText1"
        android:layout_width="wrap_content"
        android:layout_height="250dp"
        android:layout_alignParentLeft="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentTop="true"
        android:layout_marginTop="54dp"
        android:inputType="textMultiLine" >

        <requestFocus />
    </EditText>

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/editText1"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="43dp"
        android:text="CREATE TEXT FILE IN A EXTERNALMEMORY" />

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:layout_marginBottom="27dp"
        android:text=""
    </TextView>

```

```

        android:textAppearance="?android:attr/textAppearanceLarge" />

<TextView
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_alignRight="@+id/button1"
    android:text="TEXT FILE IN A EXTERNALMEMORY"
    android:textAppearance="?android:attr/textAppearanceMedium" />

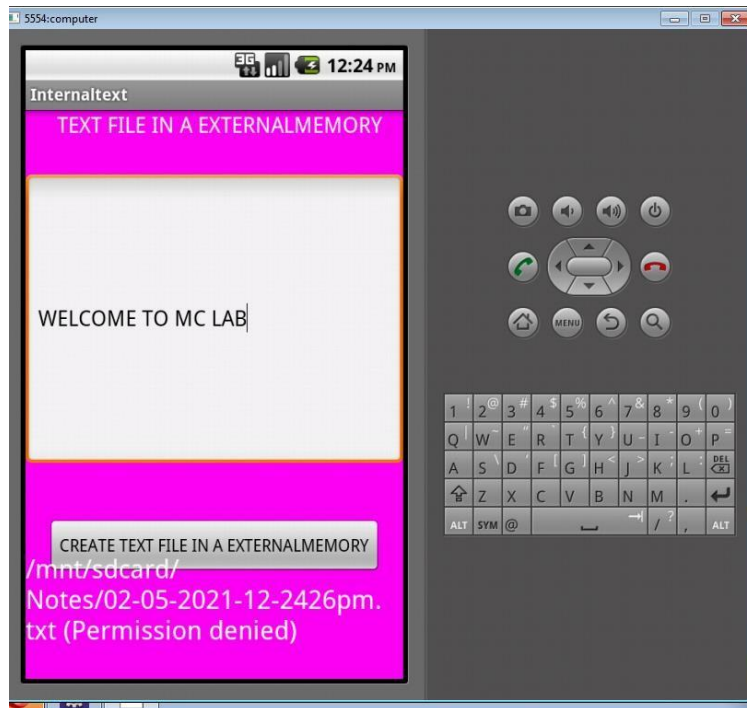
</RelativeLayout>

</LinearLayout>

```

ANDROIDMANIFEST.XML

<Uses permission android:name="android.permission WRITE_STORAGE_ INTERNA



RESULT

Thus successfully android application to create text file in internal memory.

EX:18

CREATE TEXT FILE IN A INTERNAL MEMORY

AIM

Write a program to create text file in internal memory

1. Open eclipse or android studio and select new android project
2. Give project name and select next
3. Choose the android version.Choose the lowest android version(Android 2.2) and select next
4. Enter the package name.package name must be two word seprated by comma and click finish
5. Go to package explorer in the left hand side.select our project.
6. Go to res folder and select layout.Double click the main.xml file.Don't change anything in layout.Leave as default.
- 7.Now select mainactivity.java file and to create java class file to type based on create text file in a external storage application coding.
8. Modify the default content of res/layout/main.xml file to include design for layout create text file in a external storage
9. Go to res/AndroidManifest.xml to allow uses permission for “<Uses permission android:name=”android:permission WRITE_INTERNAL_EXTERNAL”/>”.
- 10.Run the application to launch Android emulator and verify the result of the changes done in the application.

11.externalActivity.java
package your. internaltext.namespace;

```
import android.app.Activity;
import android.os.Bundle;
import android.os.Environment;
import android.text.format.DateFormat; import android.view.View;
import android.widget.Button;
```

```

import android.widget.EditText;
import android.widget.TextView;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
public class internaltextActivity extends Activity
{
    EditText editText;
    Button button;
    TextView textView;
    String h;
    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        editText = (EditText) findViewById(R.id.editText1);
        textView= (TextView) findViewById(R.id.textView1);
        button = (Button) findViewById(R.id.button1);
        button.setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View v) { try {
                h = DateFormat.format("MM-dd-yyyyy-h-mmssaa",
                System.currentTimeMillis()).toString();

```

```

File root = new File(Environment.getExternalStorageDirectory(),
"Notes");

```

```

if (!root.exists()) {
    root.mkdirs(); // this will create folder.

```

```

    }
    File filepath = new File(root, h + ".txt"); // file path to save
    FileWriter writer = new FileWriter(filepath);

```

```

writer.append(editText.getText().toString());
writer.flush(); writer.close();
String m = "File generated with name " + h + ".txt"; textView.setText(m);
} catch (IOException e) { e.printStackTrace();
textView.setText(e.getMessage().toString());
}
}
});
}
}
}

```

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"
    android:background="#FF00F0">
    <RelativeLayout
        android:id="@+id/relativeLayout1"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="center" >
        <EditText
            android:id="@+id/editText1"
            android:layout_width="wrap_content"
            android:layout_height="250dp"
            android:layout_alignParentLeft="true"
            android:layout_alignParentRight="true"
            android:layout_alignParentTop="true"
            android:layout_marginTop="54dp"
            android:inputType="textMultiLine" >

            <requestFocus />
        </EditText>

        <Button
            android:id="@+id/button1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_below="@+id/editText1"
            android:layout_centerHorizontal="true"
            android:layout_marginTop="43dp"
            android:text="CREATE TEXT FILE IN A EXTERNALMEMORY" />

        <TextView
            android:id="@+id/textView1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentBottom="true"
            android:layout_centerHorizontal="true"
            android:layout_marginBottom="27dp"
            android:text=""

```

```

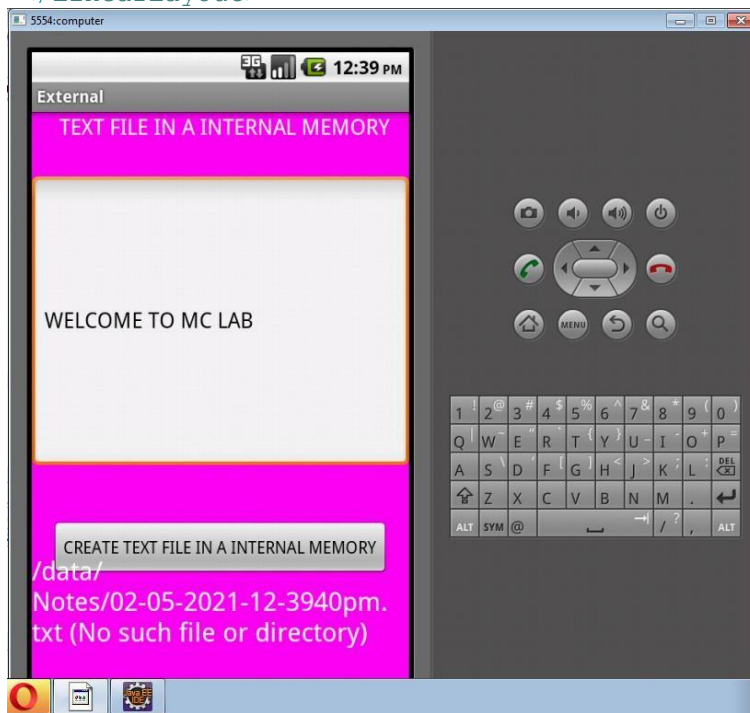
        android:textAppearance="?android:attr/textAppearanceLarge" />

<TextView
    android:id="@+id/textView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_alignRight="@+id/button1"
    android:text="TEXT FILE IN A EXTERNALMEMORY"
    android:textAppearance="?android:attr/textAppearanceMedium" />

</RelativeLayout>

</LinearLayout>

```



RESULT

Thus successfully android application to create text file in internal memory.

Ex no :19

AndroidGoogleMapsActivity.java

1. Open eclipse or android studio and select new android project
2. Give project name and select next
3. Choose the android version.Choose the lowest android version(Android 2.2) and select next

4. Enter the package name.package name must be two word seprated by comma and click finish
5. Go to package explorer in the left hand side.select our project.
6. Go to res folder and select layout.Double click the main.xml file.Don't change anything in layout.Leave as default.
7. Beginning with the Android SDK release v1.0, you need to apply for a free Google Maps API key before you can integrate Google Maps into your Android application.
8. When you apply for the key, you must also agree to Google's terms of use, so be sure to read them carefully.
9. To apply for a key, follow the series of steps outlined next.
- 10.First, if you are testing the application on the Android Emulator or an Android device directly connected to your development machine, locate the SDK debug certificate located in the default folder (C:\Users\<username>\.android for Windows 7 users).
- 11.You can verify the existence of the debug certificate by going to Eclipse and selecting Window ⇨ Preferences. Expand the Android item and select Build .
- 12.On the right side of the window, you will be able to see the debug certificate's location.
- 13.In this example, my MD5 fingerprint is
EF:7A:61:EA:AF:E0:B4:2D:FD:43:5E:1D:26:04:34:BA
- 14.Copy the MD5 certificate fingerprint and navigate your web browser to: <http://code.google.com/>

```
package com.androidhive.googlemaps;
```

```
import android.os.Bundle;
```

```

import com.google.android.maps.MapActivity;

public class AndroidGoogleMapsActivity extends
    MapActivity { @Override
    public void onCreate(Bundle
        savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }

    @Override
    protected boolean isRouteDisplayed() {
        return false;
    }
}

```

AndroidGoogleMapsActivity.java

package

com.androidhive.googlemaps;

import android.os.Bundle;

```

import
com.google.android.maps.MapActivity;
import
com.google.android.maps.MapView;

```

```

public class AndroidGoogleMapsActivity extends
    MapActivity { @Override
    public void onCreate(Bundle
        savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        // Displaying Zooming controls
        MapView mapView = (MapView)
        findViewById(R.id.mapView);
        mapView.setBuiltInZoomControls(true);
    }
}

```



```

    }

    @Override
    protected boolean isRouteDisplayed() {
        return false;
    }
}

```

main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<com.google.android.maps.MapView
    xmlns:android="http://schemas.android.com/apk/res/a
    ndroid" android:id="@+id/mapView"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:clickable="true"
    android:apiKey="05M-7qOGbEjYduPPUdQgJt9ysL8HToawGdvu_ow"
/>

```

ANDROIDMANIFEST.XML

```

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

<!-- Add Google Map Library -->
    <uses-library android:name="com.google.android.maps" />

```



Result

Thus successfully android application to create text file in internal memory.