

# Practical No. 1

Experiment No. \_\_\_\_\_ Name: \_\_\_\_\_

DOMS Page No. 1  
Date: \_\_\_\_\_

Write a program to demonstrate different types of layouts.

Main Activity

Package com.example.layout;

```
import android.os.Bundle; import  
android.view.Menu;
```

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

@Override

```
public boolean onCreateOptionsMenu(Menu menu) {  
    // inflate the menu; this adds items to the action  
    // bar if it is present.  
    getMenuInflater().inflate(R.menu.main, menu);  
    return true;  
}
```

}

}

Activity - Main

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk  
    xmlns:tools="https://schemas.android.com/tools" android:layout  
    width="match_parent" ...
```

android:layout-height = "match-parent" android:paddingBottom = "@dimen/activity-vertical-margin"  
android:paddingLeft = "@dimen/activity-horizontal-margin"  
android:paddingRight = "@dimen/activity-horizontal-margin"  
android:paddingTop = "@dimen/activity-vertical-margin"  
tools:context = "Main activity" >

## < TextView

    android:id = "@+id/TextView1"  
    android:layout-width = "wrap-content"  
    android:layout-height = "wrap-content"  
    android:text = "@string/hello-world" />

## < LinearLayout

    android:layout-width = "wrap-content" android:layout-height = "wrap-content"  
    android:paddingBottom = "@dimen/activity-vertical-margin"  
    android:paddingLeft = "@dimen/activity-horizontal-margin"  
    android:paddingLeft = "@dimen/activity-horizontal-margin"  
    android:paddingTop = "@dimen/activity-vertical-margin" tools:context = "Main activity"

## < TextView

    android:id = "@+id/textView2" android:layout-width = "wrap-content"  
    android:layout-height = "wrap-content" android:text = "@string/hello-world" />

## < TextView

    android:id = "@+id/textView3" android:layout-width = "wrap-content"  
    android:layout-height = "wrap-content" android:text = "@string/hello-world" />

</LinearLayout>

</RelativeLayout>

## Practical - 2

Experiment No. \_\_\_\_\_ Name : \_\_\_\_\_

DOMS Page No. 3  
Date | |

Write a program to implement simple calculator.

Navigate to the app > res > Layout > activity\_main.xml and add the below code to that file. Below is the code for the activity\_main.xml file.

XML

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_Parent"
    android:layout_height="match_Parent"
    android:background="#8BC34A"
    android:backgroundTint="@android:color/darker_gray"
    tools:context=".MainActivity">
```

<!-- TextView to display our basic heading of "calculator-->

<TextView

```
    android:layout_width="194dp"
    android:layout_height="43dp"
    android:layout_marginStart="114dp"
    android:layout_marginLeft="114dp"
    android:layout_marginTop="58dp"
    android:layout_marginEnd="103dp"
```

Teacher's Signature: \_\_\_\_\_

```
    android:layout_marginRight = "103dp"
    android:layout_marginBottom = "502dp"
    android:scrollbarSize = "30dp"
    android:text = "calculator"
    android:textAppearance = "@style/TextAppearance.AppCompat.Body1"
    android:textSize = "30dp"
    app:layout_constraintBottom_toBottomOf = "Parent"
    app:layout_constraintEnd_toEndOf = "Parent"
    app:layout_constraintStart_toStartOf = "Parent"
    app:layout_constraintTop_toTopOf = "Parent" />
```

<1-. Edit view to input the values ->

<Edit View

```
    android:id = "@+id/num1"
    android:layout_width = "346dp"
    android:layout_height = "28dp"
    android:layout_marginStart = "72dp"
    android:layout_marginEnd = "71dp"
    android:layout_marginBottom = "416dp"
    android:ems = "10"
    android:hint = "number 1 (0)"
    android:inputType = "number"
    app:layout_constraintBottom_toBottomOf = "Parent"
    app:layout_constraintEnd_toEndOf = "Parent"
    app:layout_constraintStart_toStartOf = "Parent"
```

<!-- A button to perform 'sum' operation -->  
<Button

```
    android:id = "@+id/sum"
    android:layout_width = "wrap_content"
    android:layout_height = "wrap_content"
    android:layout_marginStart = "16dp"
    android:layout_marginTop = "292dp"
    android:layout_marginEnd = "307dp"
    android:layout_marginBottom = "263dp"
    android:backgroundTint = "@android:color/holo-red-light"
    android:onClick = "dosum"
    android:text = "+"
    app:layout_constraintBottom_toBottomOf = "Parent"
    app:layout_constraintEnd_toEndOf = "Parent"
    app:layout_constraintStart_toStartOf = "Parent"
    app:layout_constraintTop_toTopOf = "Parent"/>
```

< Button

```
    android:id = "@+id/button"
    android:layout_width = "92dp"
    android:layout_height = "48dp"
    android:layout_marginStart = "113dp"
    android:layout_marginTop = "356dp"
    android:layout_marginEnd = "206dp"
    android:layout_marginBottom = "199dp"
    android:layoutBackgroundTint = "@android:color/holo-red-light"
    android:onClick = "domood"
```

android:text = "% (mod)"

app:layout\_constraintEnd\_toEndOf = "parent"

app:layout\_constraintStart\_toStartOf = "Parent"

app:layout\_constraintTop\_toTopOf = "Parent" />

<!-- A button to perform a power function -->

<Button

    android:id = "@+id/pow"

    android:layout\_width = "wrap\_content"

    android:layout\_height = "wrap\_content"

    android:layout\_marginStart = "113dp"

    android:layout\_marginTop = "292dp"

    android:layout\_marginEnd = "210dp"

    android:layout\_marginBottom = "263dp"

    android:backgroundTint = "@android:color/holo-red-light"

    android:onClick = "doPow"

    android:text = "n<sup>1</sup> ^ n<sup>2</sup>"

    app:layout\_constraintBottom\_toBottomOf = "parent"

    app:layout\_constraintEnd\_toEndOf = "parent"

    app:layout\_constraintStart\_toStartOf = "parent"

    app:layout\_constraintTop\_toTopOf = "parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

Log

# Practical No. 3

Experiment No. \_\_\_\_\_ Name: \_\_\_\_\_

DOMS Page No. 7  
Date: \_\_\_\_\_

Write a program to demonstrate list view.

Structure of listview project

- △  main
- △  java
  - △  ListViewExample.com.ListView
  - ④ MainActivity
- △  res
  - △  drawable
  - △  drawable - v24
  - △  layout
    - activity\_main.xml
    - myList.xml
  - △  mipmap-anydpi-v26
  - △  mipmap-hdpi-
  - △  mipmap-mdpi
  - △  mipmap-xhdpi
  - △  mipmap-xxhdpi
  - △  mipmap-xxxhdpi
  - △  values
    - colors.xml
    - strings.xml
    - styles.xml

activity\_main.xml

First we need to drag and drop listview component from palette to activity\_main.xml file.

Teacher's Signature: \_\_\_\_\_

## File : activity\_main.xml

1. <?xml version = "1.0" encoding = "utf-8"?>
2. <android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android">
3. xmlns:app = "http://schemas.android.com/apk/res-auto"
4. xmlns:tools = "http://schemas.android.com/tools"
5. android:layout\_width = "match\_parent"
6. android:layout\_height = "match\_parent"
7. tools:context = "Listview.example.com.listview.MainActivity">
- 8.
9. <ListView
10. android:id = "@+id/listview"
11. android:layout\_width = "match\_parent"
12. android:layout\_height = "fill\_parent"
13. />
14. </android.support.constraint.ConstraintLayout>

Create an additional mylist.xml file in layout folder which contains view components displayed in ListView.

~~mylist.xml~~

## File : mylist.xml

1. <?xml version = "1.0" encoding = "utf"?>
- 2.
3. <TextView xmlns:android = "http://schemas.android.com/res">
4. android:id = "@+id/textview"
5. android:layout\_width = "wrap\_content"

6. android:layout\_height = "wrap-content"
7. android:text = "Medium Text"
8. android:textStyle = "bold"
9. android:textAppearance = "?android:attr/textAppearanceMedium"
10. android:layout\_marginLeft = "10dp"
11. android:layout\_marginTop = "5dp"
12. android:padding = "2dp"
13. android:textColor = "#4d4d4d"
14. />

Now place the list of data in strings.xml file by creating string\_array.

### strings.xml

#### File: strings.xml

1. <resources>
2. <string name = "app-name"> Listview </string>
3. <string-array name = "array-technology">
4. <item> Android </item>
5. <item> java </item>
6. <item> php </item>
7. <item> Hadoop </item>
8. <item> Sab. </item>
9. <item> Python </item>
10. <item> Ajax </item>
11. <item> C++ </item>
12. <item> Rubix </item>
13. <item> Rails </item>
14. <item> Net </item>

Experiment No. \_\_\_\_\_ Name: \_\_\_\_\_

DOMS Page No. 10  
Date | |

15. <item> perl </item>
16. <item> </string-array>
17. <item> </resources>

Output

listview

Android

Java

PHP

Hadoop

Sap

Python

Ajax

C++

Ruby

Rails

Wet

# Practical - 4

Experiment No. \_\_\_\_\_ Name: \_\_\_\_\_

DOMS Page No. 11  
Date: \_\_\_\_\_

Write a program to demonstrate date picker and time picker.

Step 1 - Create a new project in Android Studio, go to file = New project and fill all required details to create a new project.

Step 2:- Add the following code to res/layout/activity-main.xml.

```
<?xml version = "1.0" encoding = "utf-8"?>
<RelativeLayout xmlns:android = "http://schemas.android.com/res/android"
    xmlns:tools = "https://schemas.android.com/tools"
    android:layout_width = "match_parent" android:layout_height = "match_parent"
    tools:context = "MainActivity">
    <Button
        android:id = "@+id/btnPick" android:layout_width = "wrap_content"
        android:layout_height = "wrap_content"
        android:layout_centerHorizontal = "true"
        android:layout_marginTop = "30dp" android:text = "pick date and Time"/>
    <TextView
        android:id = "@+id/textview" android:layout_width = "wrap_content"
        android:layout_height = "wrap_content"
        android:layout_centerInParent = "true" android:textSize = "16sp"
        android:textStyle = "bold"/>
</RelativeLayout>
```

Step 3:- Add the following code to src/MainActivity.java

```
import android.app.DatePickerDialog;
import android.app.TimePickerDialog;
```

Teacher's Signature: \_\_\_\_\_

```
import android.support.v7.app.AppCompatActivity; import  
android.os.Bundle;  
import android.text.format.DateFormat; import  
android.view.View;  
import android.widget.Button; import  
android.widget.DatePicker; import  
android.widget.TextView; import  
android.widget.TimePicker; import  
java.util.Calendar;  
  
public class MainActivity extends AppCompatActivity implements  
DatePickerDialog.OnDateSetListener, TimePickerDialog.OnTimeSetListener {  
    TextView textView;  
    Button button;  
    int day, month, year, hour, minute;  
    int myday, mymonth, myyear, myhour, myminute; @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        setContentView(R.layout.activity_main);  
        textView = findViewById(R.id.textView); button =  
        findViewById(R.id.btnPick);  
        button.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View v) {  
                Calendar calendar = Calendar.getInstance();  
                Year = calendar.get(Calendar.YEAR); Month =  
                calendar.get(Calendar.MONTH);  
                day = calendar.get(Calendar.DAY_OF_MONTH);  
                DatePickerDialog datePickerDialog = new  
                DatePickerDialog(MainActivity.this, MainActivity.this, year, month, day);  
                datePickerDialog.show();  
            }  
        });  
    }  
}
```

3)

3

Public void onDateSet (DatePicker view, int year, int month, int dayof

my Year = year;

my day = day;

my Month = month;

calendar c = Calendar.getInstance(); hour =  
c.get(Calendar.HOUR);

minute = c.get(Calendar.MINUTE);

TimePickerDialog timePickerDialog = New

TimePickerDialog (MainActivity.this, MainActivity.this, hour, minute,  
Date Format is 24 HourFormat (this));

timePickerDialog.show();

3

@override

Public void onTimeSet (TimePicker view, int hourOfDay, int minute)

myHour = hourOfDay;

my Minute = minute;

TextView.setText ("Year " + my year + "\n" + month +  
my Month + "\n" +

"Day " + my day + "\n" + "Hour "

my Hour + "\n" + "Minute " +

my Minute);

3

~~Step 4:- Add the following code to androidManifest.xml~~

<?xml version="1.0" encoding="UTF-8"?>

<manifest xmlns:android="http://schemas.android.com/res

```
Package = "app.com.sample">
<application android.allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app-name"
    android:roundIcon="@mipmap/ic_launcher_round"
    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>
```

Ques

## Practical No - 5

Experiment No. \_\_\_\_\_ Name : \_\_\_\_\_

DOMS Page No. 14  
Date | |

Develop a simple application With Context Menu and Option Menu.

activity\_main.xml

Drag one listview from the palette , now the xml file will look like this ;

File.activity\_main.xml

```
1. <?xml version = "1.0 "encoding = "utf - 8 "?>
2. <android . support . constraintlayout xmlns: android = "https://schemas.android.com"
3. xmlns: app = "http://schemas.android.com/res/app/res-auto"
4. xmlns: tools = "http://schemas.android.com/tools"
5. android: layout - width = "match _ Parent"
6. android: layout - height = "match - Parent"
7. tools: content = "example - java point . com . contextmenu>MainActivity">
8.
9. < ListView
10. android: layout - width = "368 dP "
11. android: layout - height = "465 dP "
12. android: id = "@+ id /listview"
13. android: layout - marginEnd = "8dP "
14. android: layout - marginStart = "8dP "
15. android: layout - marginTop = "8dP "
16. app: layout - constraintEnd - toEndOf = "Parent"
17. app: layout - constraintHorizontal - bias = "0.0"
18. app: layout - constraintStart - toStartOf = "Parent"
19. app: layout - constraintTop - toTopOf = "Parent" />
20. </ android . support . constraint . constraintLayout >
```

Teacher's Signature: \_\_\_\_\_

### main\_menu.xml

Create a separate menu\_main.xml file in menu directory for menu items.

1. <?xml version="1.0" encoding="utf-8" ?>
2. <menu xmlns:android="http://schemas.android.com/apk/res">
3. <item android:id="@+id/call"
4. android:title="Call"/>
5. <item android:id="@+id/sms"
6. android:title="SMS"/>
7. </menu>

### Activity Menu

Let's write the code to display context menu on press of the listview.

#### File MainActivity.java

1. Package example.javapoint.com.contextmenu;
- 2.
3. import android.support.v7.app.AppCompatActivity;
4. import android.os.Bundle;
5. import android.view.MenuInflater;
6. import android.view.MenuItem;
7. import android.view.ContextMenu;
8. ~~import android.view.View;~~
9. ~~import android.widget.ArrayAdapter;~~
10. ~~import android.widget.ListView;~~
11. ~~import android.widget.Toast;~~
- 12.

13. Public class MainActivity extends AppCompatActivity {
14. ListView listView;
15. String contacts [] = {"Ajay", "Sohin", "Sumit", "Tawun", "Yogesh"};
16. @Override
17. Protected void onCreate(Bundle savedInstanceState) {
18. Super.onCreate(savedInstanceState);
19. SetContentView(R.layout.activity\_main);
20. ListView = (ListView) findViewById(R.id.listView);
21. ArrayAdapter<String> adapter = new ArrayAdapter<String>(this, android.R.layout.simple\_list\_item\_1, contacts);
22. ListView.setAdapter(adapter);
23. // Register the ListView for Context menu
24. registerForContextMenu(ListView);
25. }
26. @Override
27. Public void onCreateContextMenu(ContextMenu menu, View v, ContextMenu.ContextMenuInfo menuInfo)
28. {
29. Super.onCreateContextMenu(menu, menuInfo);
30. getMenuInflater inflater = getMenuInflater();
31. inflater.inflate(R.menu.main\_menu);
32. menu.setHeaderTitle("Select The Action");
33. }
34. @Override
35. Public boolean onContextItemSelected(MenuItem item){
36. if(item.getItemId() == R.id.call){
37. Toast.makeText(getApplicationContext(), "Calling code", Toast.

LENGTH - LONG).

show();

38. }

39. else if(item.getItemId() == R.id.sms) {

40. Toast.makeText(getApplicationContext(),"sending sms code",Toast.LENGTH\_LONG).  
show();

41. } else {

42. return false;

43. }

44. return true;

45. }

46. }

Output

context menu

Ajay

Sachin

Sumit

Tarun

Yogesh

By

# Practical No. 6

2015 Page 18

Write a program to send email in android.

We can easily send email in android via activity\_main.xml

Drag the 2 EditTexts, 1 MultiLine EditText, 3 TextViews and 1 button from the palette, now the activity\_main.xml file will like this:

File.activity\_main.xml

```
1. <RelativeLayout xmlns:android="http://schemas.android.com/res-  
2.     xmlns:tools="https://schemas.android.com/tools"  
3.     android:layout_width="match_parent"  
4.     android:layout_height="match_parent"  
5.     tools:context="MainActivity">  
6.
```

```
7.     <EditText  
8.         android:id="@+id/editText"  
9.         android:layout_width="wrap-content"  
10.        android:layout_height="wrap-content"  
11.        android:layout-alignParentRight="true"  
12.        android:layout_alignParentTop="true"  
13.        android:layout_marginRight="22dp"  
14.        android:layout_marginTop="16dp"  
15.        android:ems="10" />  
16.    16.
```

```
17.     <EditText  
18.         android:id="@+id/editText2"  
19.         android:layout_width="wrap_content"  
20.         android:layout_height="wrap_content"
```

21. android:layout\_alignLeft = "@+id/editText1"
22. android:layout\_below = "@+id/editText1"
23. android:layout\_marginTop = "18dp"
24. android:ems = "10" />
25. 25.
26. <requestFocus />
27. </EditText>
28. 28.
29. <EditText
30. android:id = "@+id/editText3"
31. android:layout\_width = "wrap-content"
32. android:layout\_height = "wrap-content"
33. android:layout\_alignLeft = "@+id/editText2"
34. android:layout\_below = "@+id/editText2"
35. android:layout\_marginTop = "28dp"
36. android:ems = "10"
37. android:inputType = "textMultiline" />
38. 38.
39. <TextView
40. android:id = "@+id/textview1"
41. android:layout\_width = "wrap-content"
42. android:layout\_height = "wrap-content"
43. android:layout\_alignBaseline = "@+id/editText1"
44. android:layout\_alignBottom = "@+id/editText1"
45. android:layout\_alignParentLeft = "true"
46. android:layout\_text = "70" />
47. 47.

48. < TextView

49. android:id = "@+id/textview2"

50. android:layout\_width = "wrap content"

51. android:layout\_height = "wrap content"

52. android:layout\_alignBaseline = "@+id/edittext2"

53. android:layout\_alignBottom = "@+id/edittext2"

54. android:layout\_alignParentLeft = "true"

55. android:text = "subject"/>

56. 56.

57. < TextView

58. android:id = "@+id/edittextview3"

59. android:layout\_width = "wrap content"

60. android:layout\_height = "wrap content"

61. android:layout\_alignBaseline = "@+id/edittext3"

62. android:layout\_alignBottom = "@+id/edittext3"

63. android:layout\_alignParentLeft = "true"

64. android:text = "message"/>

65. 65.

66. Button

67. android:id = "@+id/button1"

68. android:layout\_width = "wrap content"

69. android:layout\_height = "wrap content"

70. android:layout\_alignLeft = "@+id/editText3"

71. android:layout\_below = "@+id/editText3"

72. android:layout\_marginLeft = "76dp"

73. android:layout\_marginTop = "20dp"

74. android:text = "send"/>

75. 75.

76. </RelativeLayout>

Experiment No. \_\_\_\_\_ Name : \_\_\_\_\_

75. 75.

76. &lt;/RelativeLayout&gt;

Activity Class

File. MainActivity.java

1. package com.example.sendmail;
- 2.
3. import android.os.Bundle;
4. import android.app.Activity;
5. import android.content.Intent;
6. import android.view.Menu;
7. import android.view.View;
8. import android.view.View.OnClickListener;
9. import android.widget.Button;
10. import android.widget.EditText;
11. 11.
12. public class MainActivity extends Activity {
13. EditText editTextTo, editTextSubject, editTextMessage;
14. Button send;
15. @Override
16. protected void onCreate(Bundle savedInstanceState) {
17. super.onCreate(savedInstanceState);
18. setContentView(R.layout.activity\_main);
19. 19.
20. editTextTo = (EditText) findViewById(R.id.editText1)
21. editTextSubject = (EditText) findViewById(R.id.editText2)
22. editTextMessage = (EditText) findViewById(R.id.editText3)

23. 23.

24. Send = (Button) Find view By Id (R.id.button1);

25. 25.

26. send.setOnClickListener (new OnClickListener() {

27. 27.

28. @Override

29. public void onclick (View arg0) {

30. String to = editTextTo.getText () .toString ();

31. String subject = editTextSubject.getText () .toString ();

32. String message = editTextMessage.getText () .toString ();

33. 33.

34. Intent email = new Intent (Intent.ACTION\_SEND);

35. email.putExtra (Intent.EXTRA\_EMAIL, new String ["to"]);

36. email.putExtra (Intent.EXTRA\_SUBJECT, subject);

37. email.putExtra (Intent.EXTRA\_TEXT, message);

38. 38.

39. // need this to prompts email client only

40. email.setType ("message/rfc822");

41. 41.

42. startActivityForResult (Intent.createChooser (email, "choose an Email Client"));

43. 3

44. 3);

45. 3

46. @Override

47. public boolean onCreateOptionsMenu (MenuItem) {

48. // inflate the menu; this adds items to the actionbar if it is present

49. getMenuInflater ().inflate (R.menu.activity\_main\_menu);

50. return true;

51. 3

# Practical No - 7

Experiment No. \_\_\_\_\_ Name : \_\_\_\_\_

DOMS Page No. 23  
Date | | |

Write a program to demonstrate a service

activity - main.xml

Drag the 3 buttons from the palette, now the activity - main.xml will look like this ;

File: activity - main - xml

1. <?xml version = "1.0" encoding = "utf-8"?>
2. <RelativeLayout xmlns:android = "http://schemas.android.com/apk/res/android"
3. xmlns:app = "http://schemas.android.com/apk/res-auto"
4. xmlns: tools = "http://schemas.android.com/tools"
5. android:layout\_width = "match\_parent"
6. android:layout\_height = "match\_parent"
7. tool:context = "example.javapoint.com.AndroidService.MainActivity"
- 8.
- 9.
10. <Button
11. android:id = "@+id/buttonStart"
12. android:layout\_width = "wrap\_content"
13. android:layout\_height = "wrap\_content"
14. android:layout\_alignParentTop = "true"
15. android:layout\_centerHorizontal = "true"
16. android:layout\_marginTop = "74dp"
17. android:text = "Start Service"/>
18. 18:

19. &lt;Button

20. android:id = "@+id/buttonStop"

21. android:layout\_width = "wrap-content"

22. android:layout\_height = "wrap-content"

23. android:layout\_centerHorizontal = "true"

24. android:layout\_centerVertical = "true"

25. android:text = "Stop Service"/&gt;

26. 26.

27. &lt;Button

28. android:id = "@+id/buttonNext"

29. android:layout\_width = "wrap-content"

30. android:layout\_height = "wrap-content"

31. android:layout\_alignParentBottom = "true"

32. android:layout\_centerHorizontal = "true"

33. android:layout\_marginBottom = "63dp"

34. android:text = "Next Page"/&gt;

35. &lt;/RelativeLayout&gt;

activity-next.xml

gt is the layout file of next activity.

File: activity-next.xml

gt contains only one TextView displaying the message Next Page

1. &lt;?xml version="1.0" encoding="UTF-8"?&gt;

2. &lt;android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

3. `xml:app = "http://schemas.android.com/apk/res/android"`  
4. `xmlns:tools = "http://schemas.android.com/tools"`  
5. `android:layout_width = "match_parent"`  
6. `android:layout_height = "match_parent"`  
7. `tools:context = "example.javapoint.com.android.service.NextPage">`  
8.  
9. `<Text view`  
10. `android:id = "@+id/textview"`  
11. `android:layout_width = "wrap-content"`  
12. `android:layout_height = "wrap-content"`  
13. `android:layout-marginEnd = "8dp"`  
14. `android:layout_marginStart = "8dp"`  
15. `android:layout-marginTop = "200dp"`  
16. `android:text = "Next Page"`  
17. `app:layout_constraintEnd_toEndOf = "Parent"`  
18. `app:layout_constraintStart_toStartOf = "Parent"`  
19. `app:layout_constraintTop_toTopOf = "Parent"`  
20. `</android.support.constraint.ConstraintLayout>`

## Service class

Now create the service implementation class by inheriting the service class and overriding its callback methods.

File. `MyService.java`

1. `Package example.javapoint.com.android.service;`  
2.

Experiment No. \_\_\_\_\_ Name. \_\_\_\_\_

```
3. import android.app.Service;
4. import android.content.Intent;
5 import android.media.MediaPlayer;
6. import android.os.IBinder;
7. import android.support.annotation.Nullable;
8. import android.widget.Toast;
9
10. public class MyService extends Service {
11.     MediaPlayer myPlayer;
12.     @Nullable
13.     @Override
14.     public
15.     return null;
16. }
17. @Override
18. public void onCreate() {
19.     Toast.makeText(this, "service created", Toast.LENGTH_LONG).show();
20. }
21. myPlayer = MediaPlayer.create(this, R.raw.sun);
22. myPlayer.setLooping(false); // set looping
23. }
24. @Override
25. public void onStart(Intent intent, int startId) {
26.     Toast.makeText(this, "service started", Toast.LENGTH_LONG).show();
27.     myPlayer.start();
28. }
29. @Override
```

30. Public void onDestory () {

31. Toast.makeText (this, "service stopped", Toast.LENGTH\_LONG).show();

32. my player.stop ();

33.

34. }

## Activity Class

### File: MainActivity.java

1. Package example.javatpoint.com.androidservice;

2.

3. import android.content.ContentIntent;

4. import android.support.v7.app.AppCompatActivity;

5. import android.os.Bundle;

6. import android.view.View;

7. import android.widget.Button;

8.

9. Public class MainActivity extends AppCompatActivity {

10. Button buttonstart, buttonstop, buttonnext;

11. @Override

12. Protected void onCreate (Bundle savedInstanceState) {

13. Super.onCreate (savedInstanceState);

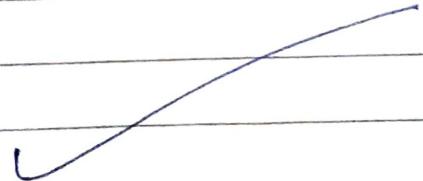
14. SetContentView (R.layout.activity\_main);

15. 15.

16. button\_start = findViewById (R.id.buttonstart);

17. button\_stop = findViewById (R.id.buttonstop);

18. buttonNext = FindViewById(R.id.buttonNext);
- 19.
20. buttonStart. SetOnClickListener(this);
21. buttonStop. SetOnClickListener(this);
22. buttonStart. SetOnClickListener(this);
23. 23.
24. 24.
25. }
26. Public void OnClick(Viewsrc) {
27. Switch(src.getId()) {
28. Case R.id.buttonStart:
29. 29.
30. StartService (new Intent(this, myservice.class));
31. break;
32. Case R.id.buttonStop;
33. StopService (new Intent(this, myservice.class));
34. break;
35. Case R.id.buttonNext;
36. Intent intent = new Intent(this, NextPage.class);
37. StartActivity(intent);
38. break;
39. }
40. }
41. }



Next Page Class  
Now creates another activity.

## File. Next Page . java

1. Package example.javapoint.com.androidservice;
- 2.
3. import android.support.v7.app.AppCompatActivity;
4. import android.os.Bundle;
- 5.
6. Public class NextPage extends AppCompatActivity {
- 7.
8. @Override
9. protected void onCreate(Bundle savedInstanceState) {
10. Super.onCreate(savedInstanceState);
11. setContentView(R.layout.activity\_next);
12. }
13. }

Declare the service in the AndroidManifest.xml file  
finally, declare the service in the manifest file.  
File. AndroidManifest.xml

1. <?xml version="1.0" encoding = "utf-8"?>
2. <manifest xmlns:android : "http://schemas.android.com/res/android"
3. Package = "example.javapoint.com.androidservice">
- 4.
5. <application
6. android:allowBackup = "true"

1. android: icon = "@ mipmap/ic\_launcher"
2. android: label = "@string/app-name"
3. android: roundIcon = "@ mipmap/ic-launcher-round"
4. android: supportsRtl = "true"
5. android: theme = "@ style/APP Theme">
6. <activity android:name = "android.intent.action.MAIN"/>
7. <intent-filter>
8. <activity android:name = "MainActivity">
9. 15.
10. <category android:name = "android.intent.category.LAUNCHER"/>
11. </intent-filter>
12. </activity>
13. <activity android:name = "NextPage"> </activity>
14. <service>
15. android: Name = "MyService"
16. android: enabled = "true" />
17. </application>
18. 24.
19. </manifest>

Ans