

## Experiment N0:- 3

**Aim:-** Write a program using activity class to show different events.

### **Procedure:-**

There are following three concepts related to Android Event Management –

- **Event Listeners** – An event listener is an interface in the View class that contains a single callback method. These methods will be called by the Android framework when the View to which the listener has been registered is triggered by user interaction with the item in the UI.
- **Event Listeners Registration** – Event Registration is the process by which an Event Handler gets registered with an Event Listener so that the handler is called when the Event Listener fires the event.
- **Event Handlers** – When an event happens and we have registered an event listener for the event, the event listener calls the Event Handlers, which is the method that actually handles the event.

### Event Listeners & Event Handlers

Event Handler	Event Listener & Description
onClick()	<b>OnClickListener()</b> This is called when the user either clicks or touches or focuses upon any

	<p>widget like button, text, image etc. You will use <code>onClick()</code> event handler to handle such event.</p>
<code>onLongClick()</code>	<p><b>OnLongClickListener()</b></p> <p>This is called when the user either clicks or touches or focuses upon any widget like button, text, image etc. for one or more seconds. You will use <code>onLongClick()</code> event handler to handle such event.</p>
<code>onFocusChange()</code>	<p><b>OnFocusChangeListener()</b></p> <p>This is called when the widget loses its focus i.e. user goes away from the view item. You will use <code>onFocusChange()</code> event handler to handle such event.</p>
<code>onKey()</code>	<p><b>OnFocusChangeListener()</b></p> <p>This is called when the user is focused on the item and presses or releases a hardware key on the device. You will use <code>onKey()</code> event handler to handle such event.</p>
<code>onTouch()</code>	<p><b>OnTouchListener()</b></p> <p>This is called when the user presses the key, releases the key, or any</p>

	movement gesture on the screen. You will use onTouch() event handler to handle such event.
onMenuItemClick()	<b>OnMenuItemClickListener()</b> This is called when the user selects a menu item. You will use onMenuItemClick() event handler to handle such event.
onCreateContextMenu()	<b>onCreateContextMenuListener()</b> This is called when the context menu is being built(as the result of a sustained "long click")

### Code:-

Following is the content of the modified main activity file

```
package com.example.myapplication;

import android.app.ProgressDialog;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {
    private ProgressDialog progress;
```

```

Button b1,b2;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    progress = new ProgressDialog(this);

    b1=(Button)findViewById(R.id.button);
    b2=(Button)findViewById(R.id.button2);
    b1.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View v) {
            TextView txtView = (TextView)
findViewById(R.id.textView);
            txtView.setTextSize(25);
        }
    });

    b2.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View v) {
            TextView txtView = (TextView)
findViewById(R.id.textView);
            txtView.setTextSize(55);
        }
    });
}
}

```

Following will be the content  
of **res/layout/activity\_main.xml** file –

```
<?xml version="1.0" encoding="utf-8"?>
```

## <RelativeLayout

```
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android: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=".MainActivity">
```

## <TextView

```
android:id="@+id/textView1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Event Handling "
android:layout_alignParentTop="true"
android:layout_centerHorizontal="true"
android:textSize="30dp"/>
```

## <TextView

```
android:id="@+id/textView2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="NK Orchid Clg "
android:textColor="#ff87ff09"
android:textSize="30dp"
android:layout_above="@+id/imageButton"
android:layout_centerHorizontal="true"
android:layout_marginBottom="40dp" />
```

## <ImageButton

```
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/imageButton"
android:src="@drawable/abc"
```

```
        android:layout_centerVertical="true"
        android:layout_centerHorizontal="true" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Small font"
    android:id="@+id/button"
    android:layout_below="@+id/imageButton"
    android:layout_centerHorizontal="true" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Large Font"
    android:id="@+id/button2"
    android:layout_below="@+id/button"
    android:layout_alignRight="@+id/button"
    android:layout_alignEnd="@+id/button" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello World!"
    android:id="@+id/textView"
    android:layout_below="@+id/button2"
    android:layout_centerHorizontal="true"
    android:textSize="25dp" />

</RelativeLayout>
```

Following will be the content of **res/values/strings.xml** to define two new constants –

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <string name="app_name">myapplication</string>
</resources>
```

Following is the default content of **AndroidManifest.xml** –

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

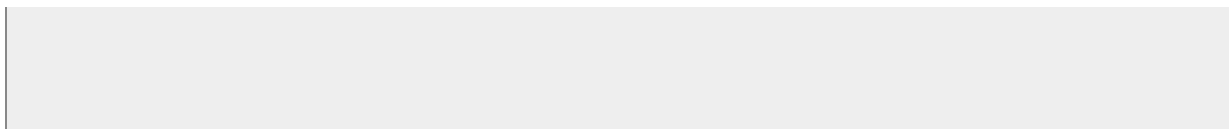
  <application
    android:allowBackup="true"
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name"
    android:theme="@style/AppTheme" >

    <activity
      android:name="com.example.myapplication.MainActivity"
      android:label="@string/app_name" >

      <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category
android:name="android.intent.category.LAUNCHER" />
        </intent-filter>

      </activity>

    </application>
</manifest>
```



**Output:-**





