UNIT - IV

ADVANCE UI PROGRAMMING

4.1 Event Driven Programming in Android

- Events are a useful way to collect data about a user's interaction with interactive components of your application, like button press or screen touch etc.

- There are following three concepts related to Android Event Management:
  - **Event Listeners**
  - **Event Handler**
  - **Event Handler Registration**

4.1 Event Driven Programming in Android

- **Event Listeners & Event Handlers**

Event Handler	Event Listener
onClick()	OnClickListener() This is called when the user either clicks or touches or focuses upon any widget like button, text,image etc. You will use onClick() event handler to handle such event.
onLongClick()	OnLongClickListener() 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 onLongClick() event handler to handle suchevent.
onFocusChange()	OnFocusChangeListener() This is called when the widget loses its focus ie. user goes away from the view item. You willuseonFocusChange() event handler to handle such event.
onKey()	OnFocusChangeListener() This is called when the user is focused on the item and presses or releases a hardware key onthe device. You will use onKey() event handler to handle such event.
onTouch()	OnTouchListener() This is called when the user presses the key, releases the key, or any movement gesture on thescreen. You will use onTouch() event handler to handle such event.
onMenuItemClick()	OnMenuItemClickListener() This is called when the user selects a menu item. You will use onMenuItemClick() event handlerto handle such event.

4.1 Event Driven Programming in Android

- **Event Handler Registration**
- Event Handler 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 receives notification when an event happens.

- There are three ways to register your event handler for any event:
  1. **Anonymous Inner Class**
  2. **Listener interface**
  3. **Layout file**

4.1 Event Driven Programming in Android

- **3. Layout File**
- In this approach, the event handler method is specified using the layout file (activity_main.xml) via the **android:onClick** attribute of the view.

- The event handler method must have a void return type and take a View as an argument.

4.1 Event Driven Programming in Android

3. Layout File (Example)

- **activity_main.xml**

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
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_centerHorizontal="true"
        android:layout_marginLeft="114dp"
        android:layout_marginTop="32dp"
        android:text="@string/hello_world" />
```

4.1 Event Driven Programming in Android

3. Layout File (Example)

- **activity_main.xml**

```
<Button
    android:id="@+id/Sbutton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/textView1"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="39dp"
    android:onClick="Sbutton_click"
    android:text="@string/btn_small" />

    <Button
    android:id="@+id/Lbutton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/Sbutton"
    android:layout_below="@+id/Sbutton"
    android:layout_marginTop="23dp"
    android:onClick="Lbutton_click"
    android:text="@string/btn_large" />

</RelativeLayout>
```

4.1 Event Driven Programming in Android

3. Layout File (Example)

- **activity_main.xml**

4.1 Event Driven Programming in Android

3. Layout File (Example)

- **strings.xml**

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

<resources>
<string name="app_name">App1</string>
<string name="hello_world">Hello world!</string>
<string name="action_settings">Settings</string>
<string name="btn_small">Smalll Font</string>
<string name="btn_large">Large Font</string>
</resources>
```

4.1 Event Driven Programming in Android

3. Layout File (Example)

- **MainActivity.java**

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

        Public void Sbutton_click(View v)
        {

           TextView txtView = (TextView) findViewById(R.id.textView1);
           txtView.setTextSize(10);
           return;
        }
```

4.1 Event Driven Programming in Android

3. Layout File (Example)

- **MainActivity.java**

```
Public void Lbutton_click(View v)
{

   TextView txtView = (TextView) findViewById(R.id.textView1);
   txtView.setTextSize(24);

   return;
}
}
```

4.1 Event Driven Programming in Android

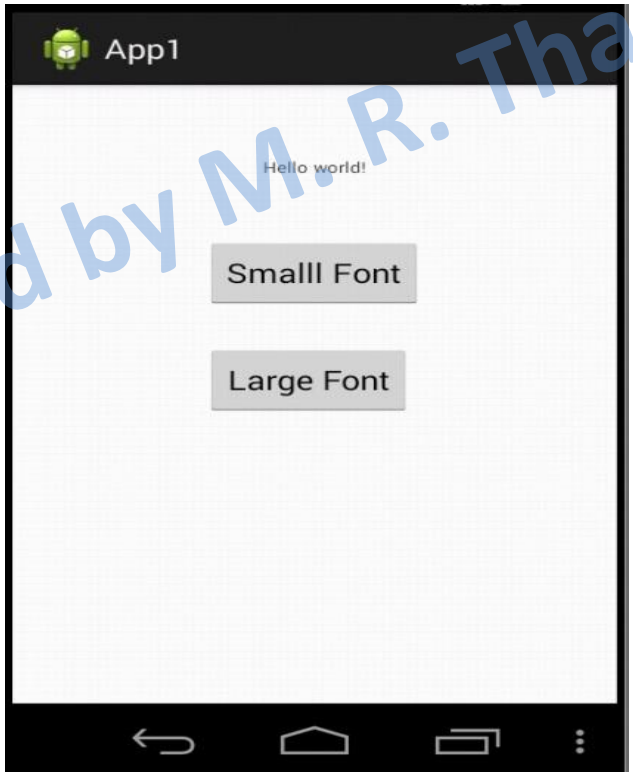
3. Layout File (Example)

- When you run the application the activity look like as shown in below figure:

4.1 Event Driven Programming in Android

3. Layout File (Example)

- When user clicks on Small Font button, the **Sbutton_click()** event handler captures the event and sets text size of the TextView to **10** as shown in below figure :



4.1 Event Driven Programming in Android

3. Layout File (Example)

- When user clicks on Large Font button, the **Lbutton_click()** event handler captures the event and sets text size of the TextView to **24** as shown in below figure :

4.1 Event Driven Programming in Android

1. Anonymous Inner Class

- In this method, you need to create an anonymous implementation of the listener class and each class is applied to only single control.

4.1 Event Driven Programming in Android

1. Anonymous Inner Class (Example)

- **activity_main.xml**

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
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_centerHorizontal="true"
        android:layout_marginLeft="114dp"
        android:layout_marginTop="32dp"
        android:text="@string/hello_world" />
```

4.1 Event Driven Programming in Android

1. Anonymous Inner Class (Example)

- **activity_main.xml**

```
<Button
    android:id="@+id/Sbutton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/textView1"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="39dp"
    android:text="@string/btn_small" />

<Button
    android:id="@+id/Lbutton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/Sbutton"
    android:layout_below="@+id/Sbutton"
    android:layout_marginTop="23dp"
    android:text="@string/btn_large" />

</RelativeLayout>
```

4.1 Event Driven Programming in Android

1. Anonymous Inner Class (Example)

- **activity_main.xml**

4.1 Event Driven Programming in Android

1. Anonymous Inner Class (Example)

- **strings.xml**

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

<resources>
<string name="app_name">App1</string>
<string name="hello_world">Hello world!</string>
<string name="action_settings">Settings</string>
<string name="btn_small">Smalll Font</string>
<string name="btn_large">Large Font</string>
</resources>
```

4.1 Event Driven Programming in Android

1. Anonymous Inner Class (Example)

- **MainActivity.java**

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

        Button Sbutton = (Button) findViewById(R.id.Sbutton);
        Button Lbutton = (Button) findViewById(R.id.Lbutton);
```

4.1 Event Driven Programming in Android

1. Anonymous Inner Class (Example)

- **MainActivity.java**

```
Sbutton.setOnClickListener(new View.OnClickListener()
{
        @Override
        Public void onClick(View v)
        {               TextView txt = (TextView) findViewById(R.id.textView1);
                        txt.setTextSize(10);
        }
    });

 Lbutton.setOnClickListener(new View.OnClickListener()
{
        @Override
        Public void onClick(View v)
        {               TextView txt = (TextView) findViewById(R.id.textView1);
                        txt.setTextSize(24);
        }
    });
 }
}
```

4.1 Event Driven Programming in Android

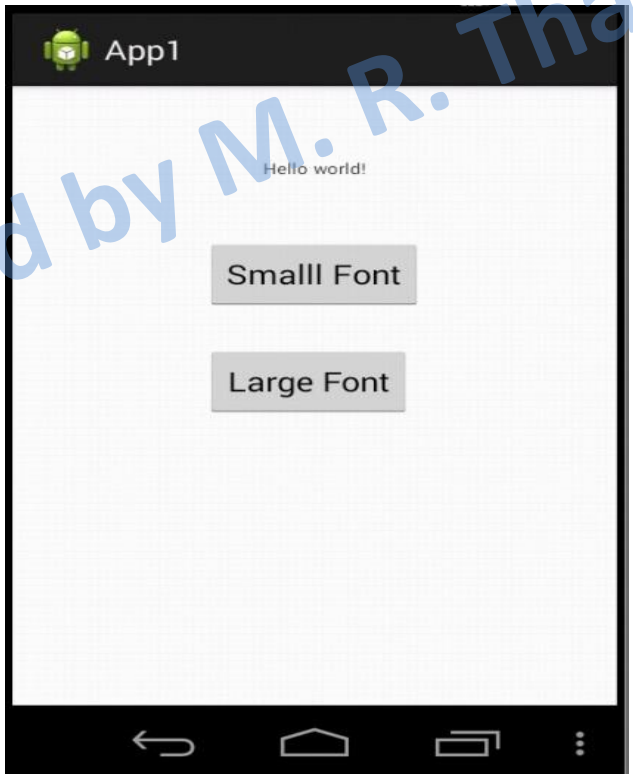
1. Anonymous Inner Class (Example)

- When you run the application the activity look like as shown in below figure:

4.1 Event Driven Programming in Android

1. Anonymous Inner Class (Example)

- When user clicks on Small Font button, the **onClickListener()** of the **Small font button** receives notification and it calls its **onclick()** event handler. The onClick() event handler sets text size of the TextView to **10** as shown in below figure:



4.1 Event Driven Programming in Android

1. Anonymous Inner Class (Example)

- When user clicks on Large Font button, the **onClickListener()** of the **Large Font button** receives notification and it calls its **onclick()** event handler. The onClick() event handler sets text size of the TextView to **24** as shown in below figure :

4.1 Event Driven Programming in Android

2. Listener interface

- In this method, activity class implements the Listener interface and the event handler method is implanted inside the activity class.

4.1 Event Driven Programming in Android

2. Listener interface (Example)

- **activity_main.xml**

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
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_centerHorizontal="true"
        android:layout_marginLeft="114dp"
        android:layout_marginTop="32dp"
        android:text="@string/hello_world" />
```

4.1 Event Driven Programming in Android

2. Listener interface (Example)

- **activity_main.xml**

```
<Button
    android:id="@+id/Sbutton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/textView1"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="39dp"
    android:text="@string/btn_small" />

<Button
    android:id="@+id/Lbutton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/Sbutton"
    android:layout_below="@+id/Sbutton"
    android:layout_marginTop="23dp"
    android:text="@string/btn_large" />

</RelativeLayout>
```

4.1 Event Driven Programming in Android

2. Listener interface (Example)

- **activity_main.xml**

4.1 Event Driven Programming in Android

2. Listener interface (Example)

- **strings.xml**

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

<resources>
<string name="app_name">App1</string>
<string name="hello_world">Hello world!</string>
<string name="action_settings">Settings</string>
<string name="btn_small">Smalll Font</string>
<string name="btn_large">Large Font</string>
</resources>
```

4.1 Event Driven Programming in Android

2. Listener interface (Example)

- **MainActivity.java**

```
public class MainActivity extends ActionBarActivity implements OnClickListener
{

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

       Button Sbutton = (Button) findViewById(R.id.Sbutton);
       Button Lbutton = (Button) findViewById(R.id.Lbutton);

       Sbutton.setOnClickListener(this);
       Lbutton.setOnClickListener(this);
    }
```

4.1 Event Driven Programming in Android

2. Listener interface (Example)

- **MainActivity.java**

```
@Override
Public void onClick(View v)
{
        if(v.getId() == R.id.Sbutton)
        {
                TextView txtView = (TextView) findViewById(R.id.textView1);
                txtView.setTextSize(10);
                return;
        }

        if(v.getId() == R.id.Lbutton)
        {
                TextView txtView = (TextView) findViewById(R.id.textView1);
                txtView.setTextSize(24);
                return;
        }
}
}
```

4.1 Event Driven Programming in Android

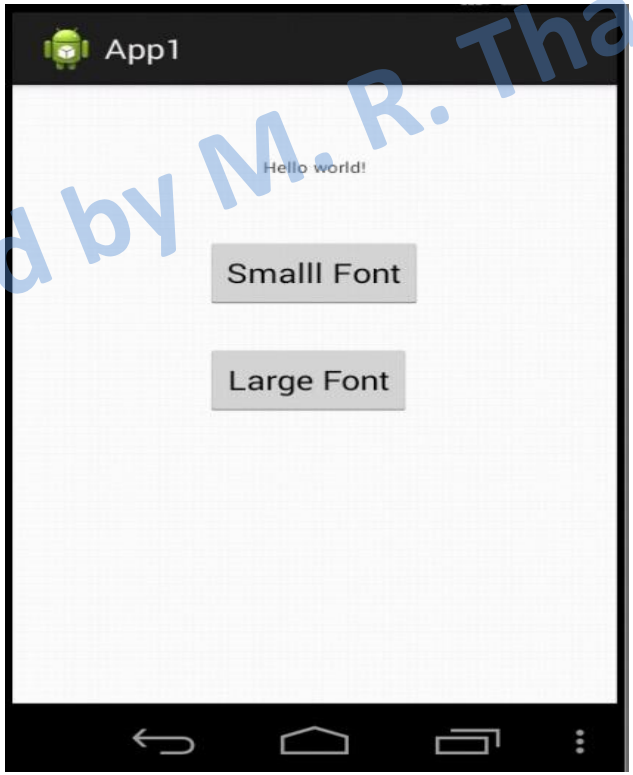
2. Listener interface (Example)

- When you run the application the activity look like as shown in below figure:

4.1 Event Driven Programming in Android

2. Listener interface (Example)

- When user clicks on Small Font button, the **onClick()** event handler captures the event and sets text size of the TextView to **10** as shown in below figure:



4.1 Event Driven Programming in Android

2. Listener interface (Example)

- When user clicks on Large Font button, the **onClick()** event handler captures the event and sets text size of the TextView to **24** as shown in below figure :

Practical – 6

Create login application where you will have to validate usename and passwords Till the username and password is not validated , login button should remain disabled.

MainActivity.java

```
public class MainActivity extends ActionBarActivity
 {
      EditText et1 ;
      EditText et2 ;
      Button bt1;
       @Override
       protected void onCreate(Bundle savedInstanceState)
       {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.activity_main);

            et1 =  (EditText) findViewById(R.id.editText1);
            et2 = (EditText) findViewById(R.id.editText2);
            bt1= (Button) findViewById(R.id.button1);
```

```
et1.setOnKeyListener(new View.OnKeyListener()
{
        @Override
        public boolean onKey(View arg0, int arg1, KeyEvent arg2)
        {
                String username = et1.getText().toString();
                String password = et2.getText().toString();

                if(username.compareTo("Mayur")==0 &&
                        password.compareTo("Thakkar") ==0 )
                                bt1.setEnabled(true);
                 else
                                bt1.setEnabled(false);

                return false;
                }});
```

```
et2.setOnKeyListener(new View.OnKeyListener()
{
        @Override
        public boolean onKey(View arg0, int arg1, KeyEvent arg2)
        {
                String username = et1.getText().toString();
                String password = et2.getText().toString();

                if(username.compareTo("Mayur")==0 &&
                        password.compareTo("Thakkar") ==0 )
                                bt1.setEnabled(true);
                 else
                                bt1.setEnabled(false);

                return false;
                }});
```

****