EVENT HANDLING

Event handling:-

* It is the mechanism that control the event and decide what should happen if event accur this mechanism have the code which is know as event handling that is executed when an event accur.

There are 3 concepts related to Android Management:-

1.Event Listeners

2.Event Listeners Registration

3.Event Handlers

Event Listeners Using An Onclick in XML Attribute and setOnclickListener in JAVA:-

* Onclick =Using the android:onclick XML attribute Where you just use the name of a public method with the signature Void name (View v) .

Example:-

* To display a button in an activity, add a button to the activity's layout XML file.

<Button

android:layout\_height="wrap\_content"

android:layout\_width="wrap\_content"

android:text=”clickt"

android:onclick=”fun”

/>

* In main activity we call a button through (fun )method out side the onCreate method .

public class MyActivity extends Activity {

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.sample\_main);

}

public void fun(View v) {

//Add some logic

}

}

setOnclickListener interface :-

* when your Activity class implements the Listener interface and you put the handler method in the main Activity and then you call setonclicklistener(this).
* when Using the setOnclickListener method Where you pass an object that implement the onclickListener interface .

Example:-

* To display a button in an activity, add a button to the activity's layout XML file

<Button

android:id="@+id/button\_id"

android:layout\_height="wrap\_content"

android:layout\_width="wrap\_content"

android:text="@string/self\_destruct"

/>

* To specify an action when the button is pressed, set a click listener on the button object in the corresponding activity code

public class MyActivity extends Activity {

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.content\_layout\_id);

final Button button = findViewById(R.id.button\_id);

button.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

myFancyMethod(v);

}

});

}

}