ASSIGNMENT - 4

Q. Explain Event in Brief.

→ It Represent a Response to user interaction with input Controls Such as press of a Button or touch of the Screen.

Android Framework Places each occurring Event into a queue, which is Based on FIFO logic.

when an Event happens the Event listener, which is involved with the View Object needs to be Registered. Then, the Registered Event listener needs to be implement a Corresponding Callback method Event Handler.

Event Listener →

It is the Object that Receives notification when an Event Occurs.

Event Listener Registrations.

It 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 and event listener for the event the listener calls the Handlers which is the method that actually handles the event.

2) Explain Event Driven Programming in Brief

It is a process in which the flow of the

program is Determined by events such as user actions, System events or message from other programs

It is used to handle user instruction with GUI of an Application.

It is Structured around event handlers which are functions that are triggered when a specific event occurs

Android provides a no. of Built in event handlers that developers can use to Respond to user Interaction.

OnClick Listener -

Use to handle click events on Buttons

On Touch Listener -

Use to handle touch events such as when a user touches and drag a view.

OnlongClick Listener :-
Handle long Click events.

Onfocus Changed Listener
This is used to handle events when a view gains or loses focus.

Developers Can also Create their own Custom event handlers. to Respond to events that are not provided by the Android System.

(9) List Different ways of event handling techniques Explain long click Event with Example.

→ The most commonly used methods
1. Using an Anonymous Inner class
2. Activity class Implements the Listener Interface.

3. Using Layout file activity_main.xml to specify event handler Directly.

* onLongClickListener implements one method onLongClick() to handle long click event.

It is used when user touches and hold the control for more than one second.

Eg:-

```
<TextView
android:layout_width = "wrap_content"
android:layout_height = "wrap_content"
android:id = "@+id/t1" />

<EditText
android:layout_width = "wrap_content"
android:layout_height = "wrap_content"
android:id = "@+id/editText"
android:layout_below = "@+id/t1"
android:layout_toRightOf = "@+id/t1"
android:layout_toEndOf = "@+id/t1" />
```

2 Button.

android:layout_width = "wrap_content"
android:layout_height = "wrap_content"
android:text = "Show data".
android:id = "@+id/b1"
android:layout_below = "@+id/editText"
android:layout_marginTop = "62dp"
/>.

Java.

Button button = (Button) findViewById
(R.id.b1);
button.setOnLongClickListener(
new
Button.OnLongClickListener() {

public boolean onLongClick(View v)
{
EditText et = (EditText)
findViewById(R.id.editText);

android.widget.TextView tv =
(android.widget.TextView)
findViewById(R.id.t1)
tv.setText("long Data" + et.
getText());

return true ; }
}
);

4. Create an Android Application to Demonstrate Text Change Event.

<?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:padding = "16dp" >

<EditText
android:id = "@+id/edit_text"
android:layout_width = "match_parent"
android:layout_height = "wrap_content"
android:hint = "Type Something" />

<TextView
android:id = "@+id/textview"
android:layout_width = "match_parent"
android:layout_height = "wrap_content"
android:marginTop = "16dp"
android:textStyle = "bold" />

// Linear layout >

Java File.

```
import android.os.Bundle;
import android.text.Editable;
import androidx.appcompat.app.AppCompatActivity;

Public class MainActivity extends AppCompatActivity {

    private EditText editText;
    private TextView textView;

    @Override

    protected void onCreate(Bundle savedInstanceState) {
```

<Textview
android : id = "@ + id / text.view"
android : layout_width = "match - Parent"
android : layout_height = "wrap_content"
android : marginTop = "16dP"
android : textstyle = "bold" />

<//Linear layout>

Java file

import android.os. Bundle ;
import android.text. Editable ;
import android.x . appcompact . app. APP CompatActivity

Public class MainActivity extends
APPCompat Activity {

Private EditText editText ;
Private Textview textview ;

@override
Protected void Oncreate (Bundle
SavedInstance state) {

SuPar. onCreate (Saved InstanceState) ;
set ContentView (R. layout . activity_main) ;

edittext = findview byId (R. id . edit_text) ;
textview = findviewbyId (R. id . text_view) ;

editText.addTextChangedListener(new
TextWatcher()
{

@Override
Public void onTextchanged(Char sequence
s, intstart, int before, int count){

textview setText("Text changed to:"+s);

}
});
}
}

Q:5 Create an andoid Application to Demonstrate Button click Event.

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

< Relative layout xmlns:android="https://schemas.
android.com/apk/res/android"

<xmlns:tools="https://schemas.android.com
tools"
android:id="@id/activity_main"
android:layout_width="match_Parent"
android:layout_height="match_Parent"
tools:Context="".mainActivity">

<Button
android : id = "@+id / bt"
android : layout_width = "wrap_content"
android : layout_height = "wrap_content"
android : text = "click me!"
android : layout_centerInParent = "true" />

</Relative layout>

Java file

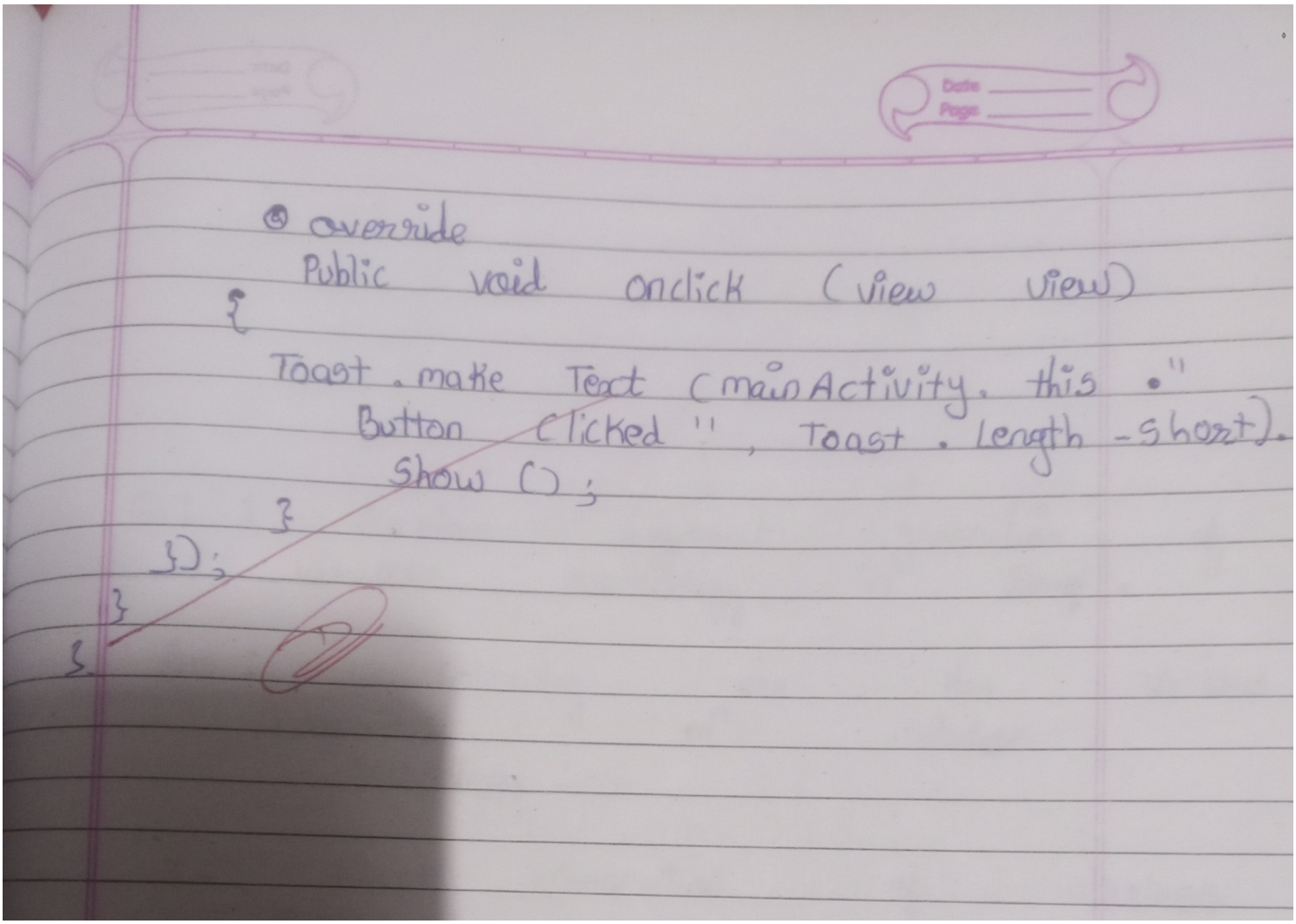
```
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;
import android.appcompat.app.AppCompatActivity;

Public class mainActivity Extends
    APPcompat Activity {

@ override
    Prodcated void onCreat (Bundle saved
        Instancestate) {

super.onCreate (Saved Instancestate);
set Contentview (R.layout.activity_main);

Button button = find ViewById (R.id.
                    button);
button.set onclick Listener (new
    view.onclick Listener () {
```


@override
Public void onclick (View view)
{
Toast.makeText(MainActivity.this,"
Button Clicked", Toast.length_short).
show();
}
});
}
}