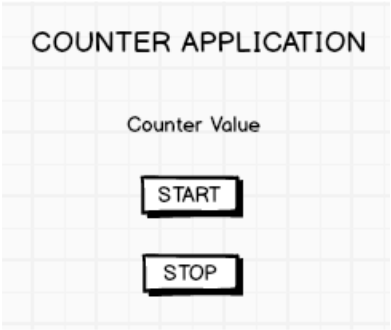
LAB PROGRAM 5

Write a program to create an activity with two buttons START and STOP. On pressing of the START button, the activity must start the counter by displaying the numbers from One and the counter must keep on counting until the STOP button is pressed. Display the counter value in a TextViewcontrol



COUNTER APPLICATION ..> TEXTVIEW

COUNTER Value --→ Text View ID: txt_value

Start -→ Button ID: btn_start

Stop →Button , ID: btn_stop

```
package com.example.counterpgm5;

import static java.lang.Thread.currentThread;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.os.Handler;
import android.os.Looper;
import android.os.Message;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity implements
View.OnClickListener {
    Button buttonStart, buttonStop;
     TextView counterValue;
     public int counter = 0;
     public boolean running = false;
```

```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        buttonStart = (Button)
findViewById(R.id.btn_start);

buttonStart.setOnClickListener(this);
        buttonStop = (Button)
findViewById(R.id.btn_stop);

buttonStop.setOnClickListener(this);


counterValue = (TextView)
findViewById(R.id.txt_value);

    }

    @Override
    public void onClick(View v) {
        if (v.equals(buttonStart)) {
                counterstart();
            } else if (v.equals(buttonStop))
{
                counterStop();
            }
    }

    private void counterStop() {
        this.running = false;
            buttonStop.setEnabled(false);
            buttonStart.setEnabled(true);
    }
```

```
private void counterstart() {
      counter = 0;
         running = true;
         System.out.println("start ->" +
Thread.currentThread().getName());
         new MyCounter().start();
         buttonStart.setEnabled(false);
         buttonStop.setEnabled(true);
    }

   Handler handler = new
Handler(Looper.getMainLooper()) {
    public void handleMessage(Message mes)
{

counterValue.setText(String.valueOf(mes.what));
         }
     };

    class MyCounter extends Thread {
         public void run() {
         System.out.println("Mycounter ->"
+ Thread.currentThread().getName());
              while (running) {
                  counter++;

handler.sendEmptyMessage(counter);
                  try {
                      Thread.sleep(1000);
                  } catch (Exception e) { }
          }
       }
    }
}
```