***TASK # 01***

*Implement handler*

***CODE***

***XML***

*<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 tools:context=".MainActivity"  
 android:orientation="vertical"  
 android:paddingTop="50dp">  
  
 <Button  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="New Thread"  
 android:onClick="newThreadState"  
 android:layout\_gravity="center\_horizontal"  
 android:background="@color/colorPrimary"  
 android:textColor="#ffffff"/>  
  
 <Button  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Blocked Thread"  
 android:onClick="blockedThreadState"  
 android:layout\_gravity="center\_horizontal" />  
  
 <Button  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Runnable Thread"  
 android:onClick="runnableThreadState"  
 android:layout\_gravity="center\_horizontal"  
 android:background="@color/colorAccent"  
 android:textColor="#ffffff"/>  
  
 <Button  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Timed\_Waiting Thread"  
 android:onClick="timededWatingThreadState"  
 android:layout\_gravity="center\_horizontal"/>  
  
 <Button  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Waiting Thread"  
 android:onClick="waitingThreadState"  
 android:layout\_gravity="center\_horizontal"  
 android:background="@color/colorPrimary"  
 android:textColor="#ffffff"/>  
  
 <Button  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Terminated Thread"  
 android:onClick="terminatedThreadState"  
 android:layout\_gravity="center\_horizontal"/>  
  
 <TextView  
 android:id="@+id/txt"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:scrollbars = "vertical"  
 android:textSize="10sp" />  
</LinearLayout>*

***JAVA***

*package com.example.thread\_assignment;*

*import android.support.v7.app.AppCompatActivity;*

*import android.os.Bundle;*

*import android.text.method.ScrollingMovementMethod;*

*import android.util.Log;*

*import android.view.View;*

*import android.widget.TextView;*

*import org.w3c.dom.Text;*

*public class MainActivity extends AppCompatActivity {*

*public static String msg = "";*

*TextView textview;*

*@Override*

*protected void onCreate(Bundle savedInstanceState) {*

*super.onCreate(savedInstanceState);*

*setContentView(R.layout.activity\_main);*

*textview = (TextView)findViewById(R.id.txt);*

*textview.setMovementMethod(new ScrollingMovementMethod()); }*

*public static final String TAG = "Thread Status ";*

*public void newThreadState(View view) {*

*Thread01 th1 = new Thread01();*

*msg = (textview.getText().toString() + TAG + th1.getState().toString() + "\n");*

*textview.setText(msg); }*

*public void runnableThreadState(View view) {*

*Thread01 th1 = new Thread01(); th1.start();*

*msg = (textview.getText().toString() + TAG + th1.getState().toString() + "\n");*

*textview.setText(msg);*

*public void timededWatingThreadState(View view) {*

*Thread01 th1 = new Thread01();*

*th1.start();*

*try {*

*th1.sleep(2000);*

*} catch (InterruptedException e) {*

*e.printStackTrace();*

*}*

*msg = (textview.getText().toString() + TAG + th1.getState().toString() + "\n");*

*textview.setText(msg);*

*}*

*public static Thread03 th;*

*public void waitingThreadState(View view) throws InterruptedException {*

*th = new Thread03();*

*th.start();*

*msg = (textview.getText().toString() + TAG + msg + "\n");*

*textview.setText(msg); }*

*public static String data = "Group";*

*public void blockedThreadState(View view) {*

*Thread02 th=new Thread02(data);*

*synchronized (data) {*

*th.start();*

*try {*

*Thread.sleep(100);*

*} catch (InterruptedException e) {*

*Log.e(TAG, e.getMessage()); }*

*msg = (textview.getText().toString() + TAG + th.getState().toString() + "\n");*

*textview.setText(msg); } }*

*public void terminatedThreadState(View view) {*

*Thread th1 = new Thread();*

*th1.start();*

*try {*

*th1.join();*

*} catch (InterruptedException e) {*

*e.printStackTrace(); }*

*textview.setText(textview.getText().toString() + TAG + th1.getState().toString() + "\n"); }}*

*Thread01*

*package com.example.thread\_assignment;  
import android.util.Log;*

*public class Thread01 extends Thread{*

*@Override*

*public void run() {  
 try {*

*for (Integer i = 1; i <= 10; i++) {*

*Log.d("Iteration ", i.toString());*

*Thread.sleep(1000); }*

*}catch(InterruptedException e){*

*Log.e(MainActivity.TAG, e.getMessage()); }*

*synchronized (MainActivity.th){*

*Log.i(MainActivity.TAG, MainActivity.th.getState().toString()); } }}*

*Thread 2:*

*package com.example.thread\_assignment;*

*import android.util.Log;*

*public class Thread02 extends Thread{*

*String data;*

*public Thread02(String data){*

*this.data = data; }*

*public Thread02(){ }*

*@Override*

*public void run() {*

*synchronized (data) {*

*try {*

*for (int k = 0; k < data.length(); k++) {*

*Log.d("Iteration ", data.charAt(k) + "");*

*Thread.sleep(100); }*

*} catch (InterruptedException e) {*

*Log.e(MainActivity.TAG, e.getMessage());} } }}*

*Thread 03:*

*package com.example.thread\_assignment;*

*public class Thread03 extends Thread{*

*Thread03 th;*

*@Override*

*public void run() {*

*th = MainActivity.th;*

*Thread04 t = new Thread04();*

*synchronized (th){  
 try {*

*t.start();*

*th.wait();*

*} catch (InterruptedException e) {*

*e.printStackTrace(); } } }}*

*Thread 04 :*

*package com.example.thread\_assignment;  
import android.util.Log;*

*public class Thread04 extends Thread{  
 @Override*

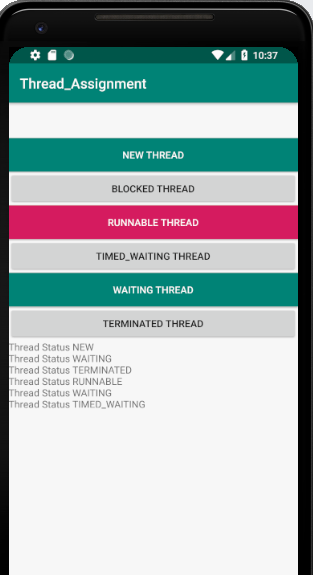
*public void run() {*

*Thread03 th = MainActivity.th;*

*synchronized (th){*

*MainActivity.msg = th.getState().toString(); } }}*

***OUTPUT***

**