**[2CEIT5PE5: MOBILE APPLICATION DEVELOPMENT]**

Practical: 2



AIM: TextView and its properties, Toast Message, Snackbar Message, Android in-built

resources like color, Activity Life Cycle, Log Message in logcat, Properties of Layout

ConstraintLayout, Generate **id**of TextView.

Submitted By: Kanzariya Dhavanik

Enrollment number: 21012022022



**Department of Computer Engineering/Information Technology**

* **Create Activity to display “Hello World” in**[**TextView**](https://developer.android.com/guide/topics/ui/look-and-feel/autosizing-textview)**in the center of Activity screen with yellow (android:background="#FFFF00") layout background of Activity.**[**TextView**](https://developer.android.com/guide/topics/ui/look-and-feel/autosizing-textview)**properties should be Holo Blue color (android:textColor=@android:color/holo\_blue\_bright), 27sp font size (android:textSize="27sp") and make text style bold and italic (android:textStyle="bold|italic"). After creating Activity, demonstrate functions of**[**Activity Life Cycle**](https://developer.android.com/guide/components/activities/activity-lifecycle)**by using**[**Log**](https://developer.android.com/studio/debug/am-logcat)**message,**[**Toast**](https://developer.android.com/guide/topics/ui/notifiers/toasts)**message &**[**Snackbar**](https://developer.android.com/training/snackbar/showing)**message. All methods of Activity life cycle should be printed in**[**Logcat**](https://developer.android.com/studio/debug/am-logcat)**.**

**Activity\_main.xml:**

***<?*xml version="1.0" encoding="utf-8"*?>*<androidx.constraintlayout.widget.ConstraintLayout**

**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:background="#ffff00"  
 android:id="@+id/mycl">  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Hello World!"  
 android:id="@+id/openActivity"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
 app:layout\_constraintEnd\_toEndOf="parent"  
 app:layout\_constraintStart\_toStartOf="parent"  
 app:layout\_constraintTop\_toTopOf="parent"  
 android:textColor="@color/holo\_blue\_bright"  
 android:textSize="27sp"  
 android:textStyle="bold|italic"/>  
  
</androidx.constraintlayout.widget.ConstraintLayout>**

**MainActivity.kt**

**package com.example.madpractical2\_21012022022  
  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.util.Log  
import android.widget.Toast  
import androidx.constraintlayout.widget.ConstraintLayout  
import com.google.android.material.snackbar.Snackbar  
  
class MainActivity : AppCompatActivity() {  
 private val TAG = "MainActivity"  
 private lateinit var mycl:ConstraintLayout  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.activity\_main)  
 mycl=findViewById(R.id.mycl)  
 Log.i(TAG, "onCreate: function called")  
 Toast.makeText(this,"onCreate: function**

**called",Toast.LENGTH\_LONG).show()  
 Snackbar.make(mycl,"onCreate: function called",**

**Snackbar.LENGTH\_LONG).show()  
 }  
 override fun onStart() {  
 Log.i(TAG, "onStart: function called")  
 Toast.makeText(this,"onStart: function**

**called",Toast.LENGTH\_LONG).show()  
 Snackbar.make(mycl,"onStart: function called",**

**Snackbar.LENGTH\_LONG).show()  
 super.onStart()  
  
 }  
  
 override fun onRestart() {  
 super.onRestart()  
 Log.i(TAG, "onRestart: function called")  
 Toast.makeText(this,"onRestart: function**

**called",Toast.LENGTH\_LONG).show()**

**Snackbar.make(mycl,"onRestart: function called",**

**Snackbar.LENGTH\_LONG).show()  
 }  
  
 override fun onResume() {  
 super.onResume()  
 Log.i(TAG, "onResume: function called")  
 Toast.makeText(this,"onResume: function**

**called",Toast.LENGTH\_LONG).show()  
 Snackbar.make(mycl,"onResume: function called",**

**Snackbar.LENGTH\_LONG).show()  
 }  
  
 override fun onPause() {  
 super.onPause()  
 Log.i(TAG, "onPause: function called")  
 Toast.makeText(this,"onPause: function**

**called",Toast.LENGTH\_LONG).show()  
 Snackbar.make(mycl,"onPause: function called",**

**Snackbar.LENGTH\_LONG).setDuration(10000)  
 }  
  
 override fun onStop() {  
 super.onStop()  
 Log.i(TAG, "onStop: function called")  
 Toast.makeText(this,"onStop: function**

**called",Toast.LENGTH\_LONG).show()  
 Snackbar.make(mycl,"onStop: function called",**

**Snackbar.LENGTH\_LONG).show()  
 }  
  
 override fun onDestroy() {  
 super.onDestroy()  
 Log.i(TAG, "onDestroy: function called")  
 Toast.makeText(this,"onDestroy: function**

**called",Toast.LENGTH\_LONG).show()  
 Snackbar.make(mycl,"onDestroy: function called",**

**Snackbar.LENGTH\_LONG).show()  
 }  
}**

**Output:**





