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

Practical: 2



**AIM-**Create Activity to display “Hello World” in **[TextView](https://developer.android.com/guide/topics/ui/look-and-feel/autosizing-textview" \t "_blank)** 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" \t "_blank)** 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" \t "_blank)** message. All methods of Activity life cycle should be printed in [**Logcat**](https://developer.android.com/studio/debug/am-logcat)

**.**

Submitted By: Pavan Patel

Enrollment number:21012021075



**Department of Computer Engineering/Information Technology**

**Practical-2**

**Create a “Hello World” application which has a yellow layout background with red color,20sp font size and display text on the center of the screen .**

**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"

android:background="#FFFF00"

tools:context=".MainActivity">

<androidx.coordinatorlayout.widget.CoordinatorLayout

xmlns:android="http://schemas.android.com/apk/res/android"

android:id="@+id/myCoordinatorLayout"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<!-- Add your other views here -->

</androidx.coordinatorlayout.widget.CoordinatorLayout>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:background="#FFFF00"

android:text="Hello World!"

android:textColor="@android:color/holo\_blue\_bright"

android:textSize="27sp"

android:textStyle="bold|italic"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

**MainActivity.kt**

package com.pavanpatel.p2\_75\_5b6

import androidx.appcompat.app.AppCompatActivity

import android.os.Bundle

import android.util.Log

import android.widget.Toast

import com.google.android.material.snackbar.Snackbar

import androidx.coordinatorlayout.widget.CoordinatorLayout

import org.w3c.dom.Text

class MainActivity : AppCompatActivity() {

val TAG = "MainActivity"

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

displayMessage("onCreate")

}

override fun onStart() {

super.onStart()

displayMessage("onStart")

showSnackbar()

}

override fun onResume() {

super.onResume()

displayMessage("onResume")

}

override fun onPause() {

super.onPause()

displayMessage("onPause")

}

override fun onStop() {

super.onStop()

displayMessage("onStop")

}

override fun onDestroy() {

super.onDestroy()

displayMessage("onDestroy")

}

fun displayMessage(msg: String) {

Toast.makeText(this, "This is a $msg", Toast.LENGTH\_LONG).show()

Log.i(TAG, "$msg" + "is called")

}

private fun showSnackbar() {

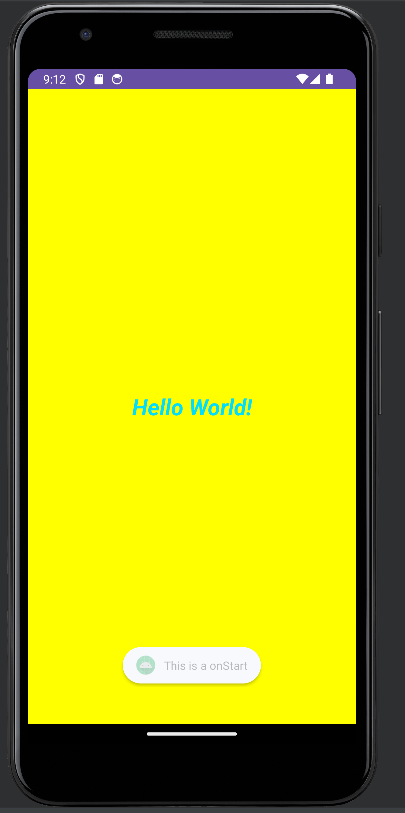
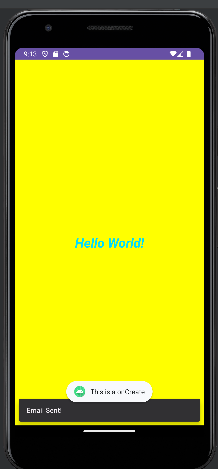
val myCoordinatorLayout = findViewById<CoordinatorLayout>(R.id.myCoordinatorLayout)

Snackbar.make(myCoordinatorLayout, R.string.email\_sent, Snackbar.LENGTH\_SHORT).show()

}

}

**Output:**

****