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

Practical: 2



**AIM-** Create Activity to display “Hello World” in TextView in the center of

Activity screen with yellow (android:background="#FFFF00") layout

background of Activity. 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 by using Log message,

Toast message & Snackbar message. All methods of Activity life cycle

should be printed in Logcat..

Submitted By: Honey Patel

Enrollment number: 20012021026



**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:id="@+id/mycl"

tools:context=".MainActivity"

android:background="#FFFF00">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Hello World!"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

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

android:textSize="27sp"

android:textStyle="bold|italic"/>

</androidx.constraintlayout.widget.ConstraintLayout>

**MainActivity.kt:**

package com.example.madpractical2\_20012021026

import androidx.appcompat.app.AppCompatActivity

import android.os.Bundle

import android.util.Log

import android.widget.Toast

import androidx.constraintlayout.widget.ConstraintLayout

import androidx.constraintlayout.widget.ConstraintSet

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: onCreate function called")

Toast.makeText(this,"onCreate: onCreate function called",Toast.LENGTH\_LONG).show()

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

}

override fun onStart() {

super.onStart()

Log.i(TAG, "onStart: onStart function called")

Toast.makeText(this,"onStart: onStart function called",Toast.LENGTH\_LONG).show()

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

}

override fun onResume() {

super.onResume()

Log.i(TAG, "onResume: onResume function called")

Toast.makeText(this,"onResume: onResume function called",Toast.LENGTH\_LONG).show()

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

}

override fun onPause() {

super.onPause()

Log.i(TAG, "onPause: onPause function called")

Toast.makeText(this,"onPause: onPause function called",Toast.LENGTH\_LONG).show()

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

}

override fun onStop() {

super.onStop()

Log.i(TAG, "onStop: onStop function called")

Toast.makeText(this,"onStop: onStop function called",Toast.LENGTH\_LONG).show()

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

}

override fun onDestroy() {

super.onDestroy()

Log.i(TAG, "onDestroy: onDestroy function called")

Toast.makeText(this,"onDestroy: onDestroy function called",Toast.LENGTH\_LONG).show()

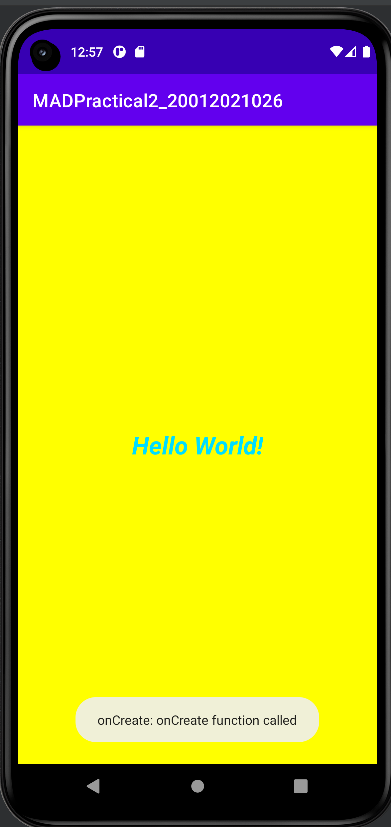
Snackbar.make(mycl,"onDestroy: onDestroy function called",Snackbar.LENGTH\_LONG).show()

}

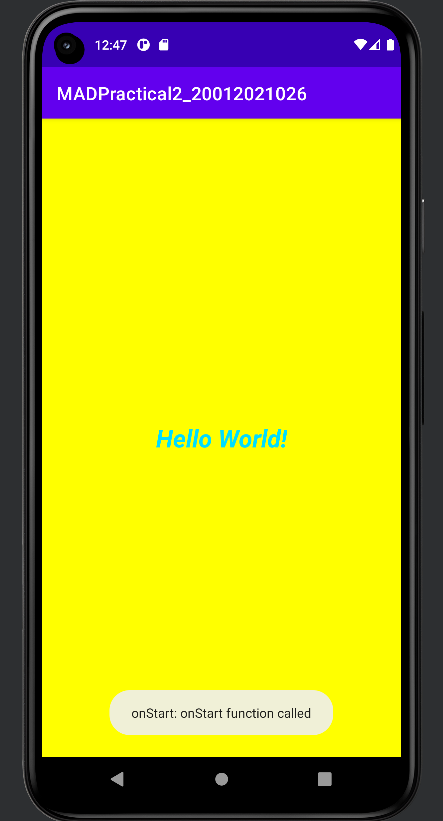
}

**Output:**

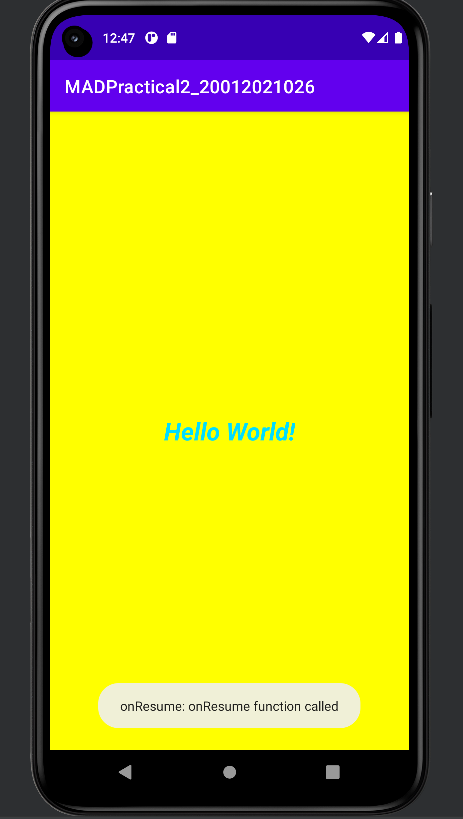
On Crate:

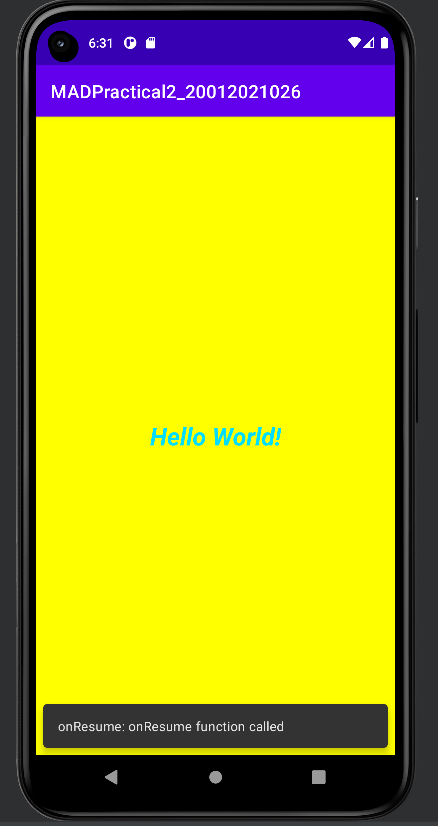


 On Start:

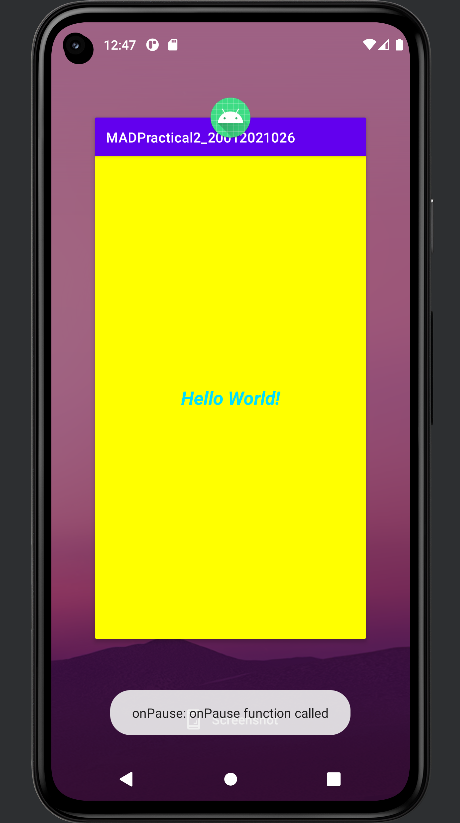


On Resume:

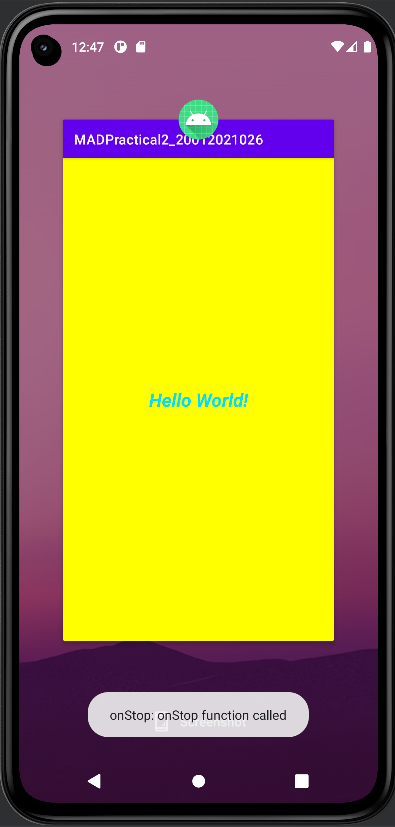




On pause:



On stop:



On destroy:



Logcat:

