**UCS6611 - MOBILE APPLICATION DEVELOPMENT LABORATORY**

**Ex. No. 5 implement an application that uses multi-threading**

**Date:** 9/9/21 **Name:** Srinath S

**Class:** CSE-C  **Roll:** 185001205

**Question:**

implement an application that uses multi-threading

**Code:**

**ActivityMain.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">

        <Button

            android:id="@+id/button"

            android:layout\_width="wrap\_content"

            android:layout\_height="wrap\_content"

            android:layout\_marginBottom="500dp"

            android:text="Start"

            app:layout\_constraintBottom\_toBottomOf="parent"

            app:layout\_constraintEnd\_toEndOf="parent"

            app:layout\_constraintHorizontal\_bias="0.500"

            app:layout\_constraintStart\_toStartOf="parent" />

        <ProgressBar

            android:id="@+id/progressBar"

            style="?android:attr/progressBarStyleHorizontal"

            android:layout\_width="300dp"

            android:layout\_height="33dp"

            app:layout\_constraintBottom\_toTopOf="@+id/button"

            app:layout\_constraintEnd\_toEndOf="parent"

            app:layout\_constraintHorizontal\_bias="0.496"

            app:layout\_constraintStart\_toStartOf="parent"

            app:layout\_constraintTop\_toTopOf="parent"

            app:layout\_constraintVertical\_bias="0.783" />

        <TextView

            android:id="@+id/textView"

            android:layout\_width="wrap\_content"

            android:layout\_height="wrap\_content"

            android:text="Initiate Task"

            app:layout\_constraintBottom\_toTopOf="@+id/progressBar"

            app:layout\_constraintEnd\_toEndOf="parent"

            app:layout\_constraintStart\_toStartOf="parent"

            app:layout\_constraintTop\_toTopOf="parent"

            app:layout\_constraintVertical\_bias="0.72" />

        <View

            android:id="@+id/divider"

            android:layout\_width="409dp"

            android:layout\_height="1dp"

            android:background="?android:attr/listDivider"

            app:layout\_constraintBottom\_toBottomOf="parent"

            app:layout\_constraintTop\_toTopOf="parent"

            app:layout\_constraintVertical\_bias="0.431"

            tools:layout\_editor\_absoluteX="1dp" />

        <TextView

            android:id="@+id/textView2"

            android:layout\_width="180dp"

            android:layout\_height="35dp"

            android:layout\_marginTop="68dp"

            android:layout\_marginEnd="24dp"

            android:text="Enter Sleep Time"

            app:layout\_constraintBottom\_toBottomOf="parent"

            app:layout\_constraintEnd\_toStartOf="@+id/input"

            app:layout\_constraintTop\_toBottomOf="@+id/divider"

            app:layout\_constraintVertical\_bias="0.0" />

        <EditText

            android:id="@+id/input"

            android:layout\_width="164dp"

            android:layout\_height="46dp"

            android:layout\_marginEnd="16dp"

            android:layout\_marginBottom="300dp"

            android:ems="10"

            android:inputType="textPersonName"

            app:layout\_constraintBottom\_toBottomOf="parent"

            app:layout\_constraintEnd\_toEndOf="parent" />

        <Button

            android:id="@+id/button3"

            android:layout\_width="wrap\_content"

            android:layout\_height="wrap\_content"

            android:layout\_marginTop="184dp"

            android:text="Sleep"

            app:layout\_constraintBottom\_toBottomOf="parent"

            app:layout\_constraintEnd\_toEndOf="parent"

            app:layout\_constraintHorizontal\_bias="0.498"

            app:layout\_constraintStart\_toStartOf="parent"

            app:layout\_constraintTop\_toBottomOf="@+id/divider"

            app:layout\_constraintVertical\_bias="0.098" />

</androidx.constraintlayout.widget.ConstraintLayout>

**Mainactivity.java:**

package com.example.progressbar;

import androidx.appcompat.app.AlertDialog;

import androidx.appcompat.app.AppCompatActivity;

import android.graphics.Color;

import android.os.Bundle;

import android.os.CountDownTimer;

import android.os.Handler;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.ProgressBar;

import android.widget.TextView;

public class MainActivity extends AppCompatActivity {

    int progress=0;

    ProgressBar progressBar;

    Handler hand = new Handler();

    Button start,reset,sleep;

    TextView value;

    EditText input;

    AlertDialog.Builder builder;

    AlertDialog alertDialog;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity\_main);

        progressBar = findViewById(R.id.progressBar);

        start = findViewById(R.id.button);

        sleep = findViewById(R.id.button3);

        value = findViewById(R.id.textView);

        input = findViewById(R.id.input);

        progressBar.setProgress(progress);

        builder=new AlertDialog.Builder(this);

        start.setOnClickListener(new View.OnClickListener() {

            @Override

            public void onClick(View v) {

                runnable.run();

            }

        });

        sleep.setOnClickListener(new View.OnClickListener() {

            @Override

            public void onClick(View view) {

                final int sleepTime = Integer.parseInt(input.getText().toString());

                System.out.println(sleepTime);

                builder.setTitle("Going to sleep for "+sleepTime+" seconds...").setMessage("Going to sleep for "+sleepTime+" seconds...");

                alertDialog=builder.create();

                alertDialog.show();

                new Thread(new Runnable() {

                    @Override

                    public void run() {

                        try {

                            Thread.sleep(sleepTime\*1000);

                        }

                        catch (Exception e) {

                            e.printStackTrace();

                        }

                        finally {

                            alertDialog.dismiss();

                        }

                    }

                }).start();

                new CountDownTimer(sleepTime\*1000, 1000) {

                    public void onTick(long millisUntilFinished) {

                        alertDialog.setMessage("Sleeping for another "+(millisUntilFinished/1000)+" seconds...");

                    }

                    public void onFinish() {

                        alertDialog.setMessage("Done!");

                    }

                }.start();

            }

        });

    }

    Runnable runnable = new Runnable() {

        @Override

        public void run() {

            try {

                doTask();

            }

            catch (Exception e) {

                e.printStackTrace();

            }

            finally {

                hand.postDelayed(runnable, 100);

            }

        }

    };

    private void doTask() {

        progress+=2;

        if(progress<=25){

            value.setText("Started");

            progressBar.getProgressDrawable().setColorFilter(Color.YELLOW, android.graphics.PorterDuff.Mode.SRC\_IN);

        }

        else if(progress<=80){

            value.setText("In Progress");

            progressBar.getProgressDrawable().setColorFilter(Color.RED, android.graphics.PorterDuff.Mode.SRC\_IN);

        }

        else if(progress<=90){

            value.setText("In Progress");

            progressBar.getProgressDrawable().setColorFilter(Color.BLUE, android.graphics.PorterDuff.Mode.SRC\_IN);

        }

        else if(progress==100){

            value.setText("Done");

        }

        if(progress>100) {

            hand.removeCallbacks(runnable);

        }

        else {

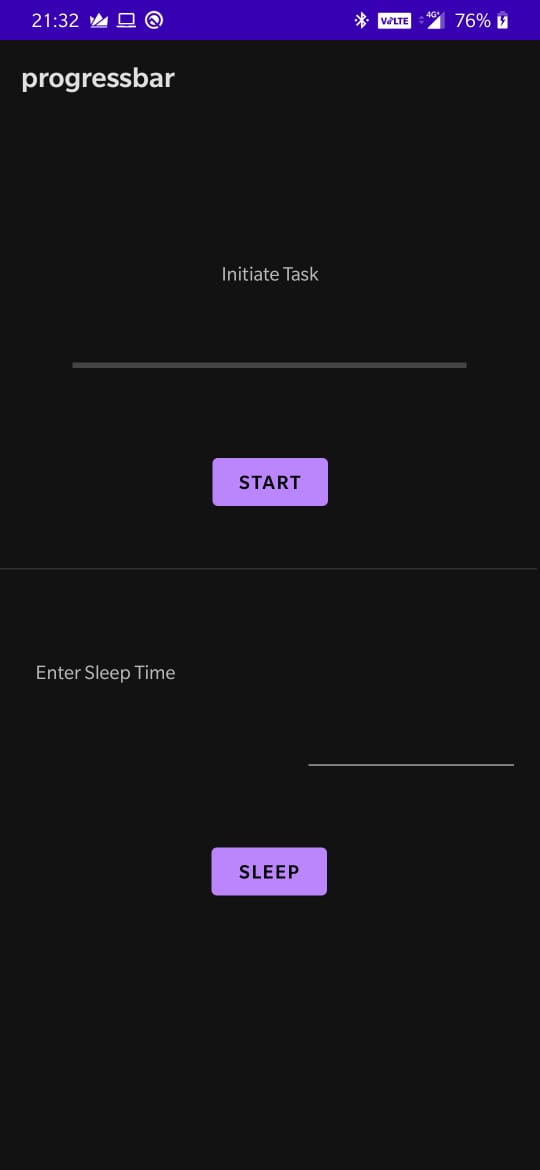
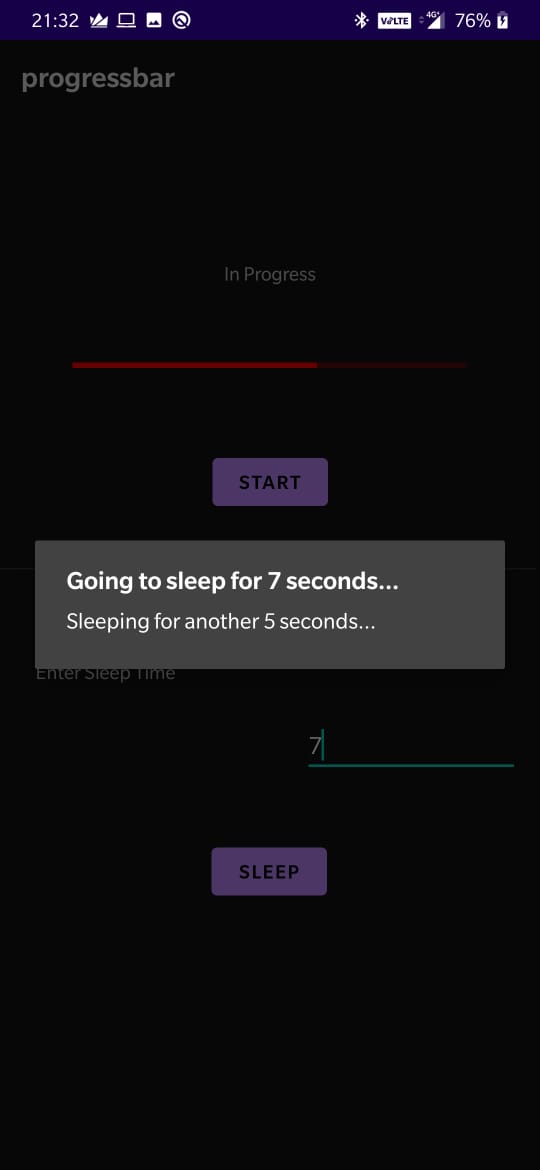
            progressBar.setProgress(progress);

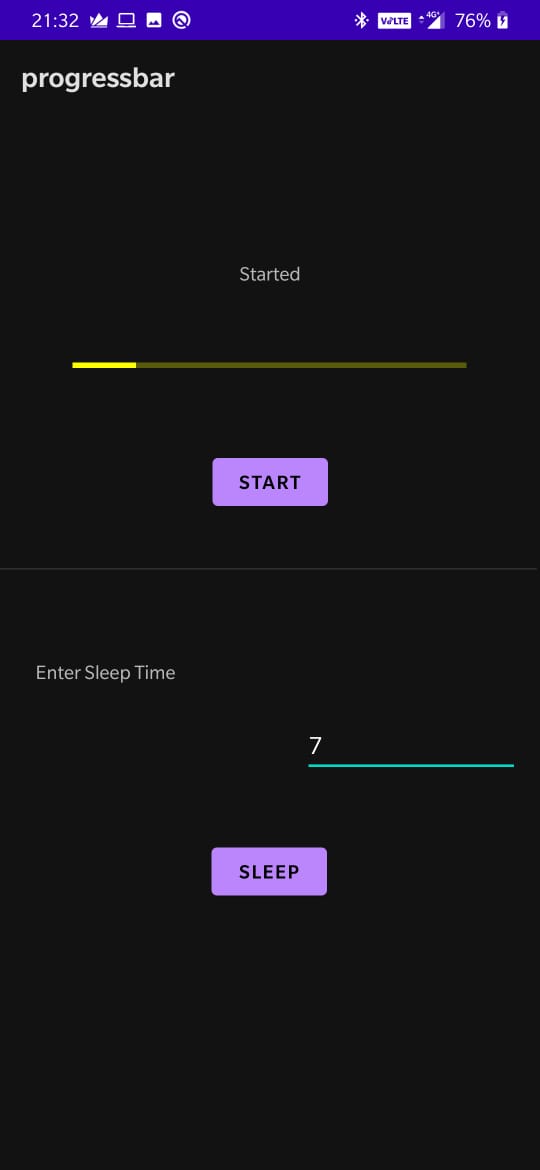
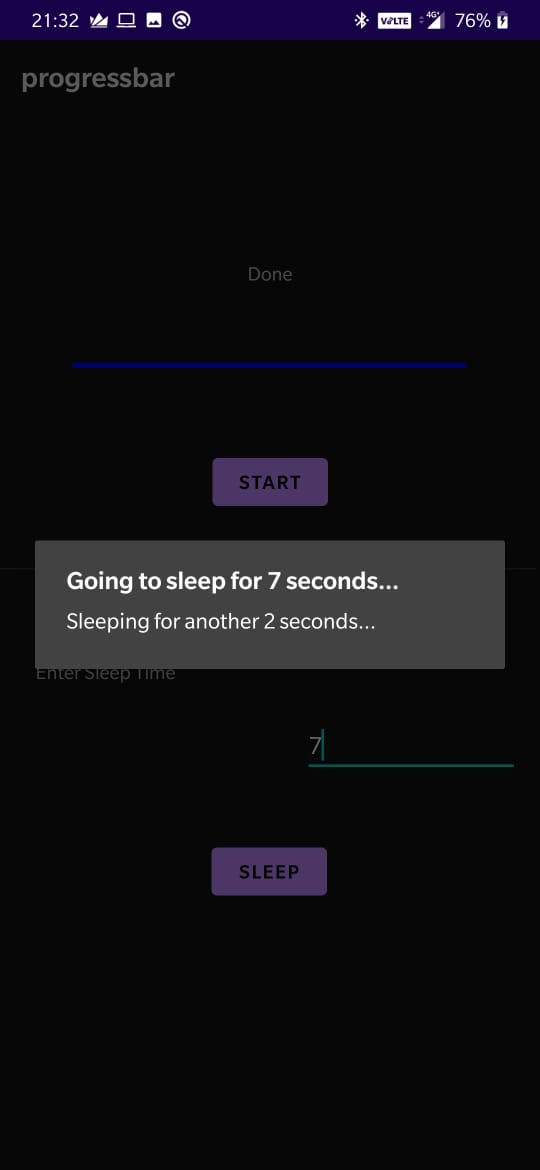
        }

    }

}

**Output:**

** **

** **