Practical - 22 : Write a program to demonstrate use of ProgressBar.

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">  
  
 <Button  
 android:id="@+id/button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentTop="true"  
 android:layout\_centerHorizontal="true"  
 android:text="download file"  
 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.java

package com.example.progressbar;  
  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.app.ProgressDialog;  
import android.os.Bundle;  
import android.os.Handler;  
import android.view.View;  
import android.widget.Button;  
  
public class MainActivity extends AppCompatActivity {  
  
 Button btnStartProgress;  
 ProgressDialog progressBar;  
 private int progressBarStatus = 0;  
 private Handler progressBarHandler = new Handler();  
 private long fileSize = 0;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 addListenerOnButtonClick();  
 }  
 public void addListenerOnButtonClick() {  
 btnStartProgress = findViewById(R.id.*button*);  
 btnStartProgress.setOnClickListener(new View.OnClickListener(){  
  
 @Override  
 public void onClick(View v) {  
 *// creating progress bar dialog* progressBar = new ProgressDialog(v.getContext());  
 progressBar.setCancelable(true);  
 progressBar.setMessage("File downloading ...");  
 progressBar.setProgressStyle(ProgressDialog.*STYLE\_HORIZONTAL*);  
 progressBar.setProgress(0);  
 progressBar.setMax(100);  
 progressBar.show();  
 *//reset progress bar and filesize status* progressBarStatus = 0;  
 fileSize = 0;  
  
 new Thread(new Runnable() {  
 public void run() {  
 while (progressBarStatus < 100) {  
 *// performing operation* progressBarStatus = doOperation();  
 try {  
 Thread.*sleep*(1000);  
 } catch (InterruptedException e) {  
 e.printStackTrace();  
 }  
 *// Updating the progress bar* progressBarHandler.post(new Runnable() {  
 public void run() {  
 progressBar.setProgress(progressBarStatus);  
 }  
 });  
 }  
 *// performing operation if file is downloaded,* if (progressBarStatus >= 100) {  
 *// sleeping for 1 second after operation completed* try {  
 Thread.*sleep*(1000);  
 } catch (InterruptedException e) {  
 e.printStackTrace();  
 }  
 *// close the progress bar dialog* progressBar.dismiss();  
 }  
 }  
 }).start();  
 }*//end of onClick method* });  
 }  
 *// checking how much file is downloaded and updating the filesize* public int doOperation() {  
 *//The range of ProgressDialog starts from 0 to 10000* while (fileSize <= 10000) {  
 fileSize++;  
 if (fileSize == 1000) {  
 return 10;  
 } else if (fileSize == 2000) {  
 return 20;  
 } else if (fileSize == 3000) {  
 return 30;  
 } else if (fileSize == 4000) {  
 return 40; *// you can add more else if* }  
 */\* else {  
 return 100;  
 }\*/* }*//end of while* return 100;  
 }  
}

Output:

