**Aim:** To create an application that displays the progress of an operation.

**Source code:**

**Activity\_main.xml:**

*<?***xml version="1.0" encoding="utf-8"***?>*<**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 xmlns:tools="http://schemas.android.com/tools"  
 tools:context="com.example.progressbar.MainActivity"**>  
 <**Button  
 android:id="@+id/button1"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentTop="true"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="116dp"  
 android:text="download file"** />  
</**RelativeLayout**>

**Mainactivity.java:**

**package** com.example.progressbar;  
  
**import** androidx.appcompat.app.AppCompatActivity;  
  
**import** android.app.ProgressDialog;  
**import** android.os.Handler;  
*//import android.support.v7.app.AppCompatActivity;***import** android.os.Bundle;  
**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\_main1***);  
 addListenerOnButtonClick();  
 }  
 **public void** addListenerOnButtonClick() {  
 **btnStartProgress** = findViewById(R.id.***button1***);  
 **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* }  
 }**return** 100;  
 }*//end of doOperation*}

 

