Practical No. 9

**Title: Android program to work with images and videos**

**Aim: Create an application to demonstrate images and videos components Introduction**

Android provides many ways to control playback of audio/video files and streams. One of this way is through a class called **MediaPlayer**.

Android is providing MediaPlayer class to access built-in mediaplayer services like playing audio,video e.t.c. In order to use MediaPlayer, we have to call a static Method **create()** of this class. This method returns an instance of MediaPlayer class. Its syntax is as follows −

# MediaPlayer mediaPlayer = MediaPlayer.create(this, R.raw.song);

The second parameter is the name of the song that you want to play. You have to make a new folder under your project with name **raw** and place the music file into it.

Once you have created the Mediaplayer object you can call some methods to start or stop the music. These methods are listed below.

**mediaPlayer.start(); mediaPlayer.pause();**

On call to **start()** method, the music will start playing from the beginning. If this method is called again after the **pause()** method, the music would start playing from where it is left and not from the beginning.

In order to start music from the beginning, you have to call **reset()** method. Its syntax is given below.

# mediaPlayer.reset();

Apart from the start and pause method, there are other methods provided by this class for better dealing with audio/video files. These methods are listed below −

|  |  |
| --- | --- |
| **Sr.No** | **Method & description** |
| 1 | **isPlaying()**  This method just returns true/false indicating the song is playing or not |

|  |  |
| --- | --- |
| 2 | **seekTo(position)**  This method takes an integer, and move song to that particular position millisecond |
| 3 | **getCurrentPosition()**  This method returns the current position of song in milliseconds |
| 4 | **getDuration()**  This method returns the total time duration of song in milliseconds |
| 5 | **reset()**  This method resets the media player |
| 6 | **release()**  This method releases any resource attached with MediaPlayer object |
| 7 | **setVolume(float leftVolume, float rightVolume)**  This method sets the up down volume for this player |
| 8 | **setDataSource(FileDescriptor fd)**  This method sets the data source of audio/video file |
| 9 | **selectTrack(int index)**  This method takes an integer, and select the track from the list on that particular index |
| 10 | **getTrackInfo()** |

|  |  |
| --- | --- |
|  | This method returns an array of track information |

**Exercise - Create android application to demonstrate images and videos components**

# Implementation:

**Program:**

# MainActivity.java

package com.example.maxpromediaplayer;

import androidx.appcompat.app.AppCompatActivity; import android.media.MediaPlayer;

import android.os.Bundle; import android.view.View; import android.widget.Button; import android.widget.SeekBar;

import java.io.IOException;

public class MainActivity extends AppCompatActivity { private Button play;

private Button pause; private SeekBar sb1; private MediaPlayer mp1;

@Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity\_main);

play=findViewById(R.id.button); pause=findViewById(R.id.button2); sb1=findViewById(R.id.seekBar);

//mp1=MediaPlayer.create(this,R.raw.inspiring);

//mp1.start();

mp1=new MediaPlayer(); try {

mp1.setDataSource("https:/[/www.soundheli](http://www.soundhelix.com/examples/mp3/SoundHelix-)x[.com/examples/mp3/SoundHelix-](http://www.soundhelix.com/examples/mp3/SoundHelix-) Song-1.mp3");

} catch (IOException e) { e.printStackTrace();

}

play.setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View view) { mp1.start();

}

});

pause.setOnClickListener(new View.OnClickListener() { @Override

public void onClick(View view) { mp1.pause();

}

});

mp1.setOnPreparedListener(new MediaPlayer.OnPreparedListener() { @Override

public void onPrepared(MediaPlayer mediaPlayer) { mediaPlayer.start(); sb1.setMax(mp1.getDuration()); sb1.setOnSeekBarChangeListener(new

SeekBar.OnSeekBarChangeListener() {

@Override

public void onProgressChanged(SeekBar seekBar, int i,

boolean b) {

}

});

if(b) {

mp1.seekTo(i);

}

}

@Override

public void onStartTrackingTouch(SeekBar seekBar) {

}

@Override

public void onStopTrackingTouch(SeekBar seekBar) {

}

});

mp1.prepareAsync();

}}

# 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"](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">

<TextView

android:id="@+id/textView" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_marginTop="156dp" android:shadowColor="#CD291D" android:text="MaxProMediaPlayer" android:textColor="#F42121" android:textSize="34sp"

app:layout\_constraintBottom\_toTopOf="@+id/imageView" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintStart\_toStartOf="parent" app:layout\_constraintTop\_toTopOf="parent" app:layout\_constraintVertical\_bias="0.0" />

<Button

android:id="@+id/button" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_marginBottom="160dp" android:text="Play" app:layout\_constraintBottom\_toBottomOf="parent" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintHorizontal\_bias="0.318" app:layout\_constraintStart\_toStartOf="parent" />

<SeekBar

android:id="@+id/seekBar" android:layout\_width="250dp" android:layout\_height="22dp" android:layout\_marginBottom="80dp" app:layout\_constraintBottom\_toTopOf="@+id/button" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintHorizontal\_bias="0.496" app:layout\_constraintStart\_toStartOf="parent" />

<ImageView

android:id="@+id/imageView" android:layout\_width="148dp" android:layout\_height="129dp" android:layout\_marginBottom="60dp" app:layout\_constraintBottom\_toTopOf="@+id/seekBar" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintHorizontal\_bias="0.498" app:layout\_constraintStart\_toStartOf="parent" app:srcCompat="@drawable/ad" tools:srcCompat="@drawable/ad" />

<Button

android:id="@+id/button2" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:text="Pause" app:layout\_constraintBaseline\_toBaselineOf="@+id/button" app:layout\_constraintEnd\_toEndOf="parent" app:layout\_constraintHorizontal\_bias="0.709" app:layout\_constraintStart\_toStartOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

# Output:

