Answers **(5)**	Coding Efficiency **(5)**	Viva **(5)**	Timely Completion **(5)**	Total **(20)**	Dated Sign of Subject Teacher

Expected Date of Completion:----------

Actual Date of Completion:----------

Experiment No: Group B-5

Problem Definition:

Design an android Application for Frame Animation.

5.1 Prerequisite:

Basics concepts of Java.

5.2 Learning Objective:

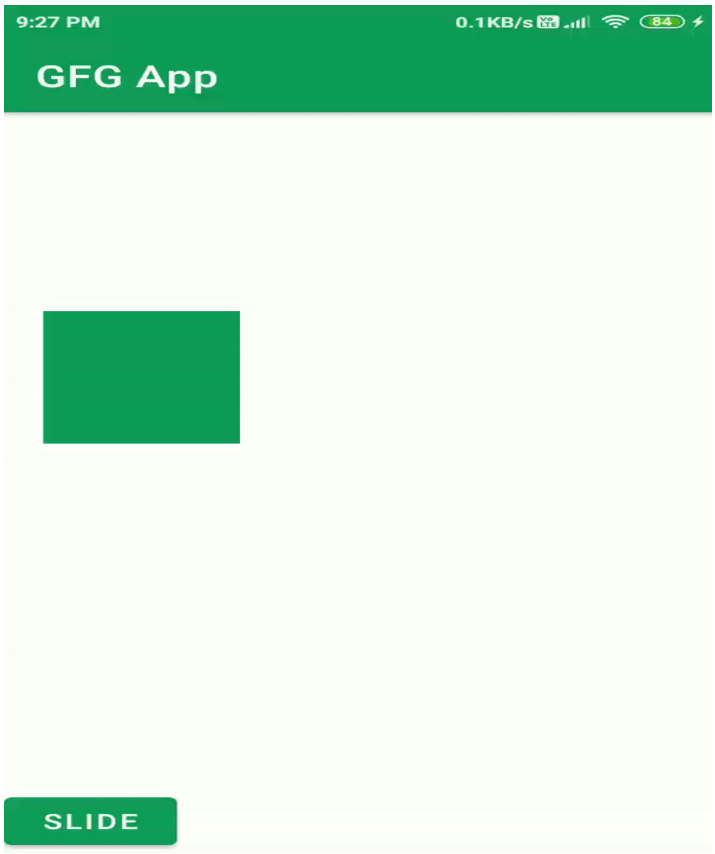
To make the animated videos.

5.3 Theory:

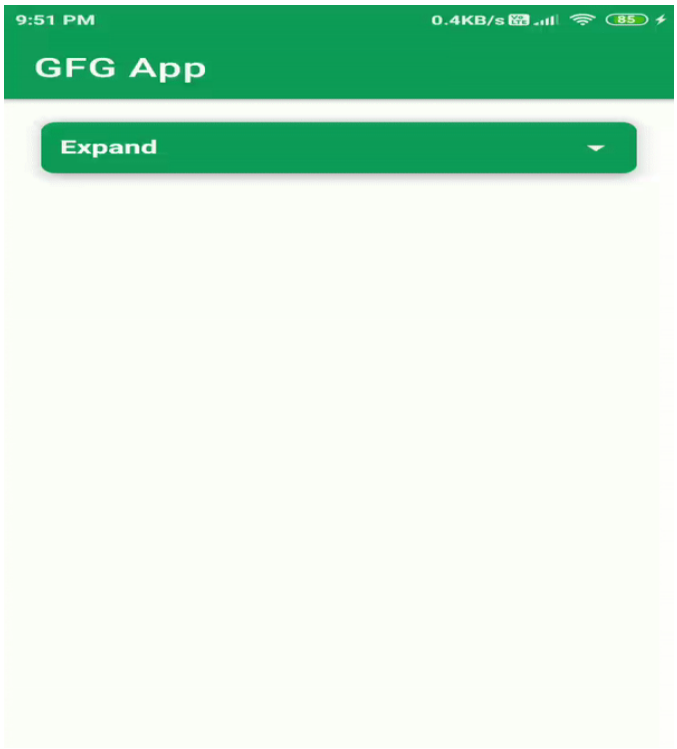
Animation is the process of adding a motion effect to any view, image, or text. With the help of an animation, you can add motion or can change the shape of a specific view. Animation in Android is generally used to give your UI a rich look and feel. The animations are basically of three types as follows:

1. Property Animation
2. View Animation
3. Drawable Animation

5.3.1 Property Animation: Property Animation is one of the robust frameworks which allows animating almost everything. This is one of the powerful and flexible animations which was introduced in Android 3.0. Property animation can be used to add any animation in the CheckBox, RadioButtons, and widgets other than any view.



5.3.2 View Animation: View Animation can be used to add animation to a specific view to perform tweened animation on views. Tweened animation calculates animation information such as size, rotation, start point, and endpoint. These animations are slower and less flexible. An example of View animation can be used if we want to expand a specific layout in that place we can use View Animation. The example of View Animation can be seen in Expandable RecyclerView.



5.3.3 Drawable Animation: Drawable Animation is used if you want to animate one image over another. The simple way to understand is to animate drawable is to load the series of drawable one after another to create an animation. A simple example of drawable animation can be seen in many apps Splash screen on apps logo animation.

5.3.4 Procedure:

Step 1: Create a New Project

To create a new project in Android Studio please refer to How to Create/Start a New Project in Android Studio. Note that select Java as the programming language.

Step 2: Working with the strings.xml file

Strings.xml can be found from the app > res > values > strings.xml. Below is the snippet for the strings.xml file.

```
<resources>
   <string name="app_name">GFG App</string>
   <string name="blink">BLINK</string>
   <string name="clockwise">ROTATE</string>
   <string name="fade">FADE</string>
   <string name="move">MOVE</string>
   <string name="slide">SLIDE</string>
   <string name="zoom">ZOOM</string>
   <string name="stop_animation">STOP ANIMATION</string>
   <string name="course_rating">Course Rating</string>
   <string name="course_name">Course Name</string>
</resources>
```

Step 3: Add google repository in the build.gradle file of the application project if by default it is not there

```
buildscript
{

repositories {

   google()

   mavenCentral()
```

```
}
```

All Jetpack components are available in the Google Maven repository, include them in the build.gradle file

```
allprojects {

 repositories {

   google()

  mavenCentral()

 }

}
```

Step 4: Working with the activity_main.xml file

Create ImageView in the activity_main.xml along with buttons that will add animation to the view. Navigate to the app > res > layout > activity_main.xml. Below is the code for the activity_main.xml file.

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
        xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:tools="http://schemas.android.com/tools"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:context=".MainActivity">

        <ImageView
                android:id="@+id/imageview"
```

```
        android:layout_width="200dp"
        android:layout_height="200dp"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="40dp"
        android:contentDescription="@string/app_name"
        android:src="@drawable/gfgimage" />

    <LinearLayout
        android:id="@+id/linear1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/imageview"
        android:layout_marginTop="30dp"
        android:orientation="horizontal"
        android:weightSum="3">

        <!--To start the blink animation of the image-->
        <Button
            android:id="@+id/BTNblink"
            style="@style/TextAppearance.AppCompat.Widget.Button"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_margin="10dp"
            android:layout_weight="1"
            android:padding="3dp"
            android:text="@string/blink"
            android:textColor="@color/white" />
```

```
        <!--To start the rotate animation of the image-->
        <Button
            android:id="@+id/BTNrotate"
            style="@style/TextAppearance.AppCompat.Widget.Button"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_margin="10dp"
            android:layout_weight="1"
            android:padding="3dp"
            android:text="@string/clockwise"
            android:textColor="@color/white" />

        <!--To start the fading animation of the image-->
        <Button
            android:id="@+id/BTNfade"
            style="@style/TextAppearance.AppCompat.Widget.Button"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_margin="10dp"
            android:layout_weight="1"
            android:padding="3dp"
            android:text="@string/fade"
            android:textColor="@color/white" />

</LinearLayout>
<LinearLayout
        android:id="@+id/linear2"
        android:layout_width="match_parent"
```

```
android:layout_height="wrap_content"
android:layout_below="@id/linear1"
android:layout_marginTop="30dp"
android:orientation="horizontal"
android:weightSum="3">

<!--To start the move animation of the image-->
<Button
        android:id="@+id/BTNmove"
        style="@style/TextAppearance.AppCompat.Widget.Button"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:layout_weight="1"
        android:padding="3dp"
        android:text="@string/move"
        android:textColor="@color/white" />

<!--To start the slide animation of the image-->
<Button
        android:id="@+id/BTNslide"
        style="@style/TextAppearance.AppCompat.Widget.Button"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:layout_weight="1"
        android:padding="3dp"
        android:text="@string/slide"
```

```
                android:textColor="@color/white" />

            <!--To start the zoom animation of the image-->
            <Button
                android:id="@+id/BTNzoom"
                style="@style/TextAppearance.AppCompat.Widget.Button"
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:layout_margin="10dp"
                android:layout_weight="1"
                android:padding="3dp"
                android:text="@string/zoom"
                android:textColor="@color/white" />

        </LinearLayout>

        <!--To stop the animation of the image-->
        <Button
            android:id="@+id/BTNstop"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_below="@id/linear2"
            android:layout_marginLeft="30dp"
            android:layout_marginTop="30dp"
            android:layout_marginRight="30dp"
            android:text="@string/stop_animation" />

</RelativeLayout>
```

Step 5: Create 6 different types of animation for ImageView

To create new animations we have to create a new directory for storing all our animations. Navigate to the app > res > Right-Click on res >> New >> Directory >> Name your directory as "anim". Inside this directory, we will create our animations. For creating a new anim right click on the anim directory >> Animation Resource file and give the name to your file. Below is the code snippet for 6 different animations.

1) Blink Animation

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">
    <alpha android:fromAlpha="0.0"
        android:toAlpha="1.0"
        android:interpolator="@android:anim/accelerate_interpolator"
        android:duration="500"
        android:repeatMode="reverse"
        android:repeatCount="infinite"/>
</set>
```

2) Fade Animation

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:interpolator="@android:anim/accelerate_interpolator">

    <!-- duration is the time for which animation will work-->
    <alpha
        android:duration="1000"
        android:fromAlpha="0"
        android:toAlpha="1" />
```

```
    <alpha
        android:duration="1000"
        android:fromAlpha="1"
        android:startOffset="2000"
        android:toAlpha="0" />

</set>
```

3) Move Animation

```
<?xml version="1.0" encoding="utf-8"?>
<set
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:interpolator="@android:anim/linear_interpolator"

android:fillAfter="true">
<translate
        android:fromXDelta="0%p"
        android:toXDelta="75%p"
        android:duration="700" />
</set>
```

4) Rotate Animation

```
<?xml version="1.0" encoding="utf-8"?>
<set
    xmlns:android="http://schemas.android.com/apk/res/android">
    <rotate
        android:duration="6000"
        android:fromDegrees="0"
```

```
            android:pivotX="50%"
            android:pivotY="50%"
            android:toDegrees="360" />

        <rotate
            android:duration="6000"
            android:fromDegrees="360"
            android:pivotX="50%"
            android:pivotY="50%"
            android:startOffset="5000"
            android:toDegrees="0" />

</set>
```

5) Slide Animation

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
        android:fillAfter="true" >
        <scale
            android:duration="500"
            android:fromXScale="1.0"
            android:fromYScale="1.0"
            android:interpolator="@android:anim/linear_interpolator"
            android:toXScale="1.0"
            android:toYScale="0.0" />
</set>
```

6) Zoom Animation

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:fillAfter="true" >

    <scale
        android:duration="500"
        android:fromXScale="1.0"
        android:fromYScale="1.0"
        android:interpolator="@android:anim/linear_interpolator"
        android:toXScale="1.0"
        android:toYScale="0.0" />
</set>
```

Step 6: Working with the MainActivity.java file

Add animation to the ImageView by clicking a specific Button. Navigate to the app > java > your apps package name >> MainActivity.java.

```
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.widget.Button;

import android.widget.ImageView;
public class MainActivity extends AppCompatActivity {

    ImageView imageView;
    Button blinkBTN, rotateBTN, fadeBTN, moveBTN, slideBTN, zoomBTN, stopBTN;
```

```
@Override
protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        imageView = findViewById(R.id.imageview);
        blinkBTN = findViewById(R.id.BTNblink);
        rotateBTN = findViewById(R.id.BTNrotate);
        fadeBTN = findViewById(R.id.BTNfade);
        moveBTN = findViewById(R.id.BTNmove);
        slideBTN = findViewById(R.id.BTNslide);
        zoomBTN = findViewById(R.id.BTNzoom);
        stopBTN = findViewById(R.id.BTNstop);

        blinkBTN.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                        // To add blink animation
Animation animation =AnimationUtils.loadAnimation(getApplicationContext(),
R.anim.blink_animation);
                        imageView.startAnimation(animation);
                }
        });

        rotateBTN.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {
```

```
// To add rotate animation
Animation animation=AnimationUtils.loadAnimation(getApplicationContext(),
R.anim.rotate_animation);
                        imageView.startAnimation(animation);
                }
        });
        fadeBTN.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                        // To add fade animation
                        Animation animation =
AnimationUtils.loadAnimation(getApplicationContext(), R.anim.fade_animation);
                        imageView.startAnimation(animation);
                }
        });
        moveBTN.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                        // To add move animation
                        Animation animation =
AnimationUtils.loadAnimation(getApplicationContext(), R.anim.move_animation);
                        imageView.startAnimation(animation);
                }
        });
        slideBTN.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                        // To add slide animation
                        Animation animation =
```

```
AnimationUtils.loadAnimation(getApplicationContext(), R.anim.slide_animation);
                        imageView.startAnimation(animation);
                    }
                });
                zoomBTN.setOnClickListener(new View.OnClickListener() {

@Override
                    public void onClick(View v)
{
                        // To add zoom animation
                        Animation animation =
AnimationUtils.loadAnimation(getApplicationContext(), R.anim.zoom_animation);
                        imageView.startAnimation(animation);
                    }
                });
                stopBTN.setOnClickListener(new View.OnClickListener() {
                    @Override
                    public void onClick(View v) {
                        // To stop the animation going on imageview
                        imageView.clearAnimation();
                    }
                });
        }
}
```

Output:

4.5 Assignment Questions

1. What is the importance of frame in animation?
2. What is frame by frame animation in Android?
3. What are Android frameworks?
4. What are the different types of animation?

Conclusion: Thus a different animation is given to the screen in different ways.