

COMPS413F-Quiz2-1516_question_Q3

Question 3 **[10 marks]**

In no more than 150 words, **describe** and **illustrate** with diagrams about what you would see when the Android application program "ViewActivity" is started and the Button is clicked.

Listing below depicts the layout resource file "main.xml".

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:orientation="vertical"
   android:gravity="center_vertical" >
```

```
    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Animation" />

</LinearLayout>
```

Listing below depicts the animation resource file “animation.xml”.

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

    <alpha
        android:fromAlpha="1"
        android:toAlpha="0"
        android:duration="3000"
        android:repeatMode="reverse"
        android:repeatCount="1" />

    <rotate
        android:fromDegrees="0"
        android:toDegrees="180"
        android:pivotX="100%"
        android:pivotY="100%"
        android:startOffset="6000"
        android:duration="5000" />

</set>
```

Listing below depicts the activity class “ViewActivity.java”.

```
public class ViewActivity extends Activity {
    Button button;

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        button = (Button) findViewById(R.id.button);

        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                buttonAnimation(button);
            }
        });
    }
```

```
    /** Performs animation on a Button. */
    private void buttonAnimation(Button button) {
        Animation animation = AnimationUtils.loadAnimation(this, R.anim.animation);
        button.startAnimation(animation);
    }
}
```

Hint:

Elements:

`android:startOffset`
The amount of milliseconds the animation delays after start() is called.

[10]

Question 3 **[10 marks]**

The Button is aligned to the left side and placed in the vertical center of its parent container.

The animation duration is 1100ms (or 11s).

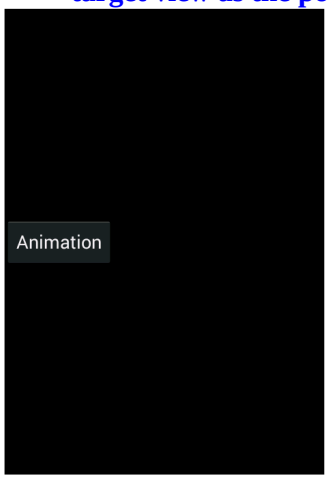
The "AnimationSet" specifies a composite of alpha and rotation animation in sequential order.

The "<alpha>" animation

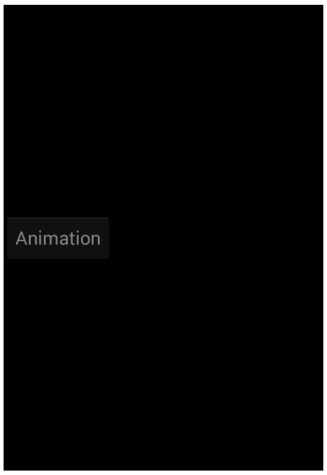
- **Change the opacity of the target view from fully opaque to fully transparent in 3000ms (or 3s).**
- **The repeated rounds are animated in reverse.**

The "<rotate>" tag

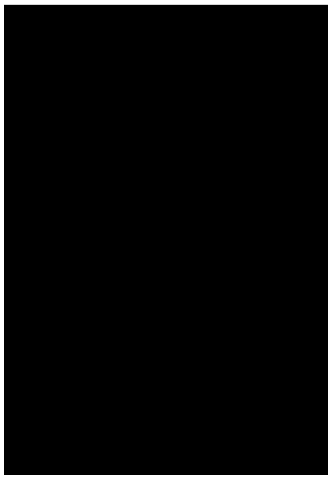
- **The rotation is from 0 to 180 degrees.**
- **The pivot attributes of "100%" and "100%" specify the bottom-right corner of the target view as the point about which the view is rotated.**



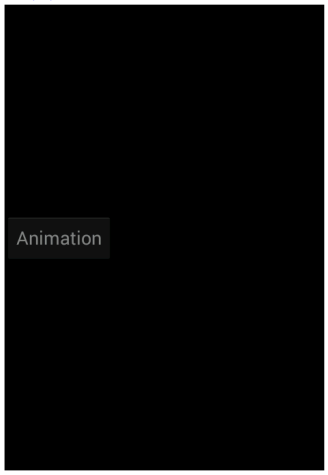
(a) Start of animation.



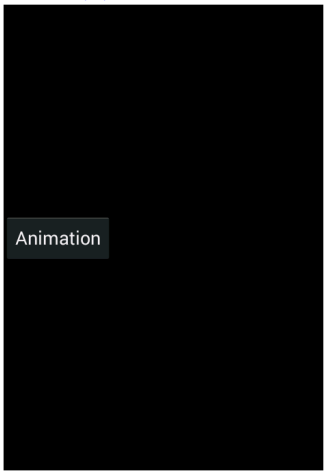
(b) After 1.5s.



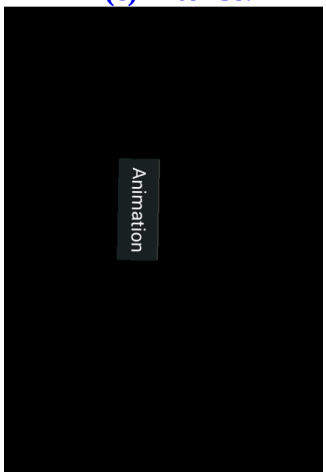
(c) After 3s.



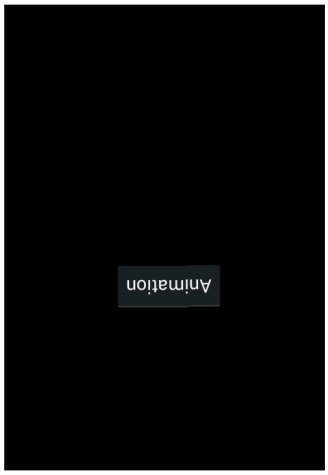
(d) After 4.5s.



(e) After 6s.



(f) After 8.5s.



(g) End of animation (11s)

Illustration of animation with diagram.

Question 3 **[11 marks]**

In no more than 150 words, **describe** and **illustrate** with diagrams about what you would see when the Android application program "ViewActivity" is started and the button is clicked.

Listing below depicts the layout resource file "main.xml".

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:orientation="vertical"
   android:gravity="right" >

   <Button
      android:id="@+id/button"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Animation" />

</LinearLayout>
```

Listing below depicts the animation resource file "animation.xml".

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
   android:duration="6000"
   android:repeatMode="reverse" >
   <alpha
      android:fromAlpha="1"
      android:toAlpha="0"
      android:repeatCount="1" />

   <scale
      android:fromXScale="1"
      android:toXScale="1"
      android:fromYScale="1"
      android:toYScale="2"
      android:pivotX="0%"
      android:pivotY="0%"
      android:repeatCount="1" />

</set>
```

Listing below depicts the activity class "ViewActivity.java".

```
public class ViewActivity extends Activity {
   Button button;

   /** Called when the activity is first created. */
   @Override
```



```
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        button = (Button) findViewById(R.id.button);

        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                buttonAnimation(button);
            }
        });
    }

    /** Performs animation on a Button. */
    private void buttonAnimation(Button button) {
        Animation animation = AnimationUtils.loadAnimation(this, R.anim.animation);
        button.startAnimation(animation);
    }
}
```

[11]

Question 3 **[11 marks]**

The ImageView is positioned at top-right corner of its parent container.
The animation duration is 12000ms (or 12s).
The repeated rounds are animated in reverse.
The "AnimationSet" specifies a composite of two transformations

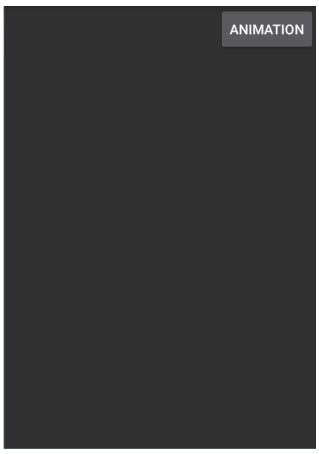
- **AlphaAnimation**
- **ScaleAnimation**

The AlphaAnimation

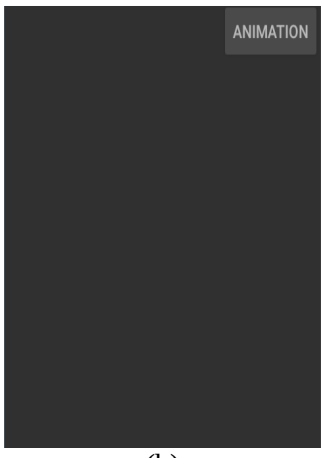
- **Changes the opacity of the target view from fully opaque to fully transparent.**

The ScaleAnimation

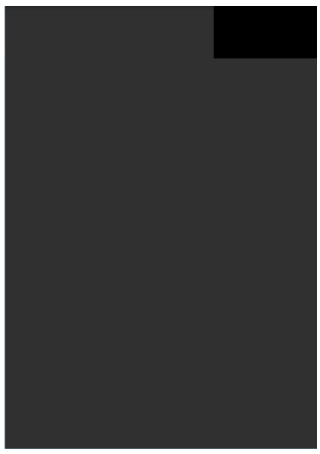
- **Maximizes the height of the target view from 100% to 200%.**
- **The pivot values specify the top-left corner of the target view as the point about which the view is scaled.**

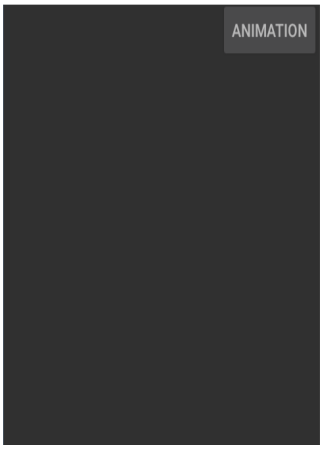


(a) Start of animation.

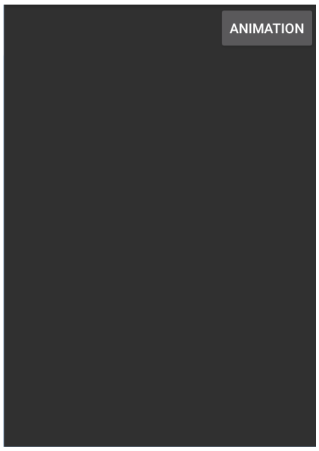


(b)


(c) After 6s.



(d)



(e) End of animation.
After 12s

Illustration of animation with diagrams.

Question 3 **[10 marks]**

In no more than 150 words, **describe** and **illustrate** with diagrams about what you would see when the Android application program "ViewActivity" is started and the View object is clicked.

Listing below depicts the layout resource file "main.xml".

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:orientation="vertical"
   android:gravity="bottom|center_horizontal" >

   <ImageView
      android:id="@+id/imageView"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:src="@drawable/t" />

</LinearLayout>
```

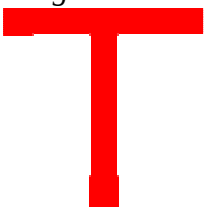
Listing below depicts the activity class "ViewActivity.java".

```
public class ViewActivity extends Activity {
   ImageView imageView;

   /** Called when the activity is first created. */
   @Override
   public void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.main);

      imageView = (ImageView) findViewById(R.id.imageView);
      imageView.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View v) {
            doAnimation(imageView);
         }
      });
   }

   /** Performs animation on a View object. */
   private void doAnimation(ImageView imageView) {
      AnimationSet animationSet = new AnimationSet(true);

      Animation animation1 = new AlphaAnimation(1, 0);
      animation1.setDuration(5000);
      animationSet.addAnimation(animation1);
```

```
        int px = imageView.getWidth()/2;
        int py = imageView.getHeight();
        Animation animation2 = new ScaleAnimation(1, 2, 1, 2, px, py);
        animation2.setDuration(5000);
        animationSet.addAnimation(animation2);

        imageView.startAnimation(animationSet);
    }
}
```

Given:

Image: "drawable/t.png"



android.view.animation.AnimationSet

public void addAnimation(Animation a)	
Add a child animation to this animation set. The transforms of the child animations are applied in the order that they were added.	
A	Animation to add.

android.view.animation.Animation

public void setDuration(long durationMillis)	
How long this animation should last. The duration cannot be negative.	
durationMillis	Duration in milliseconds.

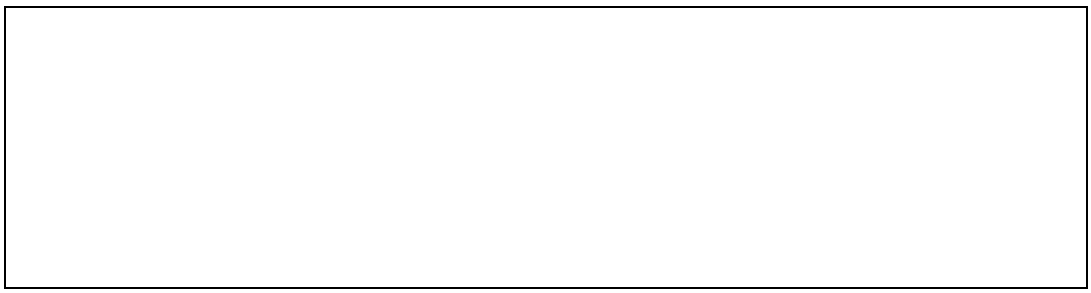
android.view.animation.AlphaAnimation

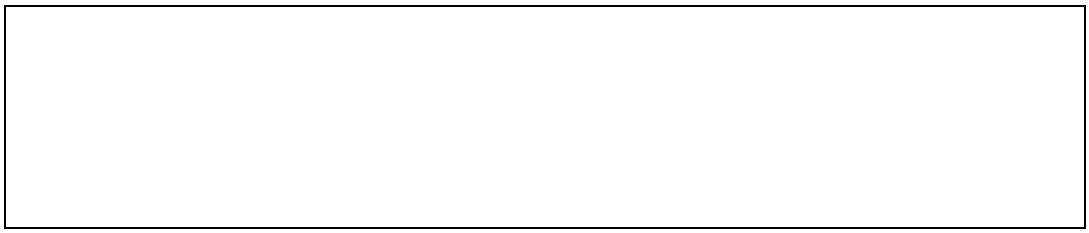
public AlphaAnimation(float fromAlpha, float toAlpha)	
Constructor to use when building an AlphaAnimation from code.	
fromAlpha	Starting alpha value for the animation, where 1.0 means fully opaque and 0.0 means fully transparent.
toAlpha	Ending alpha value for the animation.

android.view.animation.ScaleAnimation

public ScaleAnimation (float fromX, float toX, float fromY, float toY, float pivotX, float pivotY)	
Constructor to use when building a ScaleAnimation from code.	
fromX	Horizontal scaling factor to apply at the start of the animation
toX	Horizontal scaling factor to apply at the end of the animation
fromY	Vertical scaling factor to apply at the start of the animation
toY	Vertical scaling factor to apply at the end of the animation
pivotX	The X coordinate of the point about which the object is being scaled, specified as an absolute number where 0 is the left edge. (This point remains fixed while the object changes size.)
pivotY	The Y coordinate of the point about which the object is being scaled, specified as an absolute number where 0 is the top edge. (This point remains fixed while the object changes size.)

[10]





Question 3 **[10 marks]**

The ImageView is positioned at bottom and center of its parent container.
The animation duration is 5000ms (or 5s).
No repeated rounds are animated.
The "AnimationSet" specifies a composite of two transformations

- **AlphaAnimation**
- **ScaleAnimation**

The AlphaAnimation

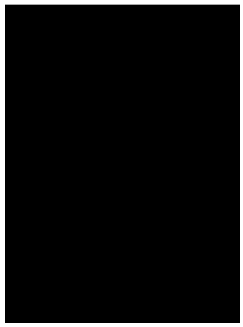
- **Changes the opacity of the target view from fully opaque to fully transparent.**

The ScaleAnimation

- **Maximizes the height and width of the target view from 100% to 200%.**
- **The pivot values specify the bottom and center of the target view as the point about which the view is scaled.**

(a) Start of animation.

(b) After 2.5s.



(c) After 5s.

Illustration of animation with diagrams.

Question 4 **[12 marks]**

Listing below depicts the layout resource file "main.xml".

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:orientation="vertical"
   android:gravity="center_vertical" >

   <Button
      android:id="@+id/button"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Animation" />

</LinearLayout>
```

Listing below depicts the activity class "ViewAnimationActivity.java".

```
public class View3Activity extends Activity {
   Button button;

   /** Called when the activity is first created. */
   @Override
   public void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.main);

      button = (Button) findViewById(R.id.button);
      button.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View v) {
            doAnimation(button);
         }
      });
   }


   /** Performs translation animation on a button. */
   private void doAnimation(Button button) {
      AnimationSet animationSet = new AnimationSet(true);

      int xDistance1 = (((View)button.getParent()).getWidth() - button.getWidth())/2;
      int yDistance1 = (((View)button.getParent()).getHeight() - button.getHeight())/2;
      Animation animation1 = new TranslateAnimation(0, xDistance1, 0, -yDistance1);
      animation1.setDuration(4000);
      animationSet.addAnimation(animation1);
```

```
        Animation animation2 = new TranslateAnimation(0, xDistance1, 0, yDistance1);
        animation2.setDuration(3000);
        animation2.setStartOffset(4000);
        animationSet.addAnimation(animation2);

        Animation animation3 = new TranslateAnimation(0, -xDistance1, 0, yDistance1);
        animation3.setDuration(2000);
        animation3.setStartOffset(7000);
        animationSet.addAnimation(animation3);

        Animation animation4 = new TranslateAnimation(0, -xDistance1, 0, -yDistance1);
        animation4.setDuration(1000);
        animation4.setStartOffset(9000);
        animationSet.addAnimation(animation4);

        button.startAnimation(animationSet);
    }

}
```

Hint:

android.view.animation.AnimationSet

`public void addAnimation(Animation a)`	
Add a child animation to this animation set. The transforms of the child animations are applied in the order that they were added.	
`a`	Animation to add.

android.view.animation.Animation

`public void setStartOffset(long startOffset)`	
When this animation should start relative to the start time	
`startOffset`	When this Animation should start, in milliseconds from the start time of the root AnimationSet.

android.view.animation.TranslateAnimation

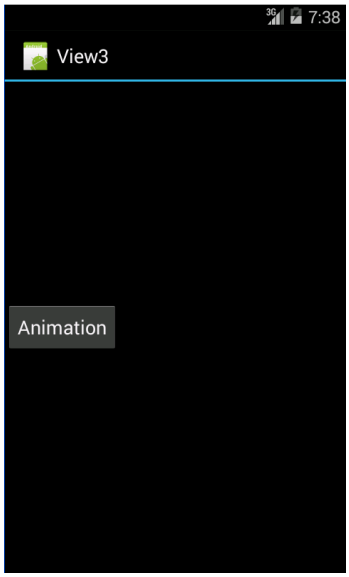
`TranslateAnimation(float fromXDelta, float toXDelta, float fromYDelta, float toYDelta)`	
Constructor to use when building a TranslateAnimation from code	
`fromXDelta`	Change in X coordinate to apply at the start of the animation
`toXDelta`	Change in X coordinate to apply at the end of the animation
`fromYDelta`	Change in Y coordinate to apply at the start of the animation
`toYDelta`	Change in Y coordinate to apply at the end of the animation

Describe and **illustrate** with diagram about what you would see when the application program "View3Activity" is started and the button labeled with "Animation" is clicked.

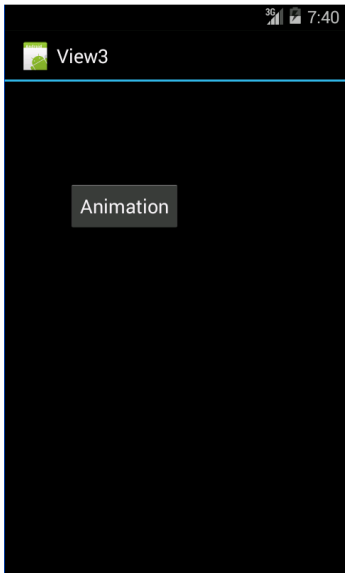
[12]

Question 4 **[12 marks]**

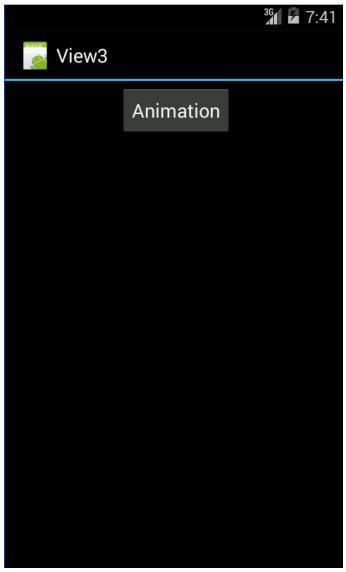
- **The Button is aligned to the left side and placed in the vertical center of its parent container.**
- **The animation duration is 10000ms (or 10s).**
- **No repeated rounds are animated.**
- **The "AnimationSet" specifies a composite of four translation animation in sequential order.**
- **The first translation animation slides from its initial position to center of the top boundary of the container in 4s.**
- **It then moves to the center of the right boundary in 3s.**
- **The third movement is completed in 2s from the right side to the center position of the bottom edge.**
- **Finally, the button moves back to its initial position in 1s.**



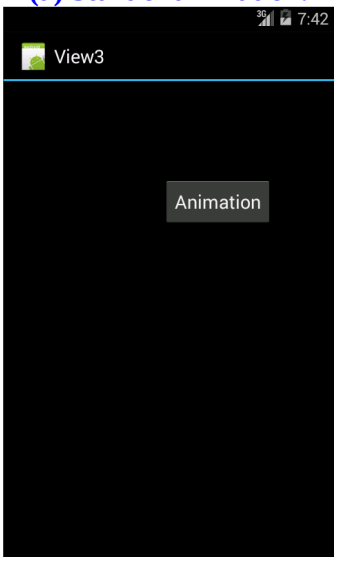
(a) Start of animation.



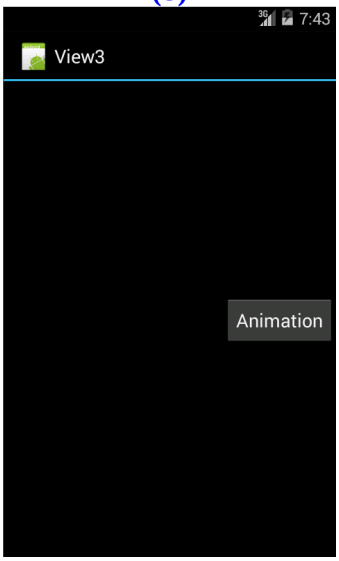
(b)



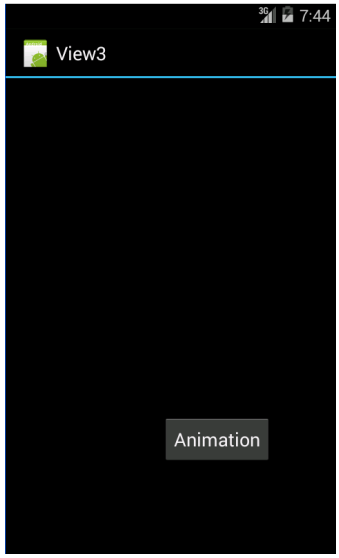
(c) After 4s.



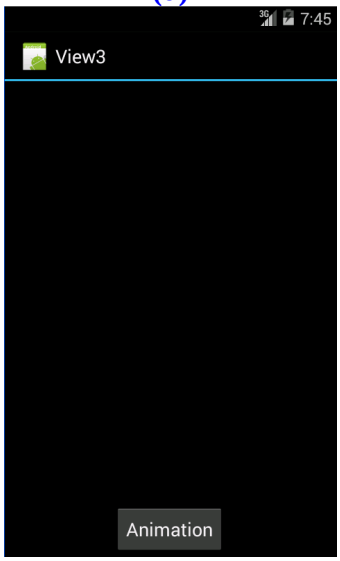
(d)



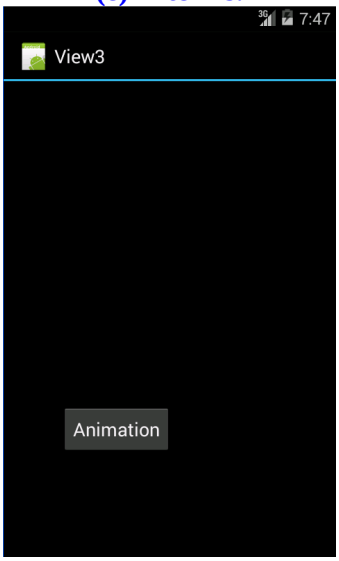
(e) After 7s.



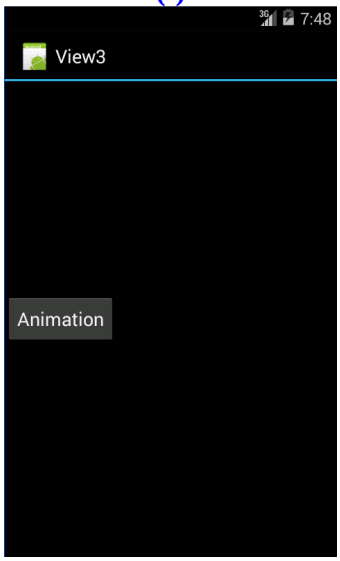
(f)



(g) After 9s.



(i)



(j) End of animation (10s).

Illustration of animation with diagram.

Question 3 **[11 marks]**

In no more than 150 words, **describe** and **illustrate** with diagrams about what you would see when the Android application program "ViewActivity" is started and the View object is clicked.

Listing below depicts the layout resource file "main.xml".

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="fill_parent"
   android:layout_height="fill_parent"
   android:orientation="vertical" >

   <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_centerInParent="true"
      android:text="Quiz 2"
      android:id="@+id/textView" />

</RelativeLayout>
```

Listing below depicts the animation resource file "animation.xml".

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

   <scale
      android:fromXScale="2"
      android:toXScale="1"
      android:fromYScale="3"
      android:toYScale="1"
      android:pivotX="50%"
      android:pivotY="50%"
      android:repeatCount="0"
      android:duration="3000" />

   <rotate
      android:fromDegrees="0"
      android:toDegrees="1080"
      android:pivotX="0%"
      android:pivotY="0%"
      android:repeatCount="0"
      android:startOffset="3000"
      android:duration="5000" />
</set>
```

Listing below depicts the activity class "ViewActivity.java".

```
public class ViewActivity extends Activity {
    TextView textView;

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        textView = (TextView) findViewById(R.id.textView);
        textAnimation(textView);
    }

    /** Performs animation on a TextView. */
    private void textAnimation(TextView textView) {
        Animation animation = AnimationUtils.loadAnimation(this, R.anim.animation);
        textView.startAnimation(animation);
    }
}
```

Elements:

`android:startOffset`
The amount of milliseconds the animation delays after start() is called.

[11]

Question 3 **[15 marks]**

(a) **Name** and **describe** the technique for improving the rendering quality of screen.

[3]

(b) **What** types of transformation are supported by view animation in Android?

[4]

(c) In no more than 150 words, **describes** and **illustrates** with diagram about what you would see when the Android application program "ViewAnimationActivity" is started.

Listing below depicts the layout resource file "main.xml"

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="fill_parent"
   android:layout_height="fill_parent"
   android:orientation="vertical" >
   <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_centerInParent="true"
      android:text="Animation"
      android:id="@+id/textView1" />
</LinearLayout>
```

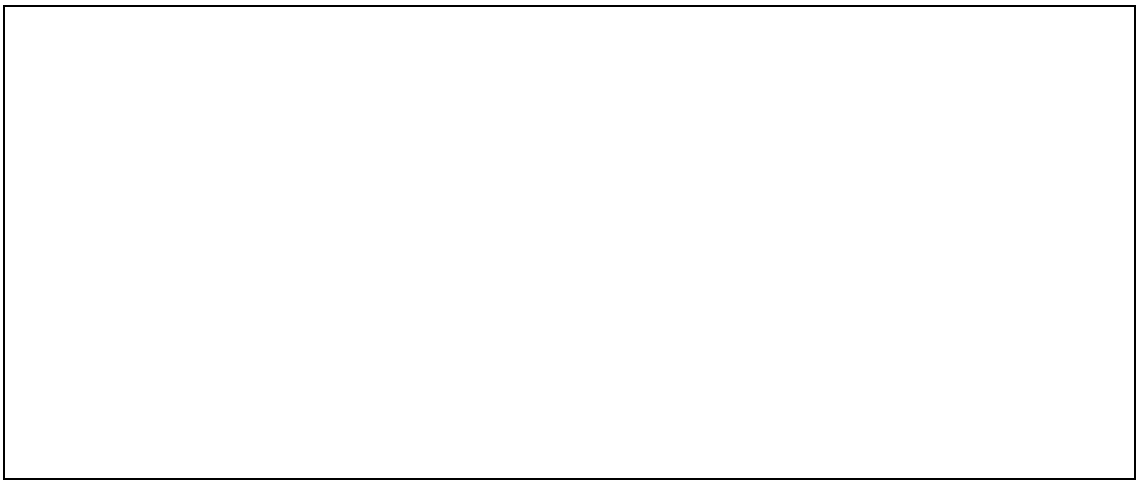
Listing below depicts the animation resource file "animation.xml"

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
   android:duration="5000">
   <alpha
      android:fromAlpha="0"
      android:toAlpha="1" />
   <scale
      android:fromXScale="0"
      android:toXScale="1"
      android:fromYScale="0"
      android:toYScale="1"
      android:pivotX="50%"
      android:pivotY="50%" />
   <rotate
      android:fromDegrees="0"
      android:toDegrees="360"
      android:pivotX="50%"
      android:pivotY="50%" >
   </rotate>
</set>
```

Listing below depicts the activity class “ViewAnimationActivity.java”

```
public class ViewAnimationActivity extends Activity {
   TextView textView1;

   /** Called when the activity is first created. */
   @Override
   public void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.main);
      textView1 = (TextView) findViewById(R.id.textView1);

      /* Performs animation on a TextView. */
      Animation animation = AnimationUtils.loadAnimation(this, R.anim.animation);
      textView1.startAnimation(animation);
   }
}
```

[8]



Question 3 **[15 marks]**

(a) **Double buffering**

- **Uses two buffers to reduce the operations on the visible display for avoiding flickering.**

OR

Use of "SurfaceView"

- **Uses two threads to reduce the usage of the main UI thread for maintaining responsiveness.**

(b) **Translation**
Rotation
Scale
Alpha

(c) **The TextView is positioned at the center of the container vertically and horizontally.**
The animation duration is 5000ms (or 5s).
No repeated rounds are animated.
The root tag "<set>" specifies a composite of three transformations

- **<alpha>**
- **<scale>**
- **<rotate>**

The "<alpha>" animation

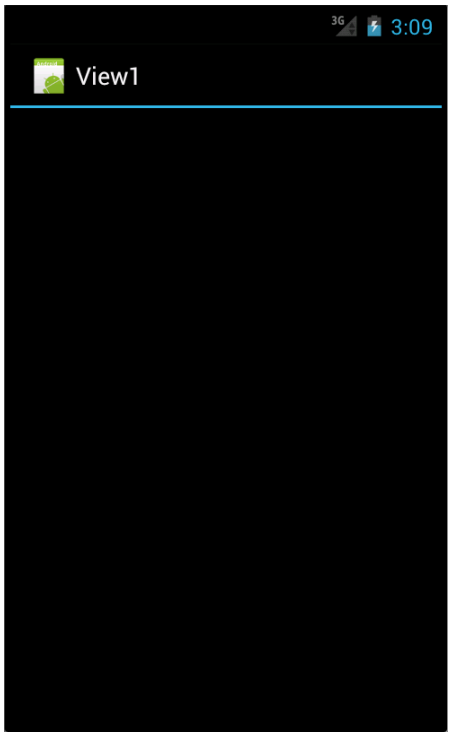
- **Change the opacity of the target view from fully transparent to fully opaque.**

The "<scale>" animation

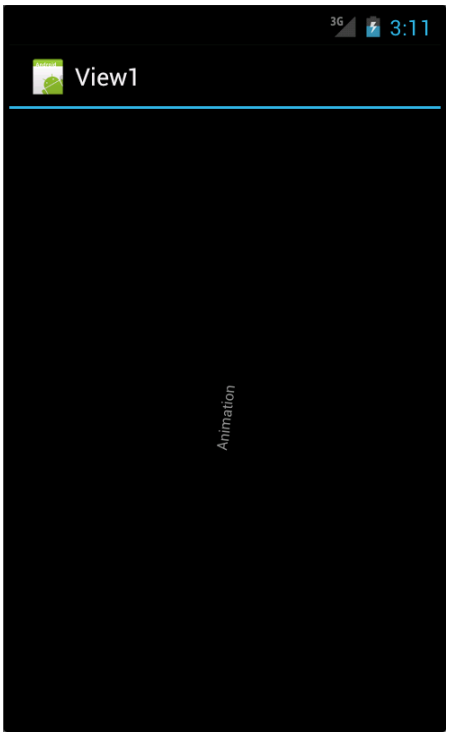
- **Totally maximizes the target view around its center.**

The "<rotate>" tag

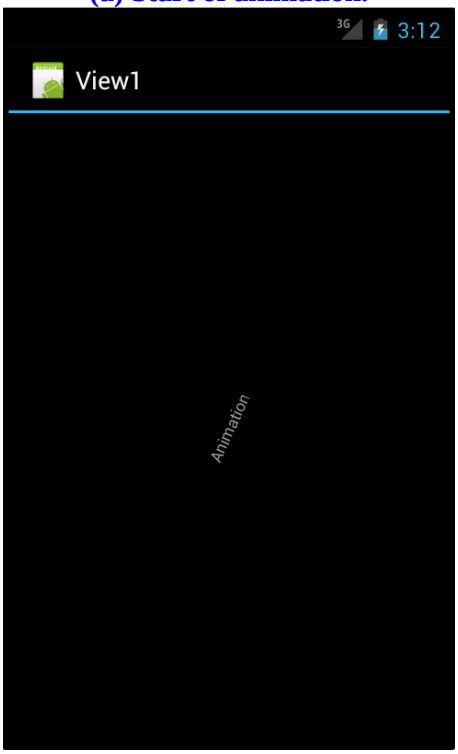
- **The rotation is a complete round, from 0 to 360 degrees.**
- **The pivot attributes of "50%" and "50%" specify the center of the target view as the point about which the view is rotated.**



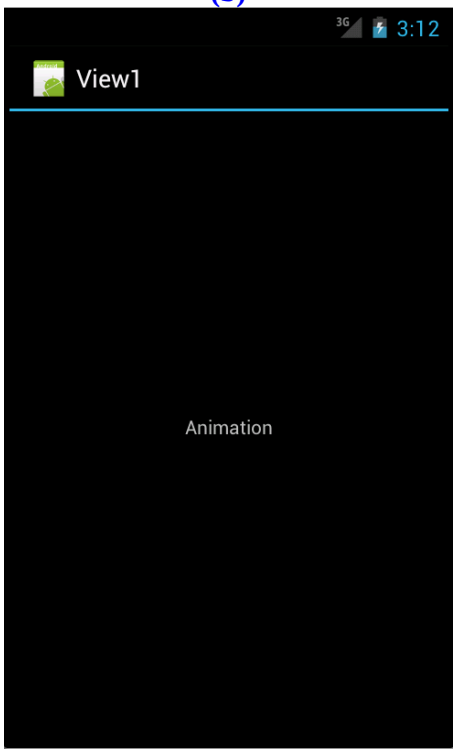
(a) Start of animation.



(b)



(c)



(d) End of animation.

Illustration of animation with diagram.