# INDEX

| Exp. No: 1a | GUI Components, Fonts and Colours - Linear Layout |
|---|---|
| Date: 30.01.2024 | |

**Aim:**

To develop an android application that invokes GUI components, Font and Colors using android studio and sdk**.**

**Algorithm:**

Step 1: File -> NewProject

Provide the application name and Click “Next”

Step 2: Select the target android devices

Select the minimum SDK to run the application. Click “Next”.

Step 3: Choose the activity for the application (By default choose “Blank Activity). Click “Next”.

Step 4: Enter activity name and click "Finish".

Step 5: Edit the program.

Step 6: Run the application, 2-ways to run the application.

1. Running through emulator

2. Running through mobile device

**Source Code:**

**Activity.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
android:background="#FFBB86FC"
android:orientation="vertical">
<TextView
android:id="@+id/textView1"
android:layout_width="158dp"
android:layout_height="66dp"
android:layout_marginLeft="50dp"
android:layout_marginTop="50dp"
```

```
android:text="SANDHIYA"
android:textColor="#FF000000"/>
<Button
android:id="@+id/button1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="submit"
tools:layout_editor_absoluteX="149dp"
tools:layout_editor_absoluteY="466dp" />
<TextView
android:id="@+id/textView2"
android:layout_width="423dp"
android:layout_height="37dp"
android:text="@string/a1"/>
</LinearLayout>
```

**AndroidManifest.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools">
<application
android:allowBackup="true"
android:dataExtractionRules="@xml/data_extraction_rules"
android:fullBackupContent="@xml/backup_rules"
android:icon="@mipmap/ic_launcher"
android:label="@string/app_name"
android:roundIcon="@mipmap/ic_launcher_round"
android:supportsRtl="true"
android:theme="@style/Theme.DURGA"
tools:targetApi="31">
<activity
android:name=".MainActivity"
android:exported="true">
<intent-filter>
<action android:name="android.intent.action.MAIN" />
<category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
```

```
<meta-data
android:name="android.app.lib_name"
android:value="" />
</activity> </application> </manifest>
```

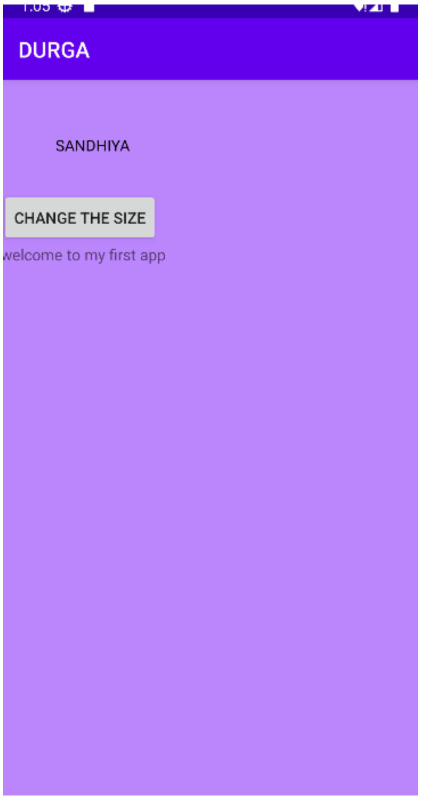
### Colors.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources> <color name="purple_200">#FFBB86FC</color>
<color name="purple_500">#FF6200EE</color>
<color name="purple_700">#FF3700B3</color>
<color name="teal_200">#FF03DAC5</color>
<color name="teal_700">#FF018786</color>
<color name="black">#FF000000</color>
<color name="white">#FFFFFFFF</color> </resources>
```

### String.xml

```
<resources>
<string name="app_name">GUI</string>
<string name="a1">my first app</string>
</resources>
```

### Output:



DURGA

SANDHIYA

welcome to my first app

### Result:

Thus, the program has been successfully executed and output is verified.

| **Exp. No: 1b** | **GUI Components, Fonts and Colours - Relative Layout** |
|---|---|
| **Date: 06.02.2024** | |

**Aim:**

To develop an application that uses GUI components, Font and Colors, using Relative Layout.

**Algorithm:**

Step 1: File -> NewProject

Provide the application name and Click “Next”

Step 2: Select the target android devices

Select the minimum SDK to run the application. Click “Next”.

Step 3: Choose the activity for the application (By default choose “Blank Activity). Click “Next”.

Step 4: Enter activity name and click "Finish".

Step 5: Edit the program.

Step 6: Run the application, 2-ways to run the application.

1. Running through emulator
2. Running through mobile device

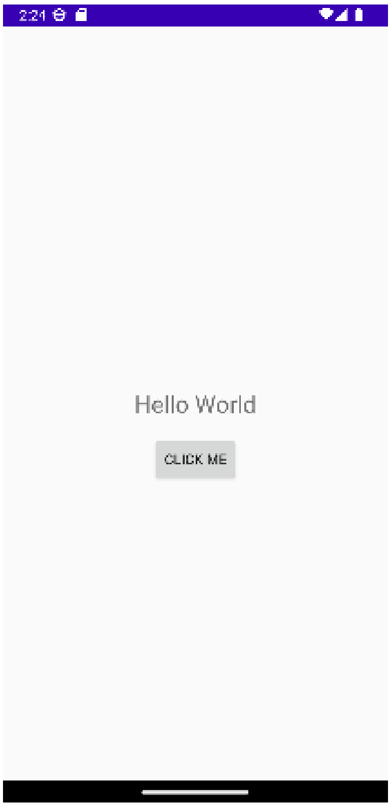
**Source Code:**

**Activity_main.xml**

```
<?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:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp"
    tools:context=".MainActivity">
    <TextView
        android:id="@+id/textViewHello"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World"
        android:textSize="24sp"
```

```
            android:layout_centerInParent="true"/>
        <Button
            android:id="@+id/buttonClickMe"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Click Me"
            android:layout_below="@id/textViewHello"
            android:layout_centerHorizontal="true"
            android:layout_marginTop="16dp"/>
</RelativeLayout>
```

**Output:**



**Result:**

Thus, the program has been successfully executed and output is verified.

| Exp. No: 1c | GUI Components, Fonts and Colours - Java Coding |
|---|---|
| Date: 13.02.2024 | |

**Aim:**

To Develop an application that uses GUI components, Font and Colours.(with java coding)

**Algorithm:**

Step 1: File -> NewProject

Provide the application name and Click “Next”

Step 2: Select the target android devices

Select the minimum SDK to run the application. Click “Next”.

Step 3: Choose the activity for the application (By default choose “Blank Activity). Click “Next”.

Step 4: Enter activity name and click "Finish".

Step 5: Edit the program.

Step 6: Run the application, 2-ways to run the application.

1. Running through emulator

2. Running through mobile device

**Source Code:**

**Activity_main.java**

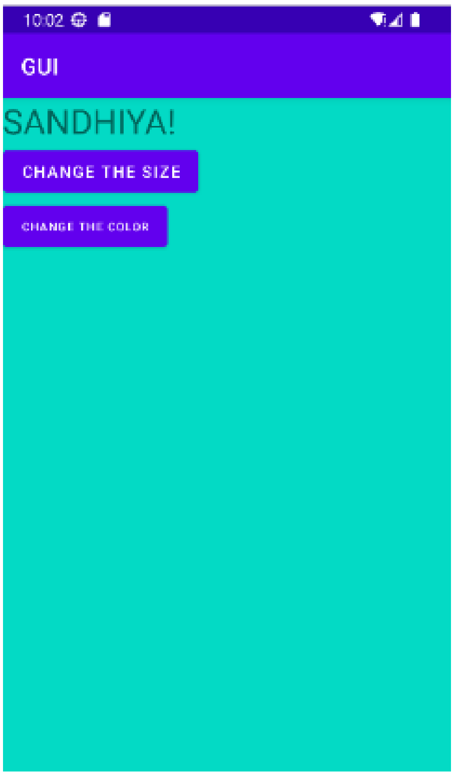
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity"
    android:background="@color/btnColor"
    android:orientation="vertical">
    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="SANDHIYA!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.529"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.396" />
    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="CHANGE THE SIZE"
        tools:layout_editor_absoluteX="149dp"
        tools:layout_editor_absoluteY="466dp" />
    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="CHANGE THE COLOR"
        android:textSize="10sp"
        tools:layout_editor_absoluteX="149dp"
        android:gravity="center"
        tools:layout_editor_absoluteY="466dp" />
</LinearLayout>
```

## MainActivity.Java

```
package com.example.gui;
import androidx.appcompat.app.AppCompatActivity;
import android.graphics.Color;
import android.view.View;
import android.widget.Button;
import android.widget.Button;
import android.widget.TextView;
import android.os.Bundle;
public class MainActivity extends AppCompatActivity {
    int ch = 1;
    float font = 30;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
```

```
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
final TextView t = (TextView) findViewById(R.id.textView);
Button b1 = (Button) findViewById(R.id.button1);
b1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        t.setTextSize(font);
        font = font + 5;
        if (font == 50)
            font = 30;
    }
});
Button b2 = (Button) findViewById(R.id.button2);
b2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        switch (ch) {
            case 1:
                t.setTextColor(Color.RED);
                break;
            case 2:
                t.setTextColor(Color.GREEN);
                break;
            case 3:
                t.setTextColor(Color.BLUE);
                break;
            case 4:
                t.setTextColor(Color.CYAN);
                break;
            case 5:
                t.setTextColor(Color.YELLOW);
                break;
            case 6:
                t.setTextColor(Color.MAGENTA);
                break;
                }
                ch++;
                if (ch == 7)
                    ch = 1;
            }
        });
    }
}
```

**Output:**



**Result:**

Thus, the program has been successfully executed and output is verified.

| Exp. No: 2a | Simple Calculation Application |
|---|---|
| Date: 20.02.2024 | |

**Aim:**

To develop an application for designing a simple calculator app.

**Algorithm:**

Step 1: File -> NewProject

Provide the application name and Click “Next”

Step 2: Select the target android devices

Select the minimum SDK to run the application. Click “Next”.

Step 3: Choose the activity for the application (By default choose “Blank Activity). Click “Next”.

Step 4: Enter activity name and click "Finish".

Step 5: Edit the program.

Step 6: Run the application, 2-ways to run the application.

1. Running through emulator
2. Running through mobile device

**Source Code:**

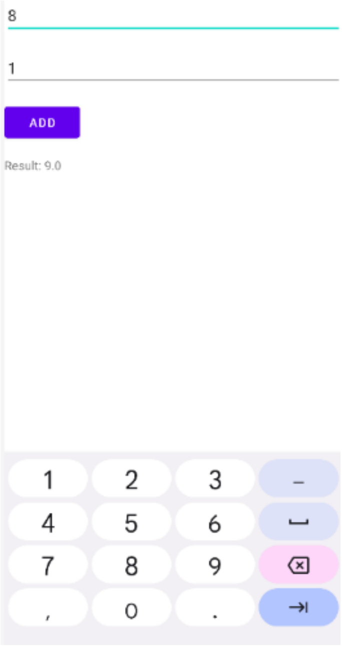
**Activity_main.XML**

```
<?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">
    <EditText
        android:id="@+id/editTextNumber1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter number 1"
        android:inputType="numberDecimal"/>
    <EditText
        android:id="@+id/editTextNumber2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/editTextNumber1"
        android:layout_marginTop="16dp"
        android:hint="Enter number 2"
        android:inputType="numberDecimal"/>
    <Button
        android:id="@+id/btnAdd"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/editTextNumber2"
        android:layout_marginTop="16dp"
        android:text="Add"/>
    <TextView
        android:id="@+id/textViewResult"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@id/btnAdd"
        android:layout_marginTop="16dp"
        android:text="Result: "/>
</RelativeLayout>
```

## Mainactivity.java

```
package com.example.addition;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;
public class MainActivity extends AppCompatActivity {
    private EditText editTextNumber1, editTextNumber2;
    private Button btnAdd;
    private TextView textViewResult;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        editTextNumber1 = findViewById(R.id.editTextNumber1);
        editTextNumber2 = findViewById(R.id.editTextNumber2);
        btnAdd = findViewById(R.id.btnAdd);
        textViewResult = findViewById(R.id.textViewResult);
        btnAdd.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                addNumbers();
            }
        });
    }
    private void addNumbers() {
     try {
       double num1=Double.parseDouble(editTextNumber1.getText().toString());
       double num2=Double.parseDouble(editTextNumber2.getText().toString());
       double result = num1 + num2;
       textViewResult.setText("Result: " + result);
     } catch (NumberFormatException e) {
       textViewResult.setText("Invalid input. Please enter valid numbers.");
    }
}}
```

**Output:**



8

1

Result: 9.0

**Result:**

Thus, the program has been successfully executed and output is verified.

| Exp. No: 2b | Application for Displaying Image |
|---|---|
| Date: 20.02.2024 | |

**Aim:**

To develop an application to display a image on the application.

**Algorithm:**

Step 1: File -> NewProject

Provide the application name and Click “Next”

Step 2: Select the target android devices

Select the minimum SDK to run the application. Click “Next”.

Step 3: Choose the activity for the application (By default choose “Blank Activity). Click “Next”.

Step 4: Enter activity name and click "Finish".

Step 5: Edit the program.

Step 6: Run the application, 2-ways to run the application.

1. Running through emulator
2. Running through mobile device

**Source Code:**

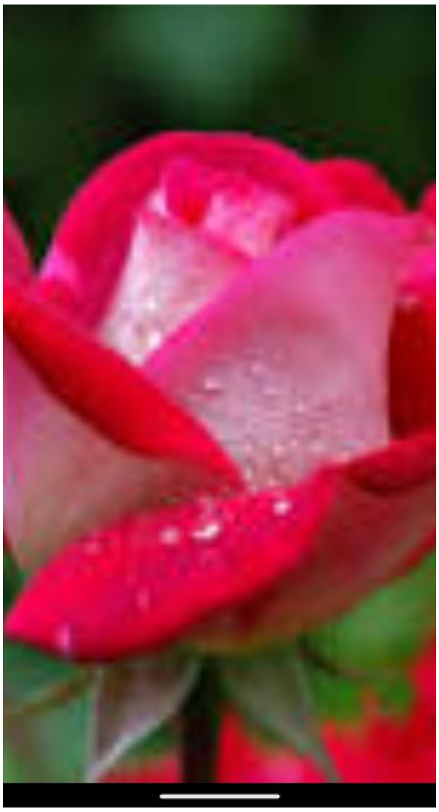
**Activity_main.xml**

```
<?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="match_parent"
        android:layout_height="match_parent"
        android:src="@drawable/rose"
        android:scaleType="centerCrop"/>
</RelativeLayout>
```

**Mainactivity.java**

```
package com.example.imageview;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

**Output:**



**Result:**

Thus, the program has been successfully executed and output is verified.

| **Exp. No: 3** | **Login Interface in Android** |
|---|---|
| **Date: 27.02.2024** | |

**Aim:**

To implement a simple login interface in Android Studio where users can input their username and password and click a login button to proceed.

**Algorithm:**

Step 1: File -> NewProject

Provide the application name and Click “Next”

Step 2: Select the target android devices

Select the minimum SDK to run the application. Click “Next”.

Step 3: Choose the activity for the application (By default choose “Blank Activity). Click “Next”.

Step 4: Enter activity name and click "Finish".

Step 5: Edit the program.

Step 6: Run the application, 2-ways to run the application.

1. Running through emulator

2. Running through mobile device

**Source Code:**

**Activity_main.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<!-- activity_main.xml -->
<LinearLayout 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"
    android:orientation="vertical"
    android:padding="16dp"
    tools:context=".MainActivity">
    <TextView
        android:layout_width="wrap_content"
       android:layout_height="wrap_content"
        android:text="User Name:"
        android:textSize="18sp"
        android:textStyle="bold" />
    <EditText
        android:id="@+id/editTextUsername"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter your username" />
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Password:"
        android:textSize="18sp"
        android:textStyle="bold"
        android:layout_marginTop="16dp" />
    <EditText
        android:id="@+id/editTextPassword"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="Enter your password"
        android:inputType="textPassword" />
    <Button
        android:id="@+id/buttonLogin"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Login"
        android:layout_marginTop="16dp" />
</LinearLayout>
```

## Activity_welcome.xml

```
<?xml version="1.0" encoding="utf-8"?>
<!-- activity_welcome.xml -->
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:text="Welcome Sandhiya!"
    android:textSize="24sp" />
```

**MainActivity.java**

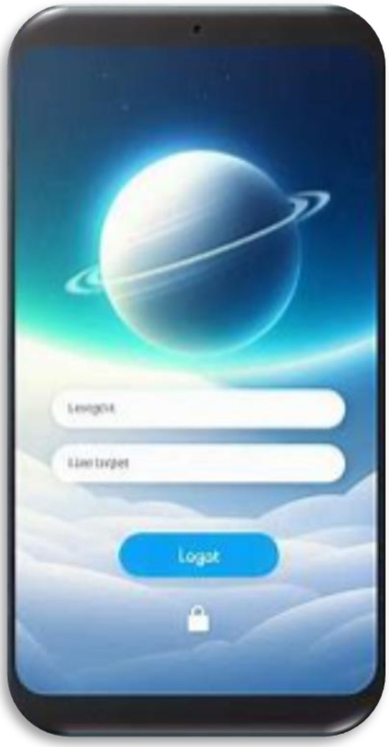
```
package com.example.draganddrop;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    EditText editTextUsername, editTextPassword;

    Button buttonLogin;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);

        editTextUsername = (EditText) findViewById(R.id.editTextUsername);

        editTextPassword = (EditText) findViewById(R.id.editTextPassword);

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

        buttonLogin.setOnClickListener(new View.OnClickListener() {

            @Override

            public void onClick(View v) {

                String username = editTextUsername.getText().toString();

                String password = editTextPassword.getText().toString();

                // Check if the username and password match

                if (username.equals("sandhiya") &&
password.equals("password")) {

                    // If they match, show welcome message and navigate to
WelcomeActivity

                    Toast.makeText(MainActivity.this, "Welcome " + username,
Toast.LENGTH_SHORT).show();

                    Intent intent = new Intent(MainActivity.this,
activity_welcome.class);

                    startActivity(intent);

                } else {

                    // If they don't match, show an error message

                    Toast.makeText(MainActivity.this, "Invalid username or
password", Toast.LENGTH_SHORT).show();
```

```
                }
            }
        });
    }
}
```

### Activity_welcome.java

```
package com.example.draganddrop;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
public class activity_welcome extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_welcome);
    }
}
```

### Output:



### Result:

Thus, the program has been successfully executed and output is verified.

| Exp. No: 4 | Utility Toolkit |
|---|---|
| Date: 05.03.2024 | |

**Aim:**

To develop an application to develop an utility toolkit suing android studio tool.

**Algorithm:**

Step 1: File -> NewProject

Provide the application name and Click “Next”

Step 2: Select the target android devices

Select the minimum SDK to run the application. Click “Next”.

Step 3: Choose the activity for the application (By default choose “Blank Activity). Click “Next”.

Step 4: Enter activity name and click "Finish".

Step 5: Edit the program.

Step 6: Run the application, 2-ways to run the application.

1. Running through emulator
2. Running through mobile device

**Source Code:**

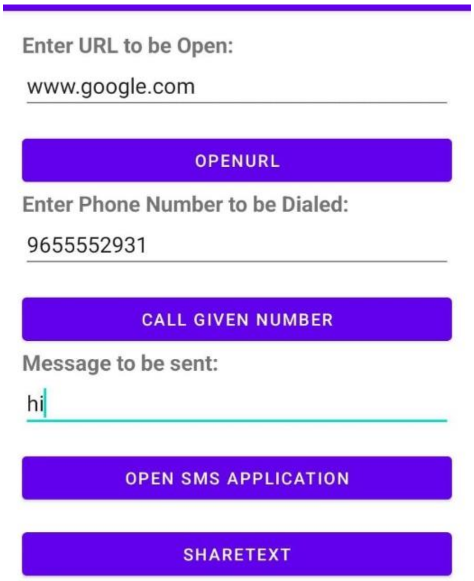
**Activity_main.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:orientation="vertical"
    android:padding="16dp"
    tools:context=".MainActivity">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Enter URL to be Open:"
        android:textSize="18sp"
        android:textStyle="bold" />
    <EditText
        android:id="@+id/txtURL"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />
    <Button
        android:id="@+id/btnOpen"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="OpenURL"
        android:layout_marginTop="16dp" />
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Enter Phone Number to be Dialed:"
        android:textSize="18sp"
        android:textStyle="bold" />
    <EditText
        android:id="@+id/txtPhone"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />
    <Button
        android:id="@+id/btnCall"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="CALL GIVEN NUMBER"
        android:layout_marginTop="16dp" />
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Message to be sent:"
        android:textSize="18sp"
        android:textStyle="bold" />
    <EditText
        android:id="@+id/txtMessage"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />
    <Button
        android:id="@+id/btnSms"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="OPEN SMS Application"
        android:layout_marginTop="16dp" />
    <Button
        android:id="@+id/btnShare"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Sharetext"
        android:layout_marginTop="16dp" />
</LinearLayout>
```

### MainActivity_Java

```
package com.example.explicitindentdemo;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import androidx.appcompat.app.AppCompatActivity;
public class MainActivity extends AppCompatActivity {
    private EditText txtURL, txtPhone, txtMessage;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        txtURL = findViewById(R.id.txtURL);
        txtPhone = findViewById(R.id.txtPhone);
        txtMessage = findViewById(R.id.txtMessage);
        Button btnOpen = findViewById(R.id.btnOpen);
        Button btnCall = findViewById(R.id.btnCall);
        Button btnSms = findViewById(R.id.btnSms);
        Button btnShare = findViewById(R.id.btnShare);
        btnOpen.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String url = txtURL.getText().toString();
                openURL(url);
            }
        });
        btnCall.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String phoneNumber = txtPhone.getText().toString();
                callPhoneNumber(phoneNumber);
            }
        });
        btnSms.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String phoneNumber = txtPhone.getText().toString();
                String message = txtMessage.getText().toString();
                sendSMS(phoneNumber, message);
            }
        })
        btnShare.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String message = txtMessage.getText().toString();
                shareText(message);
            }
        });
    }
    private void openURL(String url) {
        Intent intent = new Intent(Intent.ACTION_VIEW, Uri.parse(url));
        startActivity(intent);
    }

    private void callPhoneNumber(String phoneNumber) {
        Intent intent = new Intent(Intent.ACTION_DIAL);
        intent.setData(Uri.parse("tel:" + phoneNumber));
        startActivity(intent);
    }
    private void sendSMS(String phoneNumber, String message) {
        Intent intent = new Intent(Intent.ACTION_SENDTO, Uri.parse("smsto:" +
Uri.encode(phoneNumber)));
        intent.putExtra("sms_body", message);
        startActivity(intent);
    }
    private void shareText(String text) {
        Intent intent = new Intent(Intent.ACTION_SEND);
        intent.setType("text/plain");
        intent.putExtra(Intent.EXTRA_TEXT, text);
        startActivity(Intent.createChooser(intent, "Share via"));
    }
}
```

### Output:



Enter URL to be Open:

www.google.com

Enter Phone Number to be Dialed:

9655552931

Message to be sent:

hi

### Result:

Thus, the program has been successfully executed and output is verified.

| Exp. No: 5 | Dynamic UI Icon |
|---|---|
| Date: 05.03.2024 | |

**Aim:**

To develop an application to change the Icon of the application dynamically.

**Algorithm:**

Step 1: File -> NewProject

Provide the application name and Click “Next”

Step 2: Select the target android devices

Select the minimum SDK to run the application. Click “Next”.

Step 3: Choose the activity for the application (By default choose “Blank Activity). Click “Next”.

Step 4: Enter activity name and click "Finish".

Step 5: Edit the program.

Step 6: Run the application, 2-ways to run the application.

1. Running through emulator
2. Running through mobile device

**Source Code:**

**Activity_main.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:orientation="vertical"
    tools:context=".MainActivity">
    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/message1"
        android:textColor="@color/purple_200"/>
    <ImageView
        android:id="@+id/imageView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:srcCompat="@drawable/tajmahal" />
    <Button
        android:id="@+id/btnChangeImage"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Change Image" />
    <Button
        android:id="@+id/btnChangeText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="ChangeMessage" />
    <Button
        android:id="@+id/btnChangeColor"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Change Text Color" />
</LinearLayout>
```

## String.xml

```
<resources>
    <string name="app_name">ChangeImage</string>
    <string name="message1">Welcome to Android Mobile Application
Developmet</string>
    <string name="message2">Welcome to Application Development</string>
</resources>
```

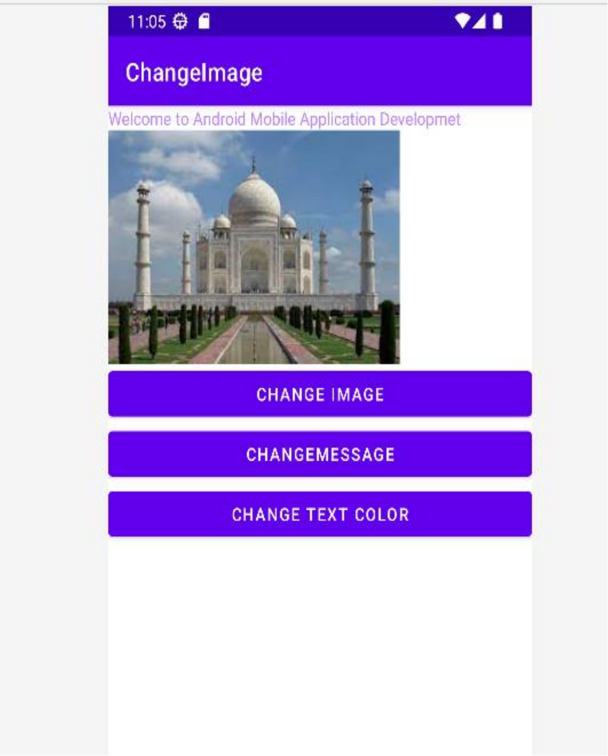
## Android Manifest.Xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">
    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/star"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/star"
        android:supportsRtl="true"
        android:theme="@style/Theme.ChangeImage"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
            <meta-data
                android:name="android.app.lib_name"
                android:value="" />
        </activity>
    </application>
</manifest>
```

### MainActivity.java

```
package com.example.changeimage;
import android.graphics.Color;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;
public class MainActivity extends AppCompatActivity {
    private TextView textView;
    private ImageView imageView;
    private Button btnChangeImage;
    private Button btnChangeText;
    private Button btnChangeColor;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        // Initialize views
        textView = findViewById(R.id.textView);
        imageView = findViewById(R.id.imageView2);
        btnChangeImage = findViewById(R.id.btnChangeImage);
        btnChangeText = findViewById(R.id.btnChangeText);
        btnChangeColor = findViewById(R.id.btnChangeColor);
        // Set click listeners for buttons
        btnChangeImage.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                changeImage();
            }
        });
        btnChangeText.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                changeText();
            }
        });
        btnChangeColor.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                changeTextColor();
            }
        });
    }
    // Method to change the image
    private void changeImage() {
        imageView.setImageResource(R.drawable.shahjahan);
        // For example, imageView.setImageResource(R.drawable.new_image);
    }
    // Method to change the text
    private void changeText() {
        textView.setText(R.string.message2);
        // For example, textView.setText(R.string.new_message);
    }
    // Method to change the text color
    private void changeTextColor() {
        textView.setTextColor(Color.RED);
        // For example, textView.setTextColor(Color.RED);
    }
}
```

**Output:**



11:05

ChangeImage

Welcome to Android Mobile Application Developmet

**Result:**

Thus, the program has been successfully executed and output is verified.

| Exp. No: 6a | Interactive Image Gallery using Radio Buttons |
|---|---|
| Date: 12.03.2024 | |

**Aim:**

To develop an android application that invokes interactive image gallery using radio buttons.

**Algorithm:**

Step 1: File → NewProject

Provide the application name and Click “Next”

Step 2: Select the target android devices

Select the minimum SDK to run the application. Click “Next”.

Step 3: Choose the activity for the application (By default choose “Blank Activity). Click “Next”.

Step 4: Enter activity name and click "Finish".

Step 5: Edit the program.

Step 6: Run the application, 2-ways to run the application.

1. Running through emulator
2. Running through mobile device

**Source Code:**

**Activity_main.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
android:orientation="vertical"
tools:context=".MainActivity">
<TextView
android:id="@+id/textView"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="Image View Demo" />
<ImageView
android:id="@+id/imageView"
```

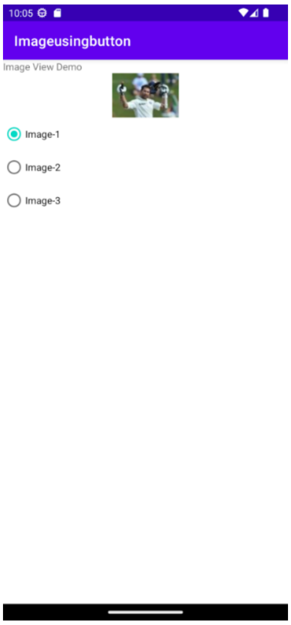
```
android:layout_width="match_parent"
android:layout_height="wrap_content"
app:srcCompat="@drawable/sachin" />
<RadioGroup
android:layout_width="match_parent"
android:layout_height="match_parent">
<RadioButton
android:id="@+id/radioButton1"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:checked="true"
android:text="Image-1" />
<RadioButton
android:id="@+id/radioButton2"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="Image-2" />
<RadioButton
android:id="@+id/radioButton3"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="Image-3" />
</RadioGroup>
</LinearLayout>
```

**MainActivity.java**

```
package com.example.imageusingbutton;
import android.view.View;
import android.os.Bundle;
import android.widget.ImageView;
import android.widget.RadioButton;
import androidx.appcompat.app.AppCompatActivity;
public class MainActivity extends AppCompatActivity {
   RadioButton r1,r2,r3;
   ImageView imageView1;
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_main);
        r1 = findViewById(R.id.radioButton1);
        r2 = findViewById(R.id.radioButton2);
        r3 = findViewById(R.id.radioButton3);
        imageView1 = findViewById(R.id.imageView);
        r1.setOnClickListener(new View.OnClickListener() {
        public void onClick(View view) {
            setImage();
        }});
    r2.setOnClickListener(new View.OnClickListener() {
    public void onClick(View view) {
        setImage();
    }});}
private void setImage() {
    if (r1.isChecked())
        imageView1.setImageResource(R.drawable.sachin);
    else if (r2.isChecked())
        imageView1.setImageResource(R.drawable.dhoni);
    else if (r3.isChecked())
        imageView1.setImageResource(R.drawable.koli);
}}
```

**Output:**



**Result:**

Thus the program has been successfully executed and output is verified.

| Exp. No: 6b | Exploring Calendar Views: A Hands-On Lab Exercise |
|---|---|
| Date: 12.03.2024 | |

**Aim:**

To develop an android application that explores Calendar Views.

**Algorithm:**

Step 1: File → NewProject

Provide the application name and Click “Next”

Step 2: Select the target android devices

Select the minimum SDK to run the application. Click “Next”.

Step 3: Choose the activity for the application (By default choose “Blank Activity). Click “Next”.

Step 4: Enter activity name and click "Finish".

Step 5: Edit the program.

Step 6: Run the application, 2-ways to run the application.

1. Running through emulator
2. Running through mobile device

**Source Code:**

**Activity.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
android:orientation="vertical"
android:padding="16dp"
tools:context=".MainActivity">
<TextView
android:id="@+id/textView2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Calendar"
android:textSize="24sp"
android:layout_gravity="center_horizontal"
android:layout_marginBottom="16dp" />
```

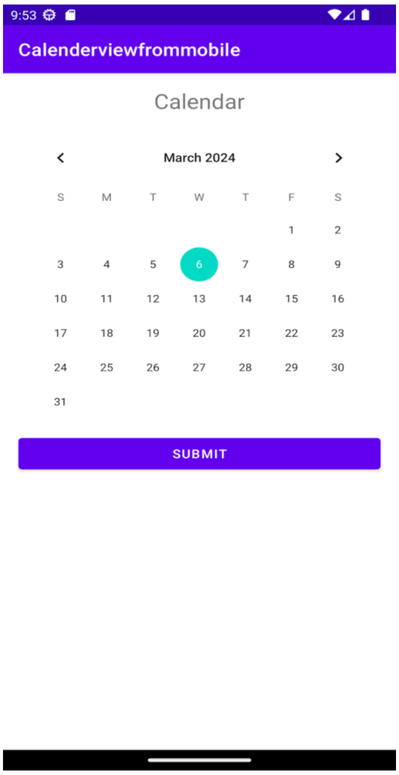
```
<CalendarView
android:id="@+id/calendarView3"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_gravity="center_horizontal"
android:layout_marginBottom="16dp"
android:maxDate="30/08/1993"
android:minDate="01/08/1993" />
<Button
android:id="@+id/button"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="Submit"
android:layout_gravity="center_horizontal" />
</LinearLayout>
```

### Main Activity.java

```
package com.example.calenderviewfrommobile;
import android.graphics.Color;
import android.os.Bundle;
import android.view.View;
import android.widget.CalendarView;
import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import com.google.android.material.snackbar.Snackbar;
public class MainActivity extends AppCompatActivity {
CalendarView calendarView1;
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
calendarView1 = findViewById(R.id.calendarView3);
((CalendarView) calendarView1).setFocusedMonthDateColor(Color.RED);
((CalendarView) calendarView1).setUnfocusedMonthDateColor(Color.BLUE);
((CalendarView) calendarView1).setSelectedWeekBackgroundColor(Color.GREEN);
((CalendarView) calendarView1).setOnDateChangeListener(new
CalendarView.OnDateChangeListener() {
```

```
@Override

public void onSelectedDayChange(@NonNull CalendarView view, int year, int
month, int

dayOfMonth) {

Snackbar.make(view, dayOfMonth + "-" + (month + 1) + "-" + year,

Snackbar.LENGTH_LONG).show();

}

});

}

}
```

## Output:



9:53

Calenderviewfrommobile

Calendar

March 2024

| S | M | T | W | T | F | S |
|---|---|---|---|---|---|---|
|  |  |  |  |  | 1 | 2 |
| 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 17 | 18 | 19 | 20 | 21 | 22 | 23 |
| 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| 31 |  |  |  |  |  |  |

## Result:

Thus, the program has been successfully executed and output is verified.

| Exp. No: 7 | Video Playback with MediaController |
|---|---|
| Date: 19.03.2024 | |

**Aim:**

To develop an android application that invokes video playback with MediaController.

**Algorithm:**

Step 1: File → NewProject

Provide the application name and Click “Next”

Step 2: Select the target android devices

Select the minimum SDK to run the application. Click “Next”.

Step 3: Choose the activity for the application (By default choose “Blank Activity). Click “Next”.

Step 4: Enter activity name and click "Finish".

Step 5: Edit the program.

Step 6: Run the application, 2-ways to run the application.

1. Running through emulator
2. Running through mobile device

**Source Code:**

**Activity_XML**

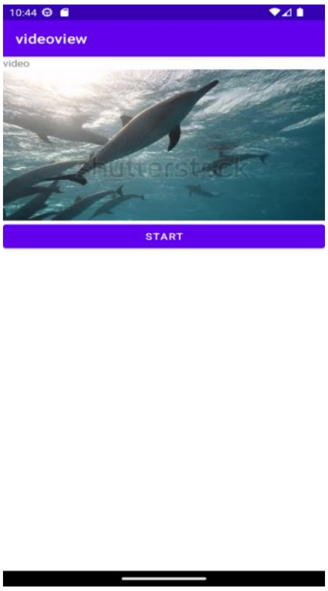
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
tools:context=".MainActivity">
<TextView
android:id="@+id/textView"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:text="video" />
<VideoView
android:id="@+id/videoView2"
android:layout_width="match_parent"
```

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

### Main Activity.java

```
package com.example.videoview;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.MediaController;
import android.widget.Toast;
import android.widget.VideoView;
import androidx.appcompat.app.AppCompatActivity;
public class MainActivity extends AppCompatActivity {
    private VideoView videoView;
    private Button button;
    private MediaController mediaController;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        videoView = findViewById(R.id.videoView2);
        button = findViewById(R.id.button);
        String videoPath = "android.resource://" + getPackageName() + "/" + R.raw.dolphin;
        Uri videoUri = Uri.parse(videoPath);
        mediaController = new MediaController(this);
        mediaController.setAnchorView(videoView);
        videoView.setMediaController(mediaController);
        videoView.setVideoURI(videoUri);
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                if (!videoView.isPlaying()) {
                    videoView.start();
                } else {
                    Toast.makeText(MainActivity.this, "Video is already
playing", Toast.LENGTH_SHORT).show();
                }
            }
        });
    }
    @Override
    protected void onPause() {
        super.onPause();
        if (videoView.isPlaying()) {
            videoView.pause();
        }
    }
    protected void onStop() {
        super.onStop();
        videoView.stopPlayback();
}}
```

**Output:**



**Result:**

Thus the program has been successfully executed and output is verified.

| Exp. No: 8a | Interactive Animation Task: Bouncing Ball |
|---|---|
| Date: 26.03.2024 | |

**Aim:**

To develop an android application that invokes an interactive animation task.

**Algorithm:**

Step 1: File → NewProject

Provide the application name and Click “Next”

Step 2: Select the target android devices

Select the minimum SDK to run the application. Click “Next”.

Step 3: Choose the activity for the application (By default choose “Blank Activity). Click “Next”.

Step 4: Enter activity name and click "Finish".

Step 5: Edit the program.

Step 6: Run the application, 2-ways to run the application.

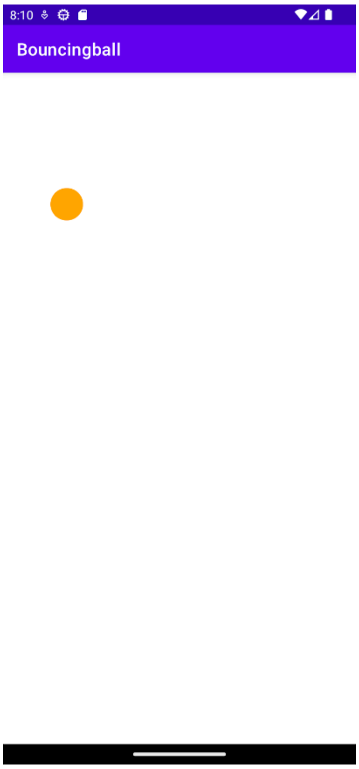
1. Running through emulator
2. Running through mobile device

**Source Code:**

**Main Activity.java**

```
package com.example.bouncingball;
import android.content.Context;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.os.Bundle;
import android.view.View;
import androidx.appcompat.app.AppCompatActivity;
import java.util.Timer;
import java.util.TimerTask;
public class MainActivity extends AppCompatActivity {
    int x = 50, y = 50;
    int dx = 5, dy = 5;
    int maxx = 0, maxy = 0;
    int r = 50;
    TimerTask tt;
    Timer timer = new Timer();
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        BouncingBall bb = new BouncingBall(MainActivity.this);
        setContentView(bb);
        tt = new TimerTask() {
            @Override
            public void run() {
                x = x + dx;
                y = y + dy;
                if (x > maxx || x < 0)
                    dx = -dx;
                if (y > maxy || y < 0)
                    dy = -dy;
                bb.invalidate();
            }
        };
        timer.scheduleAtFixedRate(tt, 0, 10);
    }
    private class BouncingBall extends View{
        public BouncingBall(Context context) {
            super(context);
        }
        @Override
        protected void onDraw (Canvas canvas) {
            maxx = getWidth();
            maxy = getHeight();
            canvas.drawColor(Color.WHITE);
            Paint paint = new Paint();

            paint.setColor(Color.parseColor("#FFA500"));
            canvas.drawCircle(x, y, r, paint);
        }
    }
}
```

**Output:**



**Result:**

Thus the program has been successfully executed and output is verified.

| Exp. No: 8b | Graphical Primitives |
|---|---|
| Date: 26.03.2024 | |

**Aim:**

To develop an android application that invokes graphical primitives.

**Algorithm:**

Step 1: File → NewProject

Provide the application name and Click “Next”

Step 2: Select the target android devices

Select the minimum SDK to run the application. Click “Next”.

Step 3: Choose the activity for the application (By default choose “Blank Activity). Click “Next”.

Step 4: Enter activity name and click "Finish".

Step 5: Edit the program.

Step 6: Run the application, 2-ways to run the application.

1. Running through emulator
2. Running through mobile device

**Source Code:**

**Activity_main.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="38dp"
        android:text="SELECT SHAPE"
        android:textColor="#E91E63" />
    <Spinner
        android:id="@+id/spinner"
        android:layout_width="match_parent"
        android:layout_height="54dp"
        android:background="#FFC107"
        android:spinnerMode="dropdown" />

    <ImageView
        android:id="@+id/imageView2"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        app:srcCompat="@drawable/ic_launcher_foreground" />
</LinearLayout>
```

**Main Activity.java**

```
package com.example.drawshapeprogramatically;
import androidx.appcompat.app.AppCompatActivity;
import android.graphics.Color;
import android.graphics.Paint;
import android.graphics.Path;
import android.graphics.drawable.ShapeDrawable;
import android.graphics.drawable.shapes.ArcShape;
import android.graphics.drawable.shapes.OvalShape;
import android.graphics.drawable.shapes.PathShape;
import android.graphics.drawable.shapes.RectShape;
import android.graphics.drawable.shapes.RoundRectShape;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.ImageView;
import android.widget.Spinner;
import android.widget.Toast;
public class MainActivity extends AppCompatActivity {
    Spinner spinner;
    ImageView imageView;
    String shapes[]={"Rectangle","Rounded
rectangle","Oval","Arc","Path","XML"};

    ArrayAdapter<String>arrayAdapter;

    Paint paint;

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_main);

        spinner=findViewById(R.id.spinner);

        imageView=findViewById(R.id.imageView2);

        spinner.setBackgroundColor(Color.YELLOW);

        arrayAdapter=new ArrayAdapter<String>

                (getApplicationContext(),
android.R.layout.simple_list_item_1,shapes);

        spinner.setAdapter(arrayAdapter);

        spinner.setOnItemSelectedListener(new
AdapterView.OnItemSelectedListener() {

        public void onItemSelected(AdapterView<?> parent, View view, int
position, long id) {

            switch (position) {

              case 0:

                 ShapeDrawable rect = new ShapeDrawable(new RectShape());

                 rect.setIntrinsicHeight(50);

                 rect.setIntrinsicWidth(100);

                 paint = rect.getPaint();

                 paint.setColor(Color.MAGENTA);

                 imageView.setImageDrawable(rect);

                 break;

              case 1:

                 ShapeDrawable roundRect;

                 roundRect = new ShapeDrawable(new RoundRectShape

                 (new float[]{5, 5, 5, 5, 5, 5, 5, 5}, null, null));

                 roundRect.setIntrinsicWidth(100);

                 roundRect.setIntrinsicHeight(50);

                 paint = roundRect.getPaint();

                 paint.setColor(Color.CYAN);

                 imageView.setImageDrawable(roundRect);

                 break;

              case 2:
```

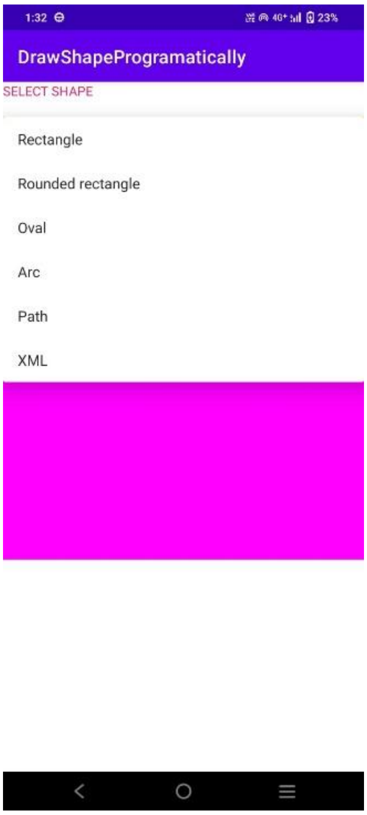
```
   ShapeDrawable oval;
   oval = new ShapeDrawable(new OvalShape());
   oval.setIntrinsicWidth(50);
   oval.setIntrinsicHeight(40);
   paint = oval.getPaint();
   paint.setColor(Color.RED);
   imageView.setImageDrawable(oval);
   break;
case 3:
   ShapeDrawable arc;
   arc = new ShapeDrawable(new ArcShape(0, 345));
   arc.setIntrinsicHeight(100);
   arc.setIntrinsicWidth(100);
   paint = arc.getPaint();
   paint.setColor(Color.CYAN);
   imageView.setImageDrawable(arc);
   break;
case 4:
   Path p = new Path();
   p.moveTo(50, 0);
   p.lineTo(25, 100);
   p.lineTo(100, 50);
   p.lineTo(0, 50);
   p.lineTo(75, 100);
   p.lineTo(50, 0);
   PathShape pathShape = new PathShape(p, 100, 100);
   ShapeDrawable star = new ShapeDrawable(pathShape);
   star.setIntrinsicWidth(100);
   star.setIntrinsicHeight(100);
   paint = star.getPaint();
   paint.setColor(Color.BLUE);
   paint.setStyle(Paint.Style.STROKE);
   paint.setStrokeWidth(1);
   imageView.setImageDrawable(star);
   break;
```

```
                case 5:

                   imageView.setImageResource(R.drawable.oval);

                   break;

              }}

         public void onNothingSelected(AdapterView<?> parent) {

              Toast.makeText(getApplicationContext(), "select shapr from list",
Toast.LENGTH_SHORT).show();

         }

     });

}}
```

## Oval.xml

```
<?xml version="1.0" encoding="utf-8"?>

<shape xmlns:android="http://schemas.android.com/apk/res/android"
android:shape="oval" >

    <solid android:color="#F00"></solid>

    <stroke android:color="#0F0" android:width="3dp" android:dashGap="2dp"
android:dashWidth="4dp"></stroke>

    <gradient android:angle="90" android:startColor="#F00"
android:endColor="#FFF"></gradient>
```

## Output:



## Result:

Thus, the program has been successfully executed and output is verified.

| Exp. No: 9 | SQLiteDatabase Connectivity |
|---|---|
| Date: 02.04.2024 | |

**Aim:**

To develop an android application that invokes SQLiteDatabase connectivity.

**Algorithm:**

Step 1: File → NewProject

Provide the application name and Click “Next”

Step 2: Select the target android devices

Select the minimum SDK to run the application. Click “Next”.

Step 3: Choose the activity for the application (By default choose “Blank Activity). Click “Next”.

Step 4: Enter activity name and click "Finish".

Step 5: Edit the program.

Step 6: Run the application, 2-ways to run the application.

1. Running through emulator
2. Running through mobile device

**Source Code:**

**Activity_main.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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"
    android:orientation="vertical"
    tools:context=".MainActivity">
    <Button
        android:id="@+id/btnCreateDB"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Create DataBase" />
    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Enter Name"/>
    <EditText
        android:id="@+id/txtName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="textPersonName"
        android:text="Name"/>
    <TextView
        android:id="@+id/textView2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="select Programme" />
    <Spinner
        android:id="@+id/txtProg"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="textPersonName"
        android:text="Name"/>
    <Button
        android:id="@+id/btnInsert"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Insert" />
    <TextView
        android:id="@+id/textView3"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="EnterID" />
    <EditText
        android:id="@+id/txtID"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
```

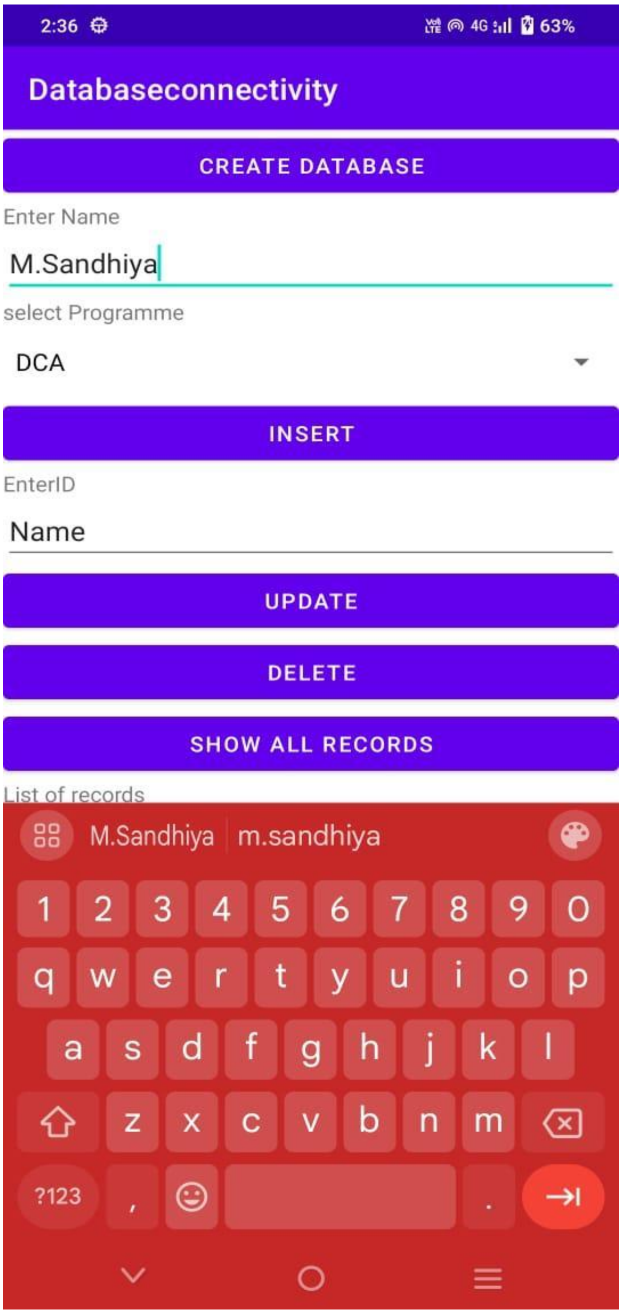
```
        android:inputType="textPersonName"
        android:text="Name"/>
    <Button
        android:id="@+id/btnUpdate"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Update"/>
    <Button
        android:id="@+id/btnDelete"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="DELETE"/>
    <Button
        android:id="@+id/btnList"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="SHOW ALL RECORDS" />
    <TextView
        android:id="@+id/txtDisplay"
        android:layout_width="match_parent"
        android:layout_height="287dp"
        android:text="List of records" />
</LinearLayout>
```

## MainActivity.java

```
package com.example.databaseconnectivity;
import androidx.appcompat.app.AppCompatActivity;
import android.content.ContentValues;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;
import android.widget.TextView;
import android.widget.Toast;
public class MainActivity extends AppCompatActivity {
    EditText txtName, txtID;
    TextView txtDisplay;
    Spinner txtProg;
    Button btnInsert, btnUpdate, btnDelete, btnList, btnCreateDB;
    SQLiteDatabase sqLiteDatabase;
    String prog[] = {"DCA", "BCA", "B.SC.(IT)", "BCA(MultiMedia)", "PGDCA",
"PGDMAD", "PGDCS", "PGDCL", "MCA"};
    ArrayAdapter<String> adapter;
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        txtID = findViewById(R.id.txtID);
        txtName = findViewById(R.id.txtName);
        txtProg = findViewById(R.id.txtProg);
        txtDisplay = findViewById(R.id.txtDisplay);
        btnDelete = findViewById(R.id.btnDelete);
        btnInsert = findViewById(R.id.btnInsert);
        btnList = findViewById(R.id.btnList);
        btnUpdate = findViewById(R.id.btnUpdate);
        btnCreateDB = findViewById(R.id.btnCreateDB);
        adapter = new ArrayAdapter<String>(getApplicationContext(),
        android.R.layout.simple_spinner_dropdown_item, prog);
        txtProg.setAdapter(adapter);
        btnCreateDB.setOnClickListener(new View.OnClickListener() {
          public void onClick(View view) {
            sqLiteDatabase = openOrCreateDatabase("StudentDB", MODE_PRIVATE,
null);
            if (sqLiteDatabase != null) {
              Toast.makeText(getApplicationContext(), "Database has been
created successfully", Toast.LENGTH_LONG).show();
              sqLiteDatabase.execSQL("CREATE TABLE students (id INTEGER
PRIMARY KEY AUTOINCREMENT, uname TEXT NOT NULL, uprog TEXT NOT NULL);");
            } else {
              Toast.makeText(getApplicationContext(), "Error",
Toast.LENGTH_LONG).show();
```

```
            }}});
        btnInsert.setOnClickListener(new View.OnClickListener() {
            public void onClick(View view) {
                sqLiteDatabase = openOrCreateDatabase("StudentDB",
MODE_PRIVATE, null);
                String name = txtName.getText().toString();
                String prog = txtProg.getSelectedItem().toString();
                ContentValues cv = new ContentValues();
                cv.put("uname", name);
                cv.put("uprog", prog);
                sqLiteDatabase.insert("Students", null, cv);
                Toast.makeText(getApplicationContext(), "Records Inserted
Successfully", Toast.LENGTH_LONG).show();
                txtID.setText("");
                txtName.setText("");
            }
        });
        btnUpdate.setOnClickListener(new View.OnClickListener() {
            public void onClick(View view) {
                sqLiteDatabase = openOrCreateDatabase("StudentDB",
MODE_PRIVATE, null);
                String id = txtID.getText().toString();
                String name = txtName.getText().toString();
                String prog = txtProg.getSelectedItem().toString();
                String whereClause = "id=?";
                String whereArgs[] = {id};
                ContentValues cv = new ContentValues();
                cv.put("uname", name);
                cv.put("uprog", prog);
                sqLiteDatabase.update("Students", cv, whereClause,
whereArgs);
                Toast.makeText(getApplicationContext(), "Records Inserted
Successfully", Toast.LENGTH_LONG).show();
                txtID.setText("");
                txtName.setText("");
            }
        });
        btnDelete.setOnClickListener(new View.OnClickListener() {
            public void onClick(View view) {
                sqLiteDatabase = openOrCreateDatabase("StudentDB",
MODE_PRIVATE, null);
                String id = txtID.getText().toString();
                String whereClause = "id=?";
                String whereArgs[] = {id};
                sqLiteDatabase.delete("Students", whereClause, whereArgs);
                Toast.makeText(getApplicationContext(), "Records Inserted
Successfully", Toast.LENGTH_LONG).show();
                txtID.setText("");
                txtName.setText("");
            }
        });
        btnList.setOnClickListener(new View.OnClickListener() {
            public void onClick(View view) {
                sqLiteDatabase = openOrCreateDatabase("StudentDB",
MODE_PRIVATE, null);
                String records = "", uname, uprog, id;
                Cursor cursor = sqLiteDatabase.query("Students", null, null,
null, null, null, null);
                while (cursor.moveToNext()) {
                    id = String.valueOf(cursor.getInt(0));
                    uname = cursor.getString(1);
                    uprog = cursor.getString(1);
                    records = records + id + " : " + uname + " :" + uprog +
"\n ";
                }
                cursor.close();
                txtDisplay.setText(records);
            }
        });
}}
```

**Output:**



2:36 63%

Databaseconnectivity

CREATE DATABASE

Enter Name

M.Sandhiya

select Programme

DCA

INSERT

EnterID

Name

UPDATE

DELETE

SHOW ALL RECORDS

List of records

**Result:**

Thus, the program has been successfully executed and output is verified.

| Exp. No: 10 | Multithreading |
|---|---|
| Date: 16.04.2024 | |

**Aim:**

To develop an android application that invokes multithreading.

**Algorithm:**

Step 1: File → NewProject

Provide the application name and Click “Next”

Step 2: Select the target android devices

Select the minimum SDK to run the application. Click “Next”.

Step 3: Choose the activity for the application (By default choose “Blank Activity). Click “Next”.

Step 4: Enter activity name and click "Finish".

Step 5: Edit the program.

Step 6: Run the application, 2-ways to run the application.

1. Running through emulator
2. Running through mobile device

**Source Code:**

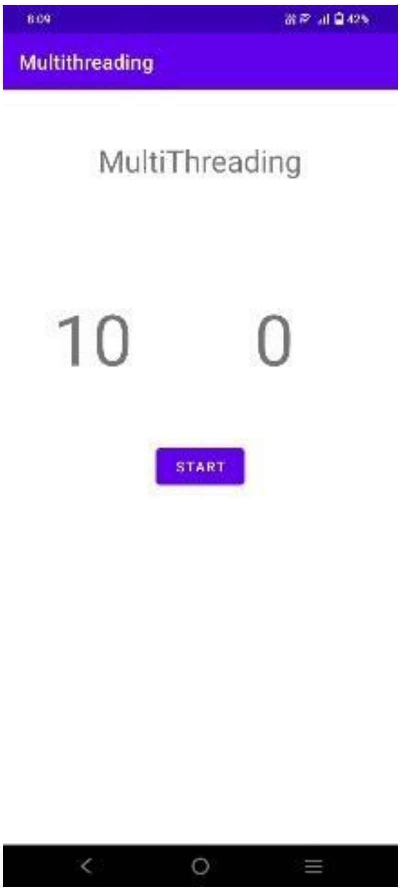
**Activity_main.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="MultiThreading"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="50dp"
        android:textSize="30dp"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="-"
        android:layout_marginLeft="50dp"
        android:layout_marginTop="200dp"
        android:textSize="70dp"
        android:id="@+id/t1"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="-"
        android:layout_marginLeft="250dp"
        android:layout_marginTop="200dp"
        android:textSize="70dp"
        android:id="@+id/t2"/>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Start"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="350dp"
        android:id="@+id/b1"/>
</RelativeLayout>
```

### MainActivity.java

```
package com.example.multithreading;
import android.os.Handler;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;
public class MainActivity extends AppCompatActivity {
    TextView t1,t2; Button b1;
    Handler hand=new Handler();
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        t1=(TextView)findViewById(R.id.t1);
        t2=(TextView)findViewById(R.id.t2);
        b1=(Button)findViewById(R.id.b1);
        b1.setOnClickListener(new View.OnClickListener() {
            public void onClick(View view) {
                t1.setText("0");
                t2.setText("10");
                hand.postDelayed(p1,1000);
                hand.postDelayed(p2,1000);
            }
        });
    }
    Runnable p1=new Runnable() {
        public void run() {
            t1.setText(""+(Integer.parseInt(t1.getText().toString())+1));
            if (Integer.parseInt(t1.getText().toString()) < 10) {
                hand.postDelayed(p1, 1000);
            }
        }
    };
    Runnable p2=new Runnable() {
        public void run() {
            t2.setText(""
                    +(Integer.parseInt(t2.getText().toString())-1));
            if (Integer.parseInt(t2.getText().toString()) > 0) {
                hand.postDelayed(p2, 1000);
            }
        }
    };
}
```

**Output:**



**Result:**

Thus, the program has been successfully executed and output is verified.

| | |
|---|---|
| **Exp. No: 11** | **GPS Location Tracking** |
| **Date: 16.04.2024** | |

**Aim:**

To develop an android application that invokes GPS location tracking.

**Algorithm:**

Step 1: File → NewProject

Provide the application name and Click “Next”

Step 2: Select the target android devices

Select the minimum SDK to run the application. Click “Next”.

Step 3: Choose the activity for the application (By default choose “Blank Activity). Click “Next”.

Step 4: Enter activity name and click "Finish".

Step 5: Edit the program.

Step 6: Run the application, 2-ways to run the application.

1. Running through emulator
2. Running through mobile device

**Source Code:**

**ActivityManifest.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/tools">
  uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
  uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
  <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.GPSLocation"
        tools:targetApi="31">
        <activity
```

```
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
            <meta-data
                android:name="android.app.lib_name"
                android:value="" />
        </activity>
    </application>
```

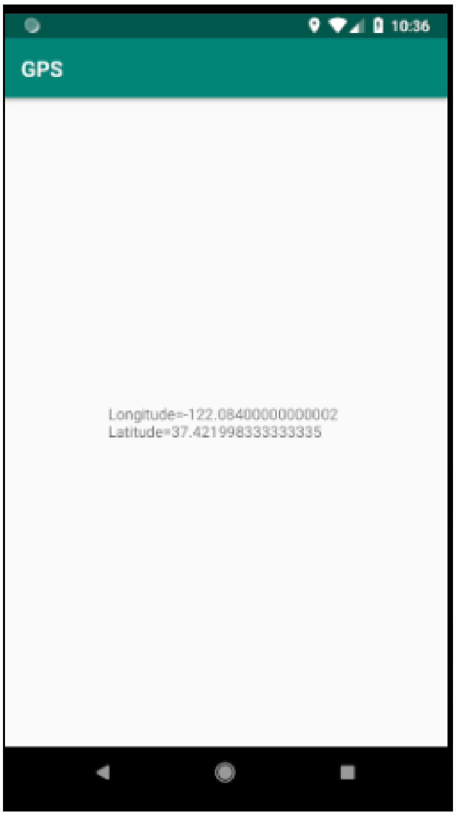
### Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        android:id="@+id/t1" />
</RelativeLayout>
```

### MainActivity.java

```
package com.example.gpslocation;
import android.Manifest;
import android.content.pm.PackageManager;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
public class MainActivity extends AppCompatActivity {
    TextView t1;
    LocationManager LM;
    LocationListener LL;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        t1=(TextView)findViewById(R.id.t1);
        String permission[]={Manifest.permission.ACCESS_FINE_LOCATION,
                        Manifest.permission.ACCESS_COARSE_LOCATION};

if(ContextCompat.checkSelfPermission(this,permission[0])!=PackageManager.PERM
ISSION_GRANTED && ContextCompat.checkSelfPermission
(this,permission[1])!=PackageManager.PERMISSION_GRANTED)
        {
            ActivityCompat.requestPermissions(this,permission,101);
        }
        LM=(LocationManager)getSystemService(LOCATION_SERVICE);
        LL = new LocationListener() {
            @Override
            public void onLocationChanged(Location location) {
                t1.setText("Longitude="+location.getLongitude()+
                        "\nLatitude="+location.getLatitude());
            }
            @Override
            public void onStatusChanged(String provider, int status, Bundle
extras) {
            }
            @Override
            public void onProviderEnabled(String provider) {
            }
            @Override
            public void onProviderDisabled(String provider) {
            }
        };
        LM.requestLocationUpdates
                (LocationManager.GPS_PROVIDER,5000,10,LL);
    }
}
```

**Output:**



```
Longitude=-122.08400000000002
Latitude=37.421998333333335
```

**Result:**

Thus, the program has been successfully executed and output is verified.

| **Exp. No: 12** | **SMS Alert** |
|---|---|
| **Date: 23.04.2024** | |

**Aim:**

To develop an android application that creates an alert upon receiving a message.

**Algorithm:**

Step 1: File → NewProject

Provide the application name and Click “Next”

Step 2: Select the target android devices

Select the minimum SDK to run the application. Click “Next”.

Step 3: Choose the activity for the application (By default choose “Blank Activity). Click “Next”.

Step 4: Enter activity name and click "Finish".

Step 5: Edit the program.

Step 6: Run the application, 2-ways to run the application.

1. Running through emulator
2. Running through mobile device

**Source Code:**

**Activity_main.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Show Notification"
        android:layout_marginLeft="20dp"
        android:layout_marginTop="50dp"
        android:id="@+id/b1" />
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hide Notification"
        android:layout_marginLeft="20dp"
        android:layout_marginTop="150dp"
        android:id="@+id/b2" />
</RelativeLayout>
```

### ActivityNotification.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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=".NotificationActivity">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        android:text="This is an information on the notification" />
</RelativeLayout>
```

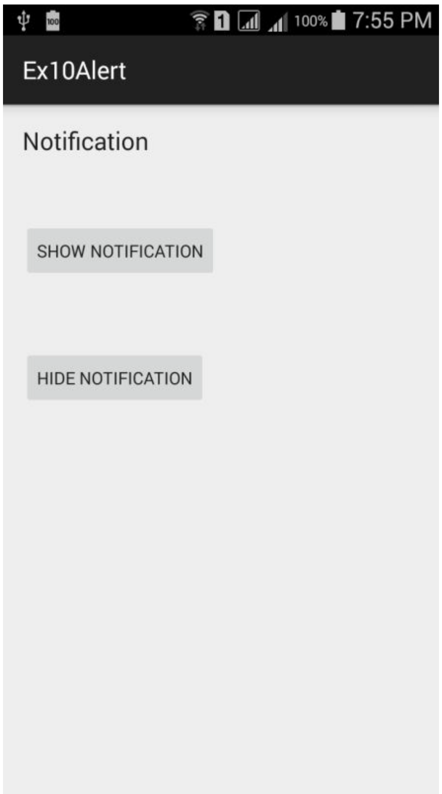
### MainActivity.java

```
package com.example.smsalert;
import static android.icu.lang.UCharacter.GraphemeClusterBreak.T;
import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.app.PendingIntent;
import android.content.Intent;
import android.os.Build;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.NotificationCompat;
public class MainActivity extends AppCompatActivity {
    Button b1,b2;
    NotificationManager nm;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        b1=(Button)findViewById(R.id.b1);
        b2=(Button)findViewById(R.id.b2);
        nm=(NotificationManager)getSystemService(NOTIFICATION_SERVICE);
        final String CHANNEL_ID = "my_channel_01";
        b1.setOnClickListener(new View.OnClickListener() {
            @Override
          public void onClick(View v) {
             CharSequence name="my_notification";
             NotificationChannel nc= null;
             if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O)
                nc = new NotificationChannel(CHANNEL_ID,name,
NotificationManager.IMPORTANCE_DEFAULT);
             if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O)
                    nc.setDescription("New Notification");
             if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O)
                    nm.createNotificationChannel(nc);
   Intent i=new Intent(MainActivity.this,NotificationActivity.class);
             PendingIntent
pi=PendingIntent.getActivity(MainActivity.this,0,i,PendingIntent.FLAG_UPDATE_
CURRENT);
NotificationCompat.Builder builder=
      new NotificationCompat.Builder (MainActivity.this,CHANNEL_ID)
      .setContentTitle("New Message")
      .setContentText("You have an unread message")
      .setSmallIcon(R.mipmap.ic_launcher)
      .setContentIntent(pi);
      nm.notify(1,builder.build());
   }        });
   b2.setOnClickListener(new View.OnClickListener() {
   @Override
      public void onClick(View v) {
         nm.cancel(1);
        }
    });
}}
```

### Notification.java

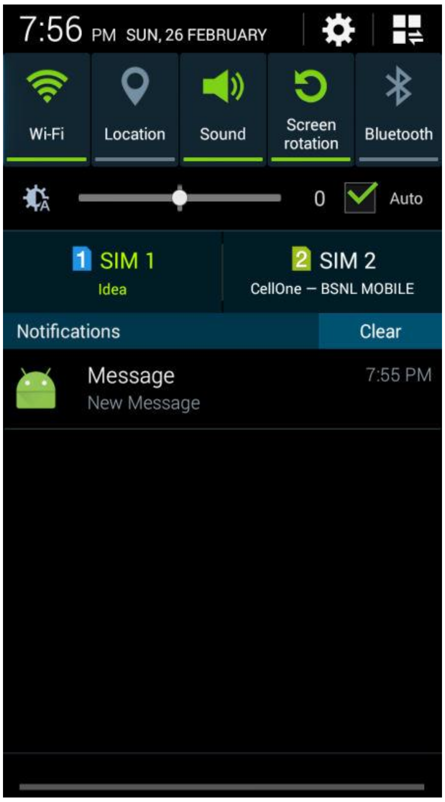
```
package com.example.smsalert;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
public class NotificationActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activitynotification);
    }
}
```

### Output:

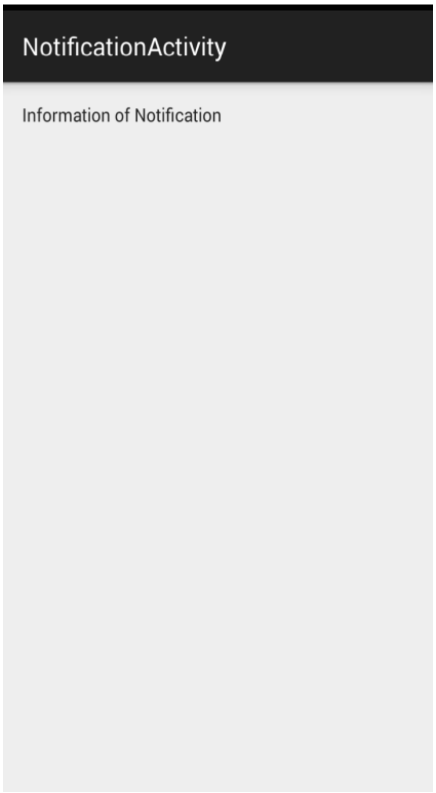


Ex10Alert

Notification



Message
New Message
7:55 PM



NotificationActivity

Information of Notification

### Result:

Thus, the program has been successfully executed and output is verified.

| Exp. No: 13 | Alarm Clock |
|---|---|
| Date: 30.04.2024 | |

**Aim:**

To develop an android application that invokes multithreading.

**Algorithm:**

Step 1: File → NewProject

Provide the application name and Click “Next”

Step 2: Select the target android devices

Select the minimum SDK to run the application. Click “Next”.

Step 3: Choose the activity for the application (By default choose “Blank Activity). Click “Next”.

Step 4: Enter activity name and click "Finish".

Step 5: Edit the program.

Step 6: Run the application, 2-ways to run the application.

1. Running through emulator
2. Running through mobile device

**Source Code:**

**Activity_main.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Set Alarm"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        android:id="@+id/b1" />
</RelativeLayout>
```

### Activityalarmreceiver.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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=".AlarmReceiver">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Alarm Started"
        android:layout_centerVertical="true"
        android:layout_centerHorizontal="true"
        />
</RelativeLayout>
```

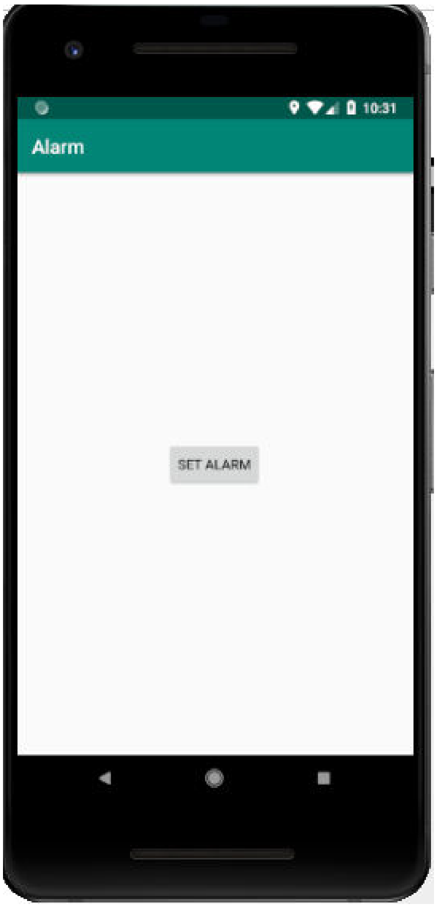
### MainActivity.java

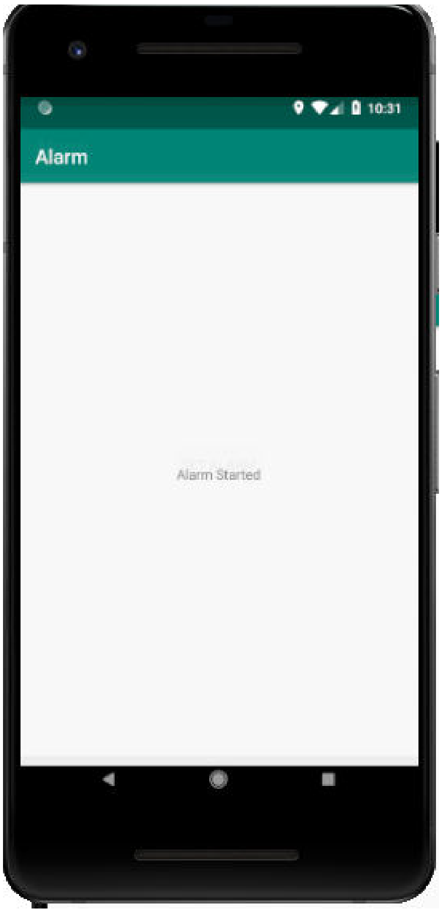
```
package com.example.alaram;
import android.app.AlarmManager;
import android.app.PendingIntent;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
public class MainActivity extends AppCompatActivity {
    Button b1;
    AlarmManager am;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        b1=(Button)findViewById(R.id.b1);
        am=(AlarmManager)getSystemService(ALARM_SERVICE);
        b1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent i=new Intent(MainActivity.this,AlarmReceiver.class);
                PendingIntent

pi=PendingIntent.getActivity(MainActivity.this,0,i,PendingIntent.FLAG_UPDATE_
CURRENT
                );

am.set(AlarmManager.RTC_WAKEUP,System.currentTimeMillis()+5000,pi);
                Toast.makeText(getApplicationContext(),"Alarm will start in 5
seconds",Toast.LENGTH_LONG).show();
            }
        });
}}
```

## AlarmReceiver.java

```
package com.example.alaram;
import android.media.Ringtone;
import android.media.RingtoneManager;
import android.net.Uri;
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;
public class AlarmReceiver extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activityalaramreceiver);
        Uri u=RingtoneManager.getDefaultUri(RingtoneManager.TYPE_ALARM);
        Ringtone r=RingtoneManager.getRingtone(this,u);
        r.play();
    }
}
```

**Output:**





Alarm Started

**Result:**

Thus, the program has been successfully executed and output is verified.