1. Install / Configure java development kit (JDK), android studio and android SDK.

1. Java Development Kit (JDK):

- Download the latest JDK from Oracle's website: Java SE Downloads: https://www.oracle.com/java/technologies/downloads/
- Choose the appropriate version for your operating system (Windows, macOS, or Linux). A minimum of JDK 11 is recommended for Android Studio.
- Follow the installation instructions for your OS.

2. Android Studio:

- Download the latest Android Studio from the official Android developer website: Android Studio download: https://developer.android.com/
- Choose the appropriate installer for your OS.
- Run the downloaded installer and follow the on-screen instructions. During installation, Android Studio will check for a compatible JDK. If one is not found, you'll be prompted to locate it.

3. Android SDK:

- Android Studio comes bundled with the SDK manager. Launch Android Studio.
- Go to **Tools > SDK Manager**.
- In the SDK manager window, you can download and install various Android SDK components like platform versions, tools, and libraries.
- It's recommended to install the latest Android SDK platform (API level) and the Android Emulator for testing your apps.

Additional Configuration:

- **Environment Variables:** For some tools to function correctly, you might need to set environment variables pointing to the installation directories of JDK and Android SDK. Consult the documentation for your specific OS on how to set environment variables.

2. Configure android development tools (ADT) plug-in and create android virtual device.

While the ADT plugin was previously used with Eclipse for Android development, it's important to note that Google no longer officially supports ADT. Android Studio is the current recommended IDE for Android development, and it has built-in functionality for managing virtual devices.

Here's how to create an Android virtual device (AVD) directly in Android Studio:

1. **Launch Android Studio.**
2. **Open the AVD Manager:**
    - From the Welcome screen, select **More Actions > Virtual Device Manager**.
    - If you have a project open, go to **View > Tool Windows > Device Manager**.
3. **Click "Create Device".**
4. **Configure your AVD:**
    - **Hardware Profile:** Select a device profile (phone, tablet, etc.) that represents the target device you want to emulate. You can customize these profiles further.
    - **System Image:** Choose an appropriate Android system image (API level) that your app targets. The "Recommended" tab provides good starting points. System images with Google APIs allow access to Google Play services.
    - **Name:** Assign a recognizable name to your AVD configuration.
    - **Other options:** You can configure additional options like RAM size, storage, and device skin (appearance).
5. **Click "Finish"** to create your AVD.

Here are some resources for reference:

- Create and manage virtual devices: https://developer.android.com/studio/run/managing-avds
- Android Studio documentation: https://developer.android.com/

3. Develop a program to display Hello World on Screen.

Here's how to develop a program to display "Hello World" on the screen in Java for an Android Studio mobile application:

1. Create a new Android Studio Project:

- Launch Android Studio.
- Click "Start a new Android Studio project".
- Choose a name for your project (e.g., HelloWorldApp).
- Select "Empty Activity" as the template and click "Next".
- Configure the project details (package name, minimum SDK, etc.) and click "Finish".

2. Modify the Activity Layout (activity_main.xml):

- Go to the "res/layout" folder in the project directory.
- Open the "activity_main.xml" file.
- This file defines the user interface (UI) elements for your app's main screen.
- We'll use a TextView element to display "Hello World".

Here's the modified code for "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"
  tools:context=".MainActivity">

  <TextView
    android:id="@+id/hello_text"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Hello World!"
    android:gravity="center"
    android:textSize="24sp" />

</LinearLayout>
```

Explanation:

- We define a LinearLayout as the root element, which arranges elements vertically.
- Inside the layout, we have a TextView element with the ID "hello_text".
- The android:text attribute sets the text displayed to "Hello World!".
- Other attributes like android:gravity and android:textSize customize the text appearance.

3. Modify the Activity Class (MainActivity.java):

- Go to the "java" folder within your project directory and open "MainActivity.java".
- This file contains the code for the main activity of your app.

Here's the code for "MainActivity.java":

```
package com.example.helloworldapp; // Replace with your package name

import android.os.Bundle;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

  @Override
```

```
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);

      // Reference the TextView element from the layout
      TextView helloTextView = findViewById(R.id.hello_text);
   }
}
```

Explanation:

- We extend the AppCompatActivity class for our activity.
- The onCreate method is called when the activity is first created.
- We set the layout for the activity using setContentView(R.layout.activity_main).
- We use findViewById(R.id.hello_text) to find the TextView element with the ID "hello_text" defined in the layout.

4. Run the App:

- In Android Studio, click the green "Run" button (play triangle icon).
- Select an emulator or connected Android device to run the app on.
- You should see the "Hello World!" text displayed on the screen of your emulator/device.

4. Develop a program to implement linear layout and absolute layout.

```
<?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">

   <LinearLayout
      android:layout_width="match_parent"
      android:layout_height="0dp"
      android:layout_weight="1"
      android:orientation="horizontal"
      android:gravity="center">

      <TextView
         android:id="@+id/text_linear_1"
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:text="This is in Linear Layout"
         android:textSize="18sp"
```

```
            android:padding="10dp"
            android:background="#CCC" />

        <TextView
            android:id="@+id/text_linear_2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Another Text"
            android:textSize="18sp"
            android:padding="10dp"
            android:background="#EEE" />

    </LinearLayout>

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1">

        <TextView
            android:id="@+id/text_absolute"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Absolute Positioned Text"
            android:textSize="20sp"
            android:layout_centerHorizontal="true"
            android:layout_marginTop="50dp"
            android:background="#FFC" />

    </RelativeLayout>

</LinearLayout>
```

Explanation:

- We use a main LinearLayout with vertical orientation.
- Inside the main layout, we have two child layouts:
    - **LinearLayout Section:** This layout is horizontal and uses android:layout_weight="1" to distribute remaining space equally among its children. TextViews display content within the linear layout.
    - **RelativeLayout Section:** This layout uses android:layout_weight="1" to occupy half the vertical space. A TextView with specific positioning using layout_centerHorizontal and layout_marginTop is placed within the relative layout.

5. Develop a program to implement frame layout and table layout.

activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
  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">

  <FrameLayout
    android:id="@+id/frame_layout"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginTop="16dp"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintTop_toTopOf="parent">

    <ImageView
      android:layout_width="100dp"
      android:layout_height="100dp"
      android:src="@drawable/your_image"
      android:layout_gravity="center" />

  </FrameLayout>

  <TableLayout
    android:id="@+id/table_layout"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    app:layout_constraintLeft_toRightOf="@id/frame_layout"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent">

    <TableRow
      android:layout_width="wrap_content"
      android:layout_height="wrap_content">

      <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Name:" />

      <EditText
        android:id="@+id/name_edit_text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:hint="Enter your name" />
    </TableRow>

    <TableRow>
      android:layout_width="wrap_content"
      android:layout_height="wrap_content">

      <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Age:" />

      <EditText
        android:id="@+id/age_edit_text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:hint="Enter your age"
        android:inputType="number" />

    </TableRow>

  </TableLayout>

</android.support.constraint.ConstraintLayout>
```

MainActivity.java:

```
package com.example.your.package.name;

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
  }
}
```

Explanation:

- The activity_main.xml file defines the user interface (UI) of your app.
  - We use ConstraintLayout as the root layout for better flexibility.
  - The FrameLayout holds an ImageView which will be displayed on top of each other (only the topmost view will be visible).
  - The TableLayout defines a two-row table with labels and EditText fields for user input.

- The MainActivity.java file extends AppCompatActivity and sets the layout (activity_main.xml) for the activity.

6. Develop a program to implement Text View and Edit Text.

```
<?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/idRLContainer"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        tools:context=".MainActivity">

        <!--on below line we are creating
                a text for our app-->
        <TextView
                android:id="@+id/idTVHeading"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_above="@id/idTVMsg"
                android:layout_centerInParent="true"
                android:layout_margin="20dp"
                android:gravity="center"
                android:padding="10dp"
                android:text="TextView in Android"
                android:textAlignment="center"
                android:textColor="@color/black"
                android:textSize="20sp"
                android:textStyle="bold" />

        <!--on below line we are creating a text view-->
        <TextView
                android:id="@+id/idTVMsg"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_centerInParent="true"
                android:layout_margin="20dp"
                android:gravity="center"
                android:padding="10dp"
                android:textAlignment="center"
                android:textColor="@color/black"
                android:textSize="20sp"
                android:textStyle="bold" />
```

```
</RelativeLayout>

Java
package com.gtappdevelopers.kotlingfgproject;

import android.os.Bundle;
import android.widget.TextView;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

	// on below line we are creating variables.
	private TextView msgTV;

	@Override
	protected void onCreate(Bundle savedInstanceState) {
		super.onCreate(savedInstanceState);
		setContentView(R.layout.activity_main);

		// on below line we are initializing our variables.
		msgTV = findViewById(R.id.idTVMsg);

		// on below line we are setting message to our text view.
		msgTV.setText("Welcome to Geeks for Geeks");
	}
}
```

7. Develop a program to implement Button, Image Button, and Toggle Button.

activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
  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:id="@+id/normal_button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Click Me"
    android:layout_marginStart="16dp"
    android:layout_marginTop="16dp"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

  <ImageButton
    android:id="@+id/image_button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginTop="16dp"
    app:layout_constraintLeft_toRightOf="@id/normal_button"
    app:layout_constraintRight_toLeftOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    android:src="@drawable/your_image" />

  <ToggleButton
    android:id="@+id/toggle_button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textOff="Off"
    android:textOn="On"
    android:layout_marginStart="16dp"
    android:layout_marginTop="16dp"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintTop_toBottomOf="@id/normal_button" />

</android.support.constraint.ConstraintLayout>
```

MainActivity.java:

```
package com.example.your.package.name;

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.widget.Toast;
import android.widget.ToggleButton;

public class MainActivity extends AppCompatActivity {

    ToggleButton toggleButton;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        // Get reference to ToggleButton
        toggleButton = findViewById(R.id.toggle_button);

        // Set click listener for ToggleButton
        toggleButton.setOnCheckedChangeListener((buttonView, isChecked) -> {
            String message;
            if (isChecked) {
```

```
            message = "Toggle Button is On";
        } else {
            message = "Toggle Button is Off";
        }
        Toast.makeText(getApplicationContext(), message, Toast.LENGTH_SHORT).show();
      });
   }
}
```

Explanation:

- The activity_main.xml file defines the UI with a Button, ImageButton, and ToggleButton.
    - Replace "your_image" with the actual image resource name in your project's drawable folder.
- The MainActivity.java file:
    - Extends AppCompatActivity and sets the layout.
    - Gets a reference to the ToggleButton using findViewById.
    - Sets an onCheckedChangeListener for the ToggleButton. This listener will be called whenever the toggle button's state changes (checked/unchecked).
    - Inside the listener, a Toast message is displayed based on the current state of the toggle button.

8. Develop a program to implement login window using above UI controls.

```
XML
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
  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=".LoginActivity">

  <TextView
    android:id="@+id/welcome_text"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Welcome!"
    android:textSize="24sp"
    android:layout_marginStart="16dp"
    android:layout_marginTop="16dp"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
  <EditText
    android:id="@+id/username_edit_text"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:hint="Username"
    android:inputType="text"
    android:layout_marginStart="16dp"
    android:layout_marginTop="16dp"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toBottomOf="@id/welcome_text" />

  <EditText
    android:id="@+id/password_edit_text"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:hint="Password"
    android:inputType="textPassword"
    android:layout_marginStart="16dp"
    android:layout_marginTop="16dp"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toBottomOf="@id/username_edit_text" />

  <Button
    android:id="@+id/login_button"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:text="Login"
    android:layout_marginStart="16dp"
    android:layout_marginTop="16dp"
    android:layout_marginBottom="16dp"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintBottom_toBottomOf="parent" />

</android.support.constraint.ConstraintLayout>
```

Java

```
package com.example.your.package.name;

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.widget.EditText;
import android.widget.Toast;

public class LoginActivity extends AppCompatActivity {
  private EditText usernameEditText;
  private EditText passwordEditText;

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    // Get references to EditTexts
    usernameEditText = findViewById(R.id.username_edit_text);
    passwordEditText = findViewById(R.replace id="@+id/password_edit_text", ""));  //
Corrected missing closing parenthesis

    // Get reference to login button (implicit)

    // (Optional) Set dummy credentials for testing (replace with actual authentication logic)
    String username = "admin";
    String password = "password";

    findViewById(R.id.login_button).setOnClickListener(v -> {
      String enteredUsername = usernameEditText.getText().toString();
      String enteredPassword = passwordEditText.getText().toString();

      if (enteredUsername.equals(username) && enteredPassword.equals(password)) {
        Toast.makeText(getApplicationContext(), "Login Successful!",
Toast.LENGTH_SHORT).show();
        // (Extend this logic to navigate to another activity or perform other actions)
      } else {
        Toast.makeText(getApplicationContext(), "Invalid username or password",
Toast.LENGTH_SHORT).show();
      }
    });
  }
}
```

9. Develop a program to implement Checkbox.

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"
  tools:context=".MainActivity">

  <TextView
```

```
        android:id="@+id/text_view"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Remember me?" />

    <CheckBox
        android:id="@+id/checkbox"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Yes" />

</LinearLayout>
```

JAVA

```
package com.example.checkboxexample; // Replace with your package name

import android.os.Bundle;
import android.widget.CheckBox;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    private CheckBox checkBox;
    private TextView textView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        checkBox = findViewById(R.id.checkbox);
        textView = findViewById(R.id.text_view);

        // Handle checkbox click event
        checkBox.setOnCheckedChangeListener((buttonView, isChecked) -> {
            if (isChecked) {
                textView.setText("Remember me selected!");
            } else {
                textView.setText("Remember me not selected.");
            }
        });
    }
}
```

10. Develop a program to implement Radio Button and Radio Group.

XML

```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"
  tools:context=".MainActivity">

  <TextView
    android:id="@+id/text_view"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="What's your favorite color?" />

  <RadioGroup
    android:id="@+id/radio_group"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">

    <RadioButton
      android:id="@+id/radio_red"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Red" />

    <RadioButton
      android:id="@+id/radio_green"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Green" />

    <RadioButton
      android:id="@+id/radio_blue"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Blue" />
  </RadioGroup>

  <TextView
    android:id="@+id/selected_text"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="No color selected yet." />

</LinearLayout>
```

JAVA

```
package com.example.radiogroupexample; // Replace with your package name

import android.os.Bundle;
import android.widget.RadioButton;
import android.widget.RadioGroup;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    private RadioGroup radioGroup;
    private TextView selectedTextView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        radioGroup = findViewById(R.id.radio_group);
        selectedTextView = findViewById(R.id.selected_text);

        // Set a listener on the RadioGroup
        radioGroup.setOnCheckedChangeListener((group, checkedId) -> {
            RadioButton radioButton = findViewById(checkedId);
            String selectedColor = radioButton.getText().toString();
            selectedTextView.setText("Selected color: " + selectedColor);
        });
    }
}
```

11. Develop a program to implement Progress Bar.

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"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/text_view"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Downloading..." />

    <ProgressBar
        android:id="@+id/progress_bar"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:style="@style/Widget.AppCompat.ProgressBar.Horizontal"
        android:max="100" />

    <TextView
        android:id="@+id/progress_text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Progress: 0%" />

</LinearLayout>
```

JAVA

```
package com.example.progressbardemo; // Replace with your package name

import android.os.Bundle;
import android.os.Handler;
import android.widget.ProgressBar;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    private ProgressBar progressBar;
    private TextView progressTextView;
    private int progressStatus = 0;

    private Handler handler = new Handler();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        progressBar = findViewById(R.id.progress_bar);
        progressTextView = findViewById(R.id.progress_text);

        // Simulate download progress
        new Thread(() -> {
            while (progressStatus < 100) {
                progressStatus += 1;
                handler.post(() -> {
                    progressBar.setProgress(progressStatus);
                    progressTextView.setText("Progress: " + progressStatus + "%");
                });
                try {
                    Thread.sleep(100); // Simulate some delay
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
        }).start();
    }
}
```

12. Develop a program to implement List View, Grid View, Image View and Scroll View.

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">

    <TextView
        android:id="@+id/title_text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="List View Example" />

    <ListView
        android:id="@+id/list_view"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />

    <TextView
        android:id="@+id/grid_title_text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Grid View Example" />

    <GridView
        android:id="@+id/grid_view"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:numColumns="2" />  <TextView
    android:id="@+id/image_title_text"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Image View Example" />

  <LinearLayout  android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center">
    <ImageView
      android:id="@+id/image_view"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:src="@drawable/your_image"  />
  </LinearLayout>

  <TextView
    android:id="@+id/scroll_title_text"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Scroll View Example" />

  <ScrollView
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView  android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="This is a scrollable text content. You can add more views here for
extensive scrolling content." />
  </ScrollView>

</LinearLayout>
```

JAVA

```
package com.example.listviewgridviewexample; // Replace with your package name

import android.os.Bundle;
import android.widget.ArrayAdapter;
import android.widget.GridView;
import android.widget.ImageView;
import android.widget.ListView;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

  private ListView listView;
  private GridView gridView;
  private ImageView imageView;

  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    // Initialize views
    listView = findViewById(R.id.list_view);
    gridView = findViewById(R.id.grid_view);
    imageView = findViewById(R.id.image_view);

    // Set up List View
    String[] items = {"Item 1", "Item 2", "Item 3"}; // Replace with your data source
    ArrayAdapter<String> adapter = new ArrayAdapter<>(this,
android.R.layout.simple_list_item_1, items);
    listView.setAdapter(adapter);

    // Set up Grid View
    String[] gridItems = {"
```

13. Develop a program for providing Bluetooth connectivity.

XML

```
<uses-permission android:name="android.permission.BLUETOOTH" />
<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
```

JAVA

```
public class BluetoothActivity extends AppCompatActivity {

  private static final int REQUEST_ENABLE_BT = 1;
  private static final UUID MY_UUID = UUID.fromString("your_unique_uuid_here"); //
Replace with your desired UUID

  private BluetoothAdapter bluetoothAdapter;
  private TextView statusTextView; // To display connection status

  private final BroadcastReceiver discoveryReceiver = new BroadcastReceiver() {
    @Override
    public void onReceive(Context context, Intent intent) {
      String action = intent.getAction();
      if (BluetoothDevice.ACTION_FOUND.equals(action)) {
        // Get the BluetoothDevice object from the Intent
```

```
                BluetoothDevice device =
intent.getParcelableExtra(BluetoothDevice.EXTRA_DEVICE);
                // Add the device to your list or perform other actions (e.g., connect)
                connectToDevice(device);
            }
        }
    };

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        statusTextView = findViewById(R.id.status_text_view); // Update with your TextView ID

        bluetoothAdapter = BluetoothAdapter.getDefaultAdapter();

        if (bluetoothAdapter == null) {
            statusTextView.setText("Device doesn't support Bluetooth");
        } else {
            checkBluetoothEnabled();
        }
    }

    private void checkBluetoothEnabled() {
        if (!bluetoothAdapter.isEnabled()) {
            Intent enableBtIntent = new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
            startActivityForResult(enableBtIntent, REQUEST_ENABLE_BT);
        } else {
            startDiscovery();
        }
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if (requestCode == REQUEST_ENABLE_BT) {
            if (resultCode == RESULT_OK) {
                startDiscovery();
            } else {
                // Handle user refusing to enable Bluetooth
                statusTextView.setText("Bluetooth permission denied");
            }
        }
    }

    private void startDiscovery() {
        // Register the broadcast receiver before discovery
        IntentFilter filter = new IntentFilter(BluetoothDevice.ACTION_FOUND);
        registerReceiver(discoveryReceiver, filter);

        bluetoothAdapter.startDiscovery();
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
        // Unregister the receiver when done
        unregisterReceiver(discoveryReceiver);

        // Stop discovery if ongoing
        if (bluetoothAdapter.isDiscovering()) {
            bluetoothAdapter.cancelDiscovery();
        }
    }

    private void connectToDevice(BluetoothDevice device) {
        // Example for insecure connection (replace with secure connection if needed)
        try {
            BluetoothSocket socket =
device.createInsecureRfcommSocketToServiceRecord(MY_UUID);
            socket.connect();
            // Handle successful connection (e.g., use streams for data exchange)
            statusTextView.setText("Connected to " + device.getName());
            // Implement data transfer logic here using streams (InputStream/OutputStream) on
the connected socket
        } catch (IOException e) {
            e.printStackTrace();
            statusTextView.setText("Connection failed: " + e.getMessage());
        }
    }
}
```

14. Develop a program for animation.

1. Animation with XML:

fade_in.xml:

XML

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">
   <alpha
      android:fromAlpha="0.0"
      android:toAlpha="1.0"
      android:duration="1000" />
```

```
</set>
```

activity_main.xml:

```
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">

    <Button
        android:id="@+id/animate_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Animate" />

    <TextView
        android:id="@+id/text_view"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="This text will fade in"
        android:alpha="0.0"  />

</LinearLayout>
```

MainActivity.java:

```
Java
package com.example.animationexample; // Replace with your package name

import android.os.Bundle;
import android.view.View;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.widget.Button;
import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    private Button animateButton;
    private TextView textView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        animateButton = findViewById(R.id.animate_button);
        textView = findViewById(R.id.text_view);

        animateButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Animation fadeInAnimation = AnimationUtils.loadAnimation(getApplicationContext(),
R.anim.fade_in);
                textView.startAnimation(fadeInAnimation);
            }
        });
    }
}
```

2. Animation with ObjectAnimator (Java code):

MainActivity.java (modified):

Java

```
package com.example.animationexample; // Replace with your package name

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

import androidx.appcompat.app.AppCompatActivity;
import android.view.animation.ObjectAnimator;

public class MainActivity extends AppCompatActivity {

    private Button animateButton;
    private TextView textView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        animateButton = findViewById(R.id.animate_button);
        textView = findViewById(R.id.text_view);

        animateButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // Animate alpha from 0.0 to 1.0 over 1 second
                ObjectAnimator fadeInAnim = ObjectAnimator.ofFloat(textView, "alpha", 0.0f, 1.0f);
                fadeInAnim.setDuration(1000);
                fadeInAnim.start();
            }
        });
    }
}
```

15. Perform Async task using SQLite.

DbHelper.java

```java
package com.example.sqlitetask; // Replace with your package name

import android.content.ContentValues;
import android.content.Context;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

public class DbHelper extends SQLiteOpenHelper {

    private static final String DATABASE_NAME = "my_database.db";
    private static final int DATABASE_VERSION = 1;
    private static final String TABLE_NAME = "items";
    private static final String COLUMN_ID = "_id"; // Use AUTOINCREMENT for ID
    private static final String COLUMN_NAME = "name";

    public DbHelper(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
        String createTable = "CREATE TABLE " + TABLE_NAME + "("
                + COLUMN_ID + " INTEGER PRIMARY KEY AUTOINCREMENT, "
                + COLUMN_NAME + " TEXT NOT NULL);";
        db.execSQL(createTable);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        // Handle database schema upgrades if needed
    }

    public void insertItemAsync(String name, InsertDataCallback callback) {
        new InsertDataTask(this, name, callback).execute();
    }

    private class InsertDataTask extends AsyncTask<Void, Void, Long> {

        private DbHelper dbHelper;
        private String name;
        private InsertDataCallback callback;

        public InsertDataTask(DbHelper dbHelper, String name, InsertDataCallback callback) {
            this.dbHelper = dbHelper;
            this.name = name;
            this.callback = callback;
        }

        @Override
        protected Long doInBackground(Void... voids) {
            SQLiteDatabase db = dbHelper.getWritableDatabase();
            ContentValues values = new ContentValues();
            values.put(COLUMN_NAME, name);
            long newRowId = db.insert(TABLE_NAME, null, values);
            db.close();
            return newRowId;
        }

        @Override
        protected void onPostExecute(Long newRowId) {
            super.onPostExecute(newRowId);
            if (callback != null) {
                if (newRowId > -1) {
                    callback.onDataInserted(newRowId);
                } else {
                    callback.onInsertError();
                }
            }
        }
    }

    public interface InsertDataCallback {
        void onDataInserted(long newRowId);
        void onInsertError();
    }
}
```

MainActivity.java

```
package com.example.sqlitetask; // Replace with your package name

import android.os.Bundle;
import android.widget.Button;
import android.widget.Toast;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity implements
DbHelper.InsertDataCallback {

    private Button insertButton;
    private DbHelper dbHelper;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        insertButton = findViewById(R.id.insert_button);
        dbHelper = new DbHelper(this);

        insertButton.setOnClickListener(v -> {
            String itemName = "Item Name"; // Replace with your data
            dbHelper.insertItemAsync(itemName, this);
        });
    }

    @Override
    public void onDataInserted(long newRowId) {
        Toast.makeText(this, "Data inserted with ID: " + newRowId, Toast.
```

16. Develop program to:
a) Send SMS b) Receive SMS.

1. Manifest Permissions (AndroidManifest.xml):

XML

```
<uses-permission android:name="android.permission.BROADCAST_RECEIVER" />
```

2. Broadcast Receiver Class (SmsReceiver.java):

This class intercepts incoming SMS broadcasts and displays a Toast notification:

Java

```
package com.example.smslistener; // Replace with your package name

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.widget.Toast;

public class SmsReceiver extends BroadcastReceiver {

    @Override
    public void onReceive(Context context, Intent intent) {
        if (intent.getAction().equals("android.provider.Telephony.SMS_RECEIVED")) {
            // Simulate receiving SMS by showing a Toast notification
            Toast.makeText(context, "New SMS received!", Toast.LENGTH_SHORT).show();
        }
    }
}
```

3. Activity Class (MainActivity.java):

This class registers the Broadcast Receiver:

Java

```
package com.example.smslistener; // Replace with your package name

import android.content.BroadcastReceiver;
import android.content.IntentFilter;
import android.os.Bundle;
import android.widget.Button;

import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {

    private Button simulateSmsButton;
    private BroadcastReceiver smsReceiver;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        simulateSmsButton = findViewById(R.id.simulate_sms_button);
        smsReceiver = new SmsReceiver();

        // Register the BroadcastReceiver to listen for SMS received events
        registerReceiver(smsReceiver, new IntentFilter("android.provider.Telephony.SMS_RECEIVED"));

        simulateSmsButton.setOnClickListener(v -> {
            // Simulate sending SMS by showing a Toast notification
            Toast.makeText(this, "SMS sent (simulated)", Toast.LENGTH_SHORT).show();
        });
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
        // Unregister the receiver when activity is destroyed
        unregisterReceiver(smsReceiver);
    }
}
```

4. Layout file (activity_main.xml - optional):

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">

    <Button
        android:id="@+id/simulate_sms_button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Simulate Send SMS" />

</LinearLayout>
```

17. Develop program to send and receive e-mail.

```
public class MainActivity extends AppCompatActivity {

    private Button sendEmailButton;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        sendEmailButton = findViewById(R.id.send_email_button);

        sendEmailButton.setOnClickListener(v -> {
            String recipient = "recipient@example.com"; // Replace with recipient email
            String subject = "Email Subject";
            String body = "This is an email sent from your Android app.";

            Intent emailIntent = new Intent(Intent.ACTION_SEND);
            emailIntent.putExtra(Intent.EXTRA_EMAIL, new String[]{recipient});
            emailIntent.putExtra(Intent.EXTRA_SUBJECT, subject);
            emailIntent.putExtra(Intent.EXTRA_TEXT, body);
            emailIntent.setType("message/rfc822"); // MIME type for email
            startActivity(Intent.createChooser(emailIntent, "Choose Email Client"));
        });
    }
}
```

18. Develop program to build a Camera.

AndroidManifest.xml

```
<uses-permission android:name="android.permission.CAMERA" />
<uses-feature android:name="android.hardware.camera" />
```

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">

    <SurfaceView
        android:id="@+id/camera_preview"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

    <Button
        android:id="@+id/capture_button"
        android:layout_width="wrap_content"
        android
```