Program 1: Program to display your name

Aim: To write a program to display your name

Procedure:
1. Create a new Android project in Android Studio.
2. Design the layout in activity_main.xml with a TextView.
3. Set the text property to your name.
4. In MainActivity.java, use setContentView() to load the layout.
5. Run the app → Your name appears on the screen.

Program 2: Program to change the screen orientation

Aim: To write a program to change the screen orientation

Procedure:
1. Create a new project.
2. In MainActivity.java, call setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_LANDSCAPE).
3. Add a TextView in activity_main.xml.
4. Run the app → Screen is forced to landscape orientation.

Program 3: Program to display a toggle button

Aim: To write a program to display a toggle button

Procedure:
1. Add a ToggleButton and TextView in activity_main.xml.
2. In MainActivity.java, initialize them with findViewById().
3. Set OnClickListener on the toggle button.
4. Change TextView text depending on ON/OFF state.
5. Run the app → Text changes when toggle is clicked.

Program 4: Program to create radio group and radio button

Aim: To write a program to create radio group and radio button

Procedure:
1. Add a RadioGroup with multiple RadioButtons in activity_main.xml.
2. Add a TextView to display the selected option.
3. In MainActivity.java, use setOnCheckedChangeListener for the group.
4. On selection, set the chosen button’s text into the TextView.
5. Run the app → Selected radio option is displayed.

Program 5: Program to display progress bar

Aim: To write a program to display progress bar

Procedure:
1. Add a ProgressBar and a Button in activity_main.xml.
2. Set the progress bar’s visibility to gone initially.
3. In MainActivity.java, on button click → toggle visibility of ProgressBar.
4. Also change button text between “Show” and “Hide”.
5. Run the app → Button controls visibility of progress bar.

Program 6: Program to create Auto Complete Text View.

Aim: To write a program to create Auto Complete Text View.

Procedure:
1. Add an AutoCompleteTextView in activity_main.xml.
2. Define an array of country names in MainActivity.java.
3. Create an ArrayAdapter and attach it to the AutoCompleteTextView.
4. Set setThreshold(1) → Suggestions appear after 1 character.
5. Run the app → Typing shows dropdown suggestions.

Program 7: Program to create Picker view (Calendar view).

Aim: To write a program to create Picker view (Calendar view).

Procedure:
1. Add a DatePicker in activity_main.xml.
2. In MainActivity.java, initialize it with today’s date using Calendar.
3. Use OnDateChangedListener to handle date changes.
4. Show a Toast message with the selected date.
5. Run the app → Selecting a date shows it as a toast.

Program 8: Program to create List view.

Aim: To write a program to create List view.

Procedure:
1. Add a ListView in activity_main.xml.
2. In MainActivity.java, create a string array of items.
3. Use an ArrayAdapter to bind the array to the ListView.
4. Set the adapter using listView.setAdapter(adapter).
5. Run the app → Items are displayed in a scrolling list.

Program 9: Program to Display Spinner view.

Aim: To write a program to Display Spinner view.

Procedure:
1. Add a Spinner in activity_main.xml.
2. In MainActivity.java, define an array (e.g., countries).
3. Create an ArrayAdapter with the array.
4. Set the adapter to the spinner using spinner.setAdapter(adapter).
5. Run the app → Spinner shows dropdown list.

Program 10: Program to Display web page using Web View.

Aim: To write a program to Display web page using Web View.

Procedure:
1. Add a WebView in activity_main.xml.
2. In MainActivity.java, initialize the WebView.
3. Set webView.setWebViewClient(new WebViewClient()) so pages open inside the app.
4. Enable JavaScript if needed using getSettings().setJavaScriptEnabled(true).
5. Load a URL using webView.loadUrl("https://wikipedia.org").
6. Run the app → Web page displays in the app.

Program 11: Program to Display image using Image view.

Aim: To write a program to Display image using Image view.

Procedure:
1. Add an ImageView in activity_main.xml.
2. Place an image file inside the res/drawable folder.
3. Set android:src="@drawable/imagename" in the XML.
4. In MainActivity.java, load the layout with setContentView().
5. Run the app → Image is displayed.

Program 12: Program to Display Digital Clock.

Aim: To write a program to Display Digital Clock.

Procedure:
1. Add a TextClock widget in activity_main.xml.
2. Customize its textSize and alignment.
3. In MainActivity.java, load the layout.
4. Run the app → Digital clock shows current time automatically.

Program 13: Program to Create Android Services (Play Music).

Aim: To write a program to Create Android Services (Play Music).

Procedure:
1. Place an audio file in res/raw.
2. Create a Service class (MusicService.java) and use MediaPlayer to play music.
3. Override onStartCommand() to start playback, and onDestroy() to stop it.
4. Register the service in AndroidManifest.xml.
5. In MainActivity.java, add Start and Stop buttons with startService() and stopService().
6. Run the app → Buttons control music playback.

Program 14: Program to Handle Run time error using Exception Handling.

Aim: To write a program to Handle Run time error using Exception Handling.

Procedure:
1. Add an EditText, a Button, and a TextView in activity_main.xml.
2. In MainActivity.java, read user input on button click.
3. Use try-catch blocks:
- Catch NumberFormatException if input is not a number.
- Catch ArithmeticException for division by zero.
- Catch generic Exception for unexpected errors.
4. Display result or error message in the TextView.
5. Run the app → Safe handling of runtime errors.

Program 15: Program to connect SQLite to Android.

Aim: To write a program to connect SQLite to Android.

Procedure:
1. Create a helper class extending SQLiteOpenHelper (e.g., Database1).
2. In onCreate(), define the table using SQL CREATE TABLE.
3. Create a Student class with methods to add, update, delete, and fetch data.
4. In activity_main.xml, add input fields and buttons.
5. In MainActivity.java, call methods to perform database operations.
6. Run the app → Data is stored, updated, and deleted from SQLite database.