Step-by-step to create and run a **complete Android app in Kotlin**:

- **Activity Lifecycle**
- **Layouts & UI Design**
- **Handling User Input**

Build a simple app that:

1. Logs the **Activity Lifecycle** events (creation, start, pause, resume, etc.).
2. Has a **designed layout** using XML (with TextView, EditText, Button).
3. Handles **user input** (calculates and displays results).

STEP 1: Create a New Android Project

1. Open **Android Studio** → Click **"New Project"**
2. Choose **Empty Activity** → click **Next**
3. Give details:
    - Name: LifeCycleDemo
    - Package: com.example.lifecycledemo
    - Language: **Kotlin**
    - Minimum SDK: **API 21 (Android 5.0 Lollipop)**
4. Click **Finish**

STEP 2: Understand the File Structure

After creation, the important folders are:

```
app/
└── src/
    └── main/
        ├── java/com/example/lifecycledemo/MainActivity.kt
        ├── res/
        │   ├── layout/activity_main.xml
        │   ├── values/colors.xml
        │   ├── values/strings.xml
        │   ├── mipmap/ (app icons)
        └── AndroidManifest.xml
```

Note:
If you **don't see the layout folder**, click on the dropdown (above the file tree) and change **"Android" view → "Project"** or **"Project Files"** view.

Then go to:
app/src/main/res/layout/activity_main.xml
If it doesn't exist, you can **right-click layout → New → Layout Resource File → activity_main.xml.**

STEP 3: Design the Layout (activity_main.xml)
Here's a layout that uses **ConstraintLayout**, **TextView**, **EditText**, and **Button**.

📁 **activity_main.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.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">

  <TextView
    android:id="@+id/tvTitle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Simple Calculator"
    android:textSize="24sp"
    android:textStyle="bold"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    android:layout_marginTop="40dp"/>

  <EditText
    android:id="@+id/etNum1"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:hint="Enter number 1"
    android:inputType="number"
    app:layout_constraintTop_toBottomOf="@id/tvTitle"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    android:layout_margin="20dp"/>

  <EditText
    android:id="@+id/etNum2"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:hint="Enter number 2"
    android:inputType="number"
    app:layout_constraintTop_toBottomOf="@id/etNum1"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    android:layout_margin="20dp"/>
```

```
    <Button
        android:id="@+id/btnAdd"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Add Numbers"
        app:layout_constraintTop_toBottomOf="@id/etNum2"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"/>

    <TextView
        android:id="@+id/tvResult"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Result will appear here"
        android:textSize="18sp"
        app:layout_constraintTop_toBottomOf="@id/btnAdd"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        android:layout_marginTop="20dp"/>
</androidx.constraintlayout.widget.ConstraintLayout>
```

STEP 4: Code the Activity (MainActivity.kt)

📁 **MainActivity.kt**

```
package com.example.lifecycledemo

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.util.Log
import android.widget.Button
import android.widget.EditText
import android.widget.TextView
import android.widget.Toast

class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
        Log.d("LifeCycleDemo", "onCreate called")

        val num1 = findViewById<EditText>(R.id.etNum1)
        val num2 = findViewById<EditText>(R.id.etNum2)
```

```
        val btnAdd = findViewById<Button>(R.id.btnAdd)
        val result = findViewById<TextView>(R.id.tvResult)

        btnAdd.setOnClickListener {
            val n1 = num1.text.toString().toDoubleOrNull()
            val n2 = num2.text.toString().toDoubleOrNull()
            if (n1 != null && n2 != null) {
                val sum = n1 + n2
                result.text = "Result: $sum"
                Toast.makeText(this, "Addition Successful!", Toast.LENGTH_SHORT).show()
            } else {
                Toast.makeText(this, "Enter valid numbers!", Toast.LENGTH_SHORT).show()
            }
        }
    }

    override fun onStart() {
        super.onStart()
        Log.d("LifeCycleDemo", "onStart called")
    }

    override fun onResume() {
        super.onResume()
        Log.d("LifeCycleDemo", "onResume called")
    }

    override fun onPause() {
        super.onPause()
        Log.d("LifeCycleDemo", "onPause called")
    }

    override fun onStop() {
        super.onStop()
        Log.d("LifeCycleDemo", "onStop called")
    }

    override fun onDestroy() {
        super.onDestroy()
        Log.d("LifeCycleDemo", "onDestroy called")
    }
}
```

STEP 5: Run and Observe

1. Click **Run** ▶
2. Open **Logcat** (bottom panel in Android Studio)
3. Search for LifeCycleDemo
4. You'll see logs like:
5. D/LifeCycleDemo: onCreate called
6. D/LifeCycleDemo: onStart called
7. D/LifeCycleDemo: onResume called

Try minimizing or closing the app — watch the lifecycle logs change.

STEP 6: Observe the following:

Activity Lifecycle:
Each onCreate(), onStart(), onResume()... is a **callback** showing how the Activity behaves as the user interacts with it.

Layouts & UI Design:
You built a static layout using XML — used **ConstraintLayout**, **TextView**, **EditText**, and **Button**.

Handling User Input:
You handled **EditText** input, processed values on **Button click**, and showed results via **TextView** and **Toast**.