

Tutorial 1

Mobile Application Development(COMP2008)

Discipline of Computing School of Electrical
Engineering, Computing and Mathematical
Sciences (EECMS)

Copyright © 2018, Curtin University CRICOS
Provide Code: 00301J

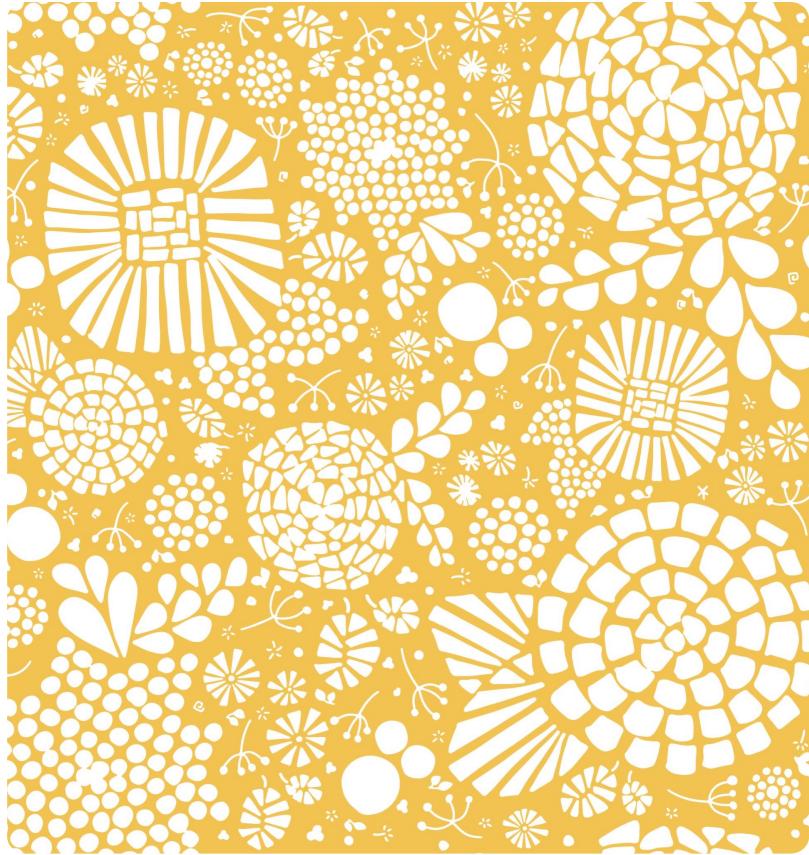


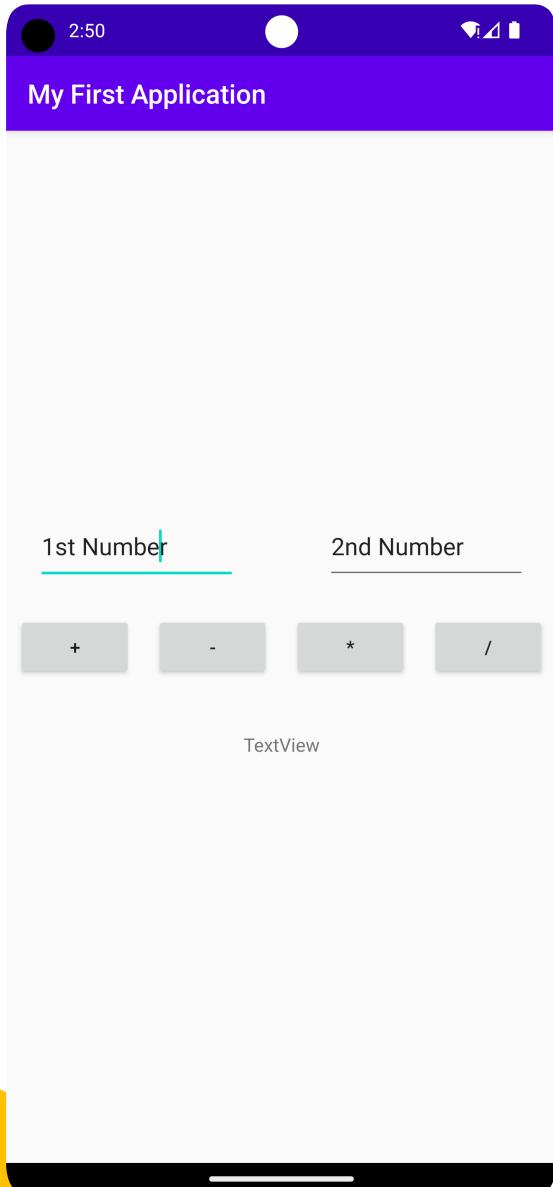
Curtin University

Objectives

- **Build a calculator**

Due date information is on the last page.





- To give you a gentle introduction to app design, we're going to make a four-function calculator that will end up looking vaguely like this:

Reading: Lecture 1

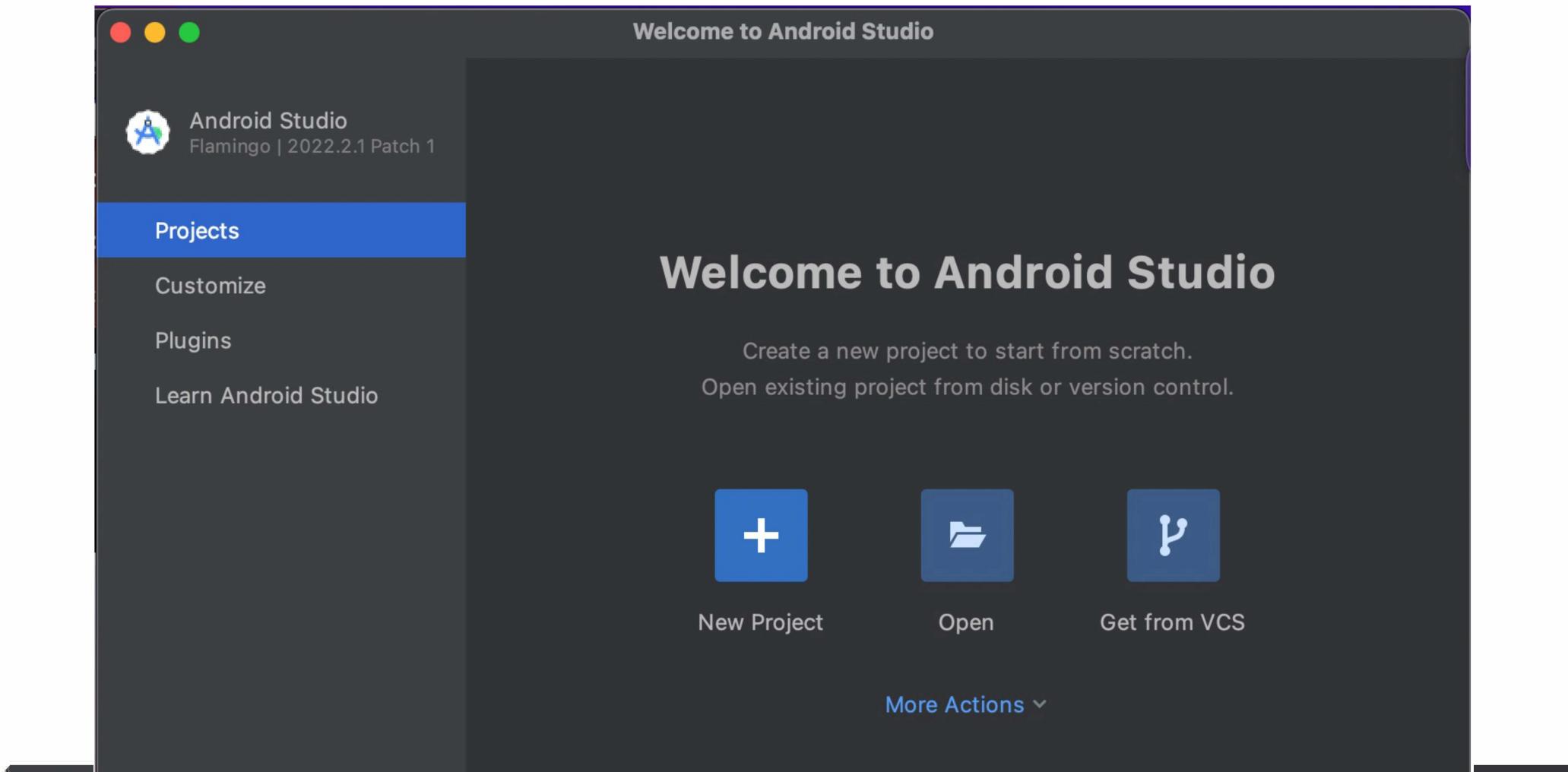
Source code: `calculator.zip` (attached in the lecture 1 module of the BB). It adds two number in a calculator.

You can directly build by editing the given code.

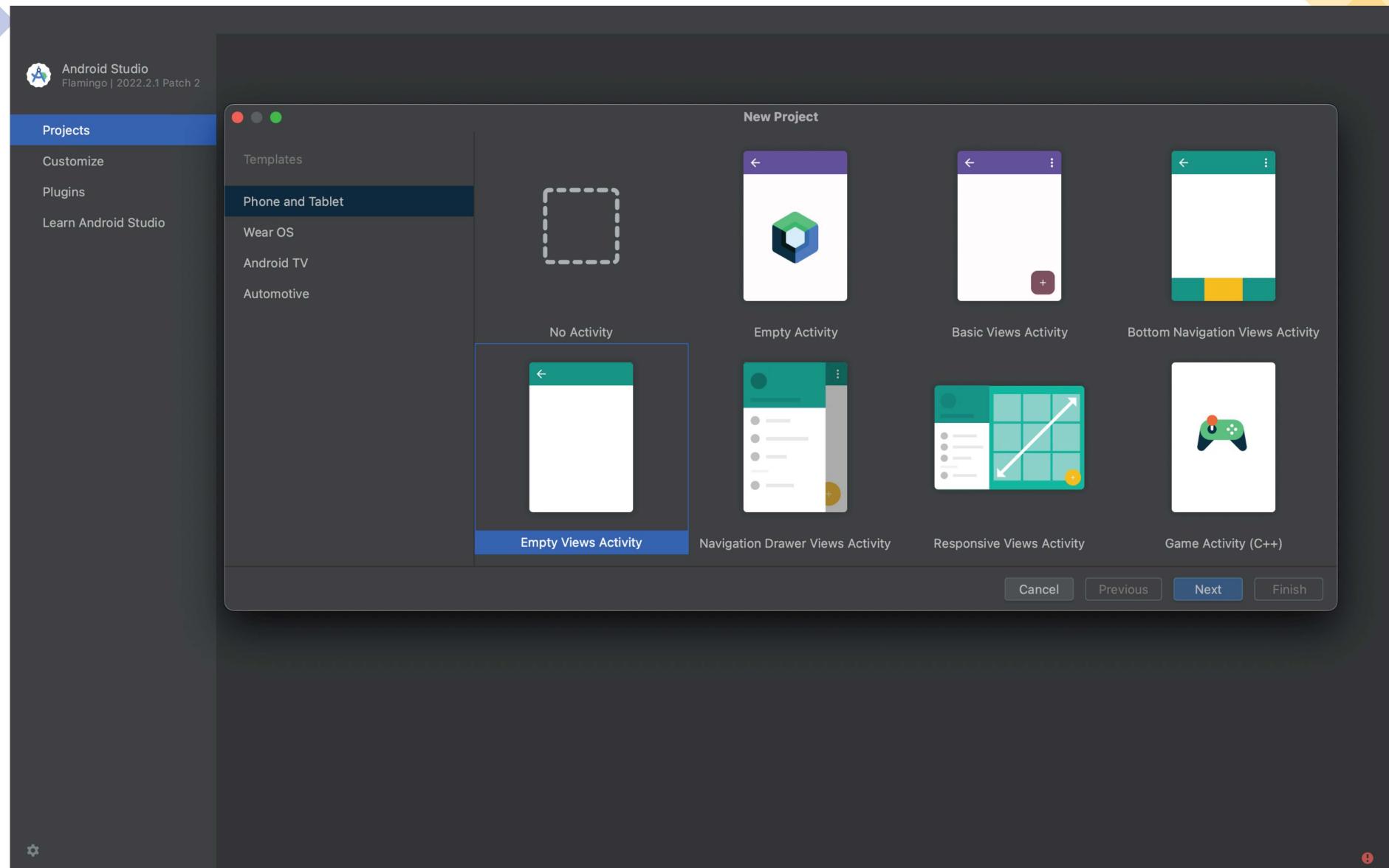
You are also allowed to build app from scratch.

The next slides reiterate some of the Lecture 1 slides. If you understand how to build simple android app, you can directly go to slide 18, which specifies the task/modification required. **We are going to build a calculator that adds two numbers first.** Your task is to implement multiplication, subtraction and division.

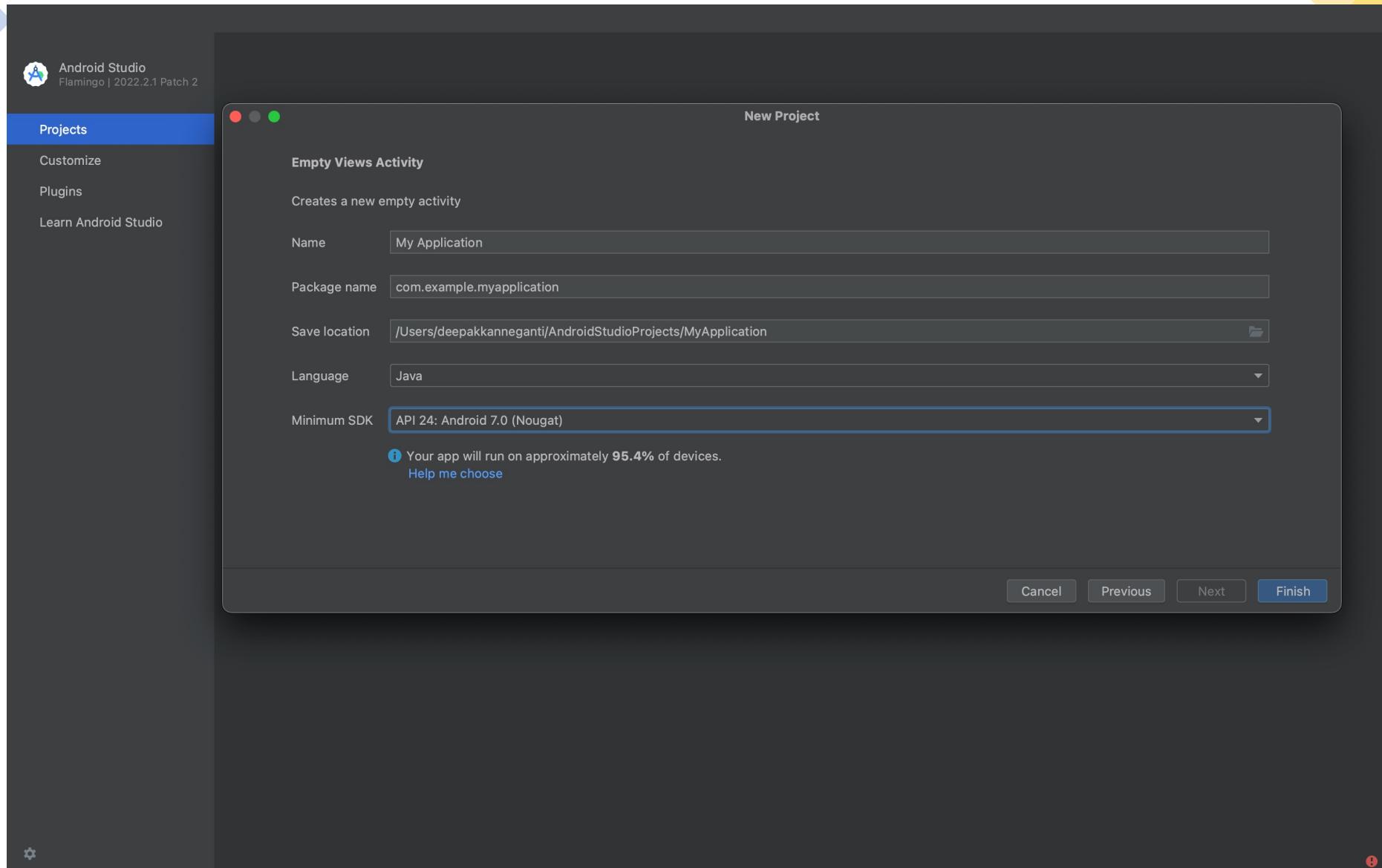
STEP 1: Creating a New Project



STEP 2: Select Empty Views Activity



Step 3: Give a name of your application. Select your language, java or kotlin, and press finish



This is an empty project look like.

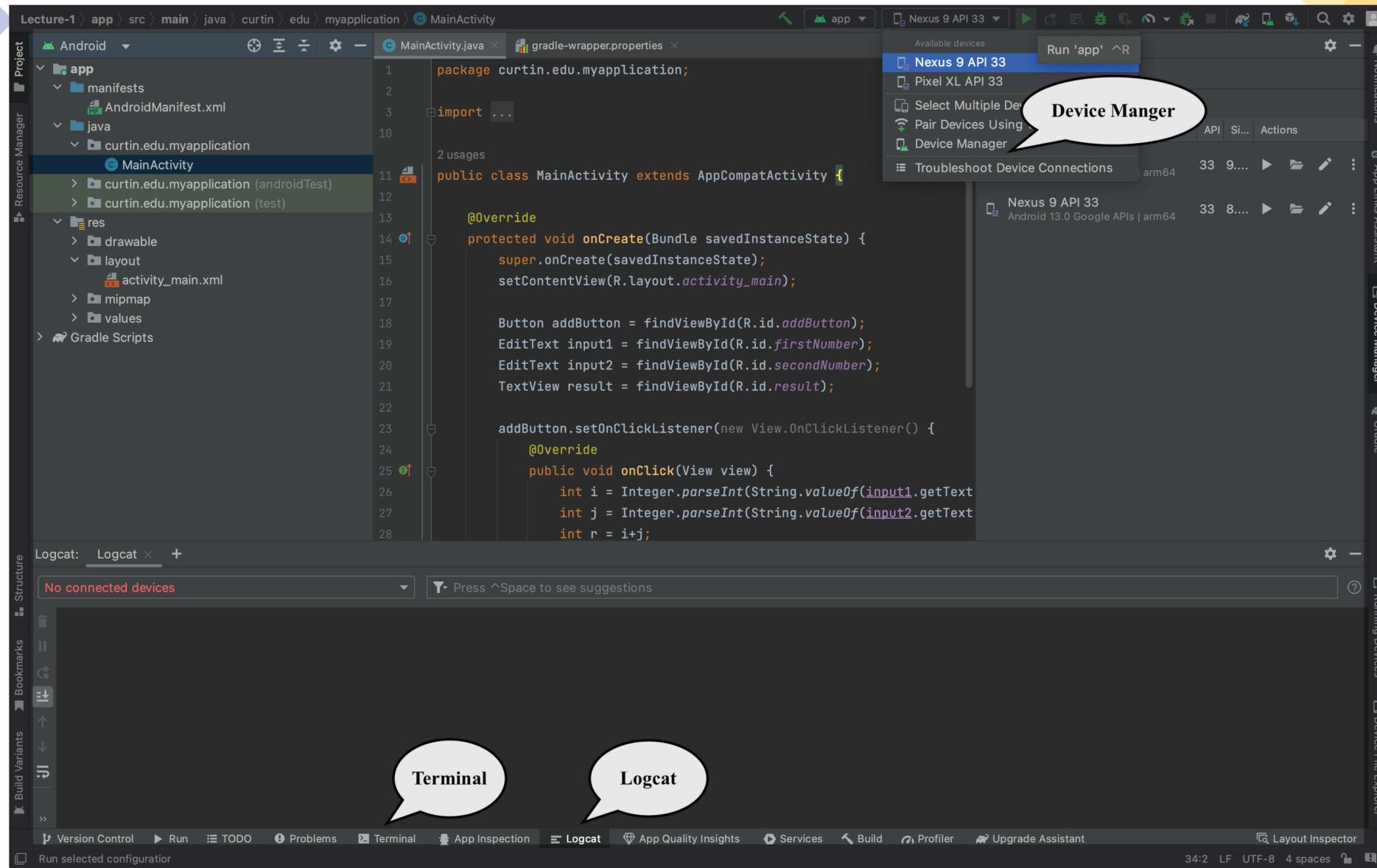
The screenshot shows the Android Studio interface with the following details:

- Project Bar:** Shows "MyApplication > app > src > main > res > layout > activity_main.xml".
- Toolbar:** Includes icons for Run, Stop, Build, and Device Manager.
- ActionBar:** Displays "Pixel 6 Pro API 33" and other navigation icons.
- Code Editor:** The file "activity_main.xml" is selected in the Project Bar. The code in "MainActivity.java" is as follows:

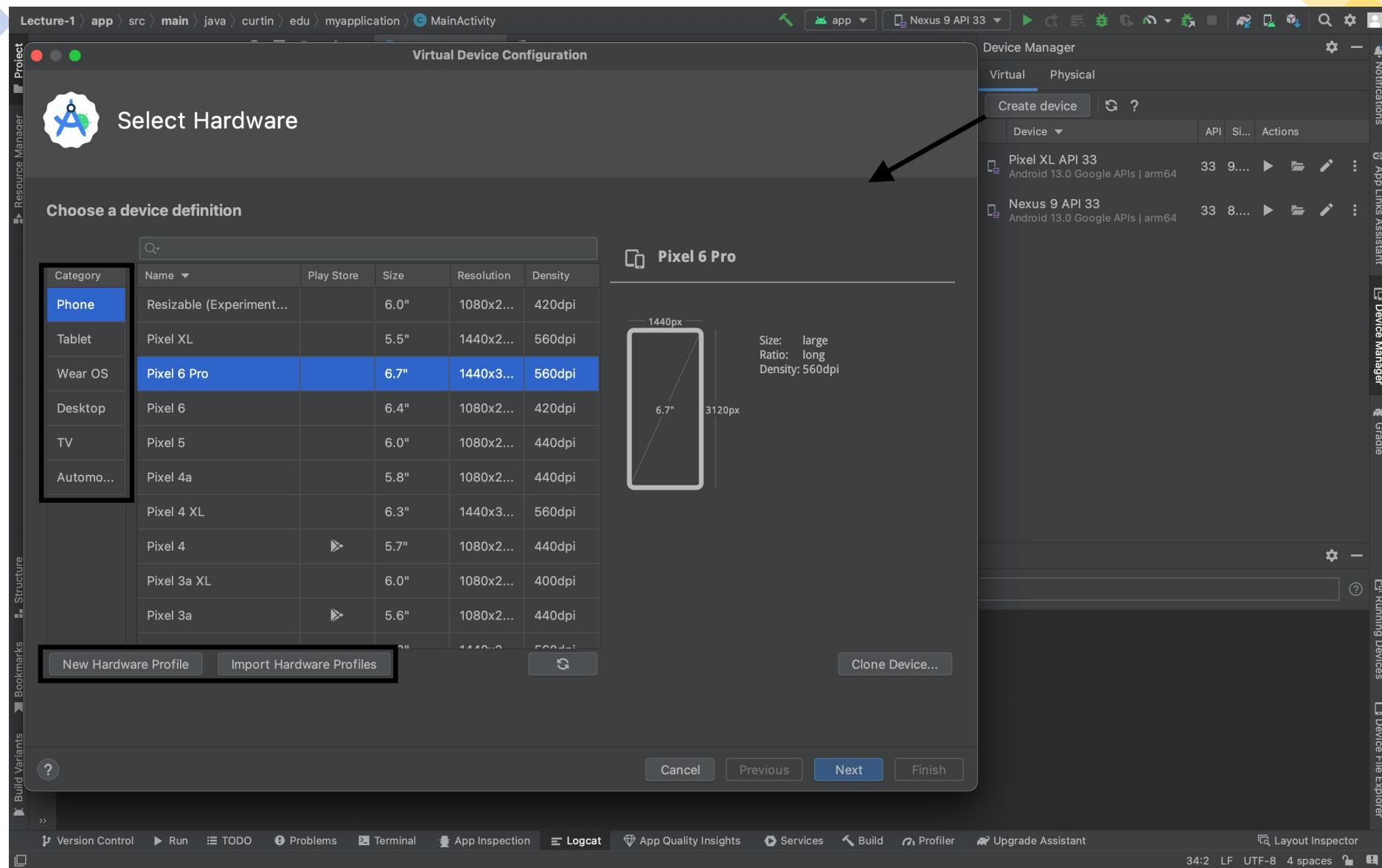
```
1 package com.example.myapplication;
2
3 import ...
4
5 public class MainActivity extends AppCompatActivity {
6
7     @Override
8     protected void onCreate(Bundle savedInstanceState) {
9         super.onCreate(savedInstanceState);
10        setContentView(R.layout.activity_main);
11    }
12}
```

- Project Structure:** The "activity_main.xml" file is highlighted in the Project Bar under the "layout" folder.
- Side Panels:** Notifications, Device Manager, and Gradle panels are visible on the right side.
- Bottom Navigation:** Includes tabs for Version Control, TODO, Problems, Terminal, App Inspection, Logcat, App Quality Insights, Services, Build, Profiler, and Layout Inspector.
- Status Bar:** Shows "Gradle sync finished in 4 s 794 ms (10 minutes ago)" and "1:1 LF UTF-8 4 spaces".

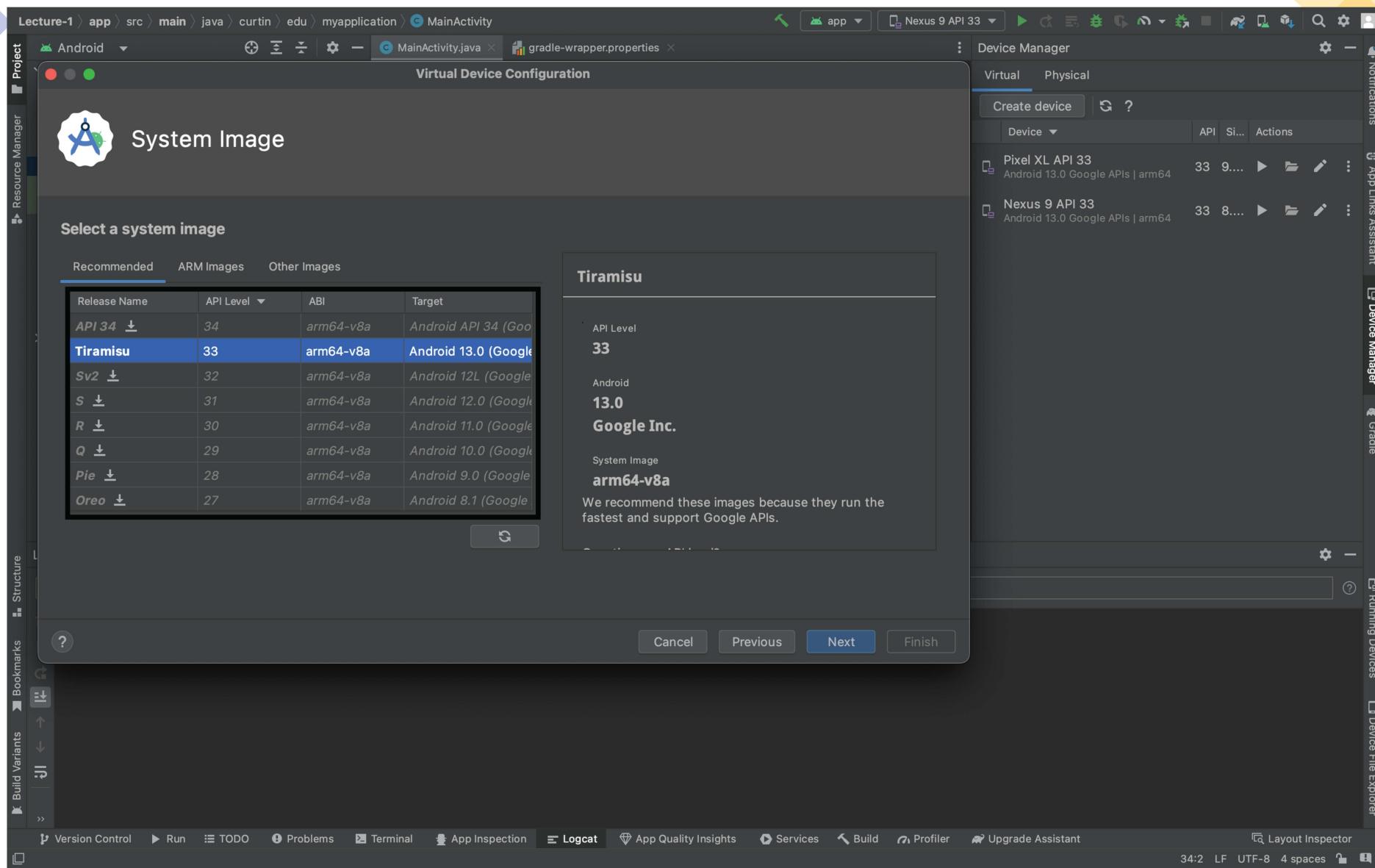
Make sure you have added a device in the device manager. Try to install a phone and a tablet.



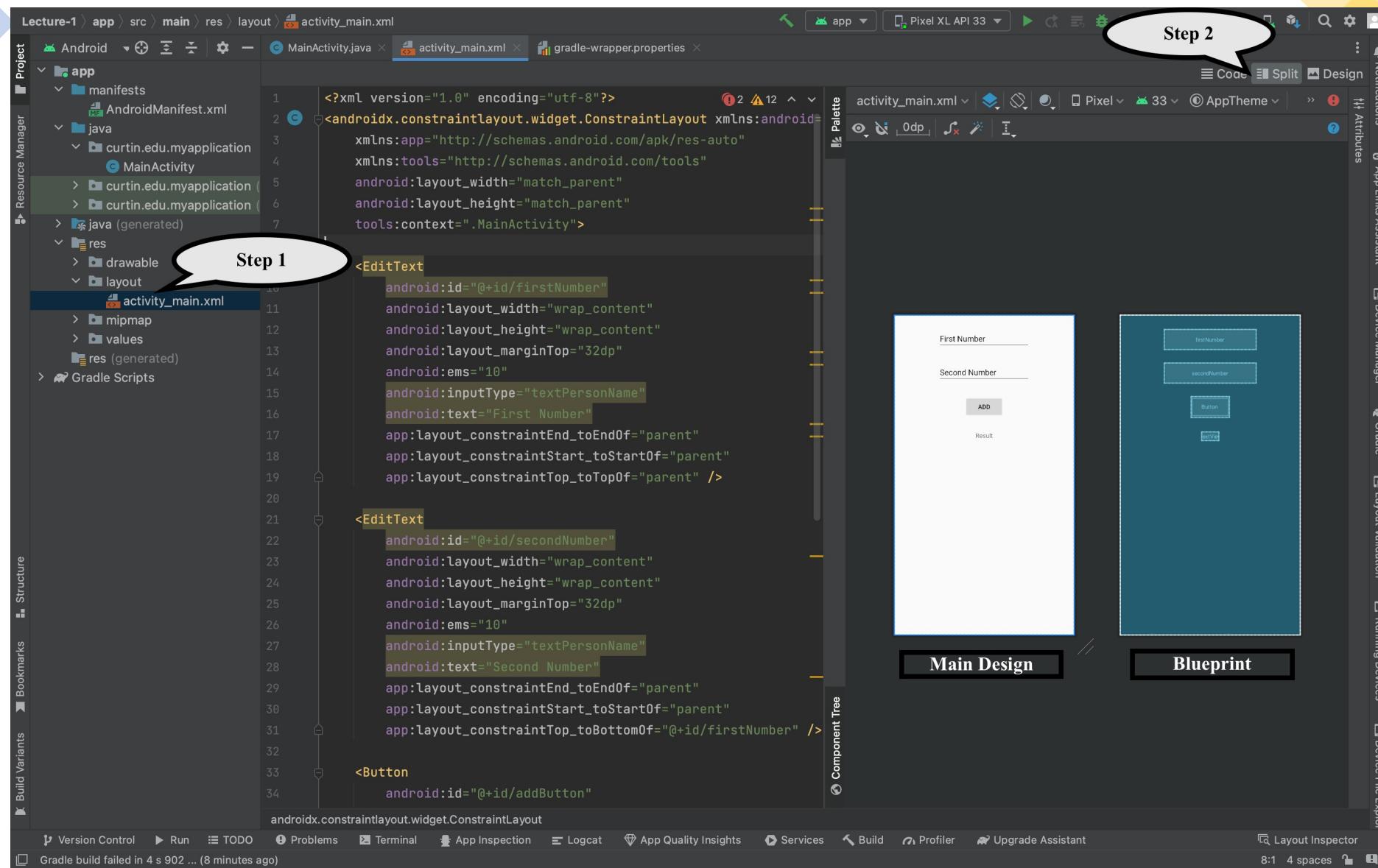
Selecting your device where the code will be run



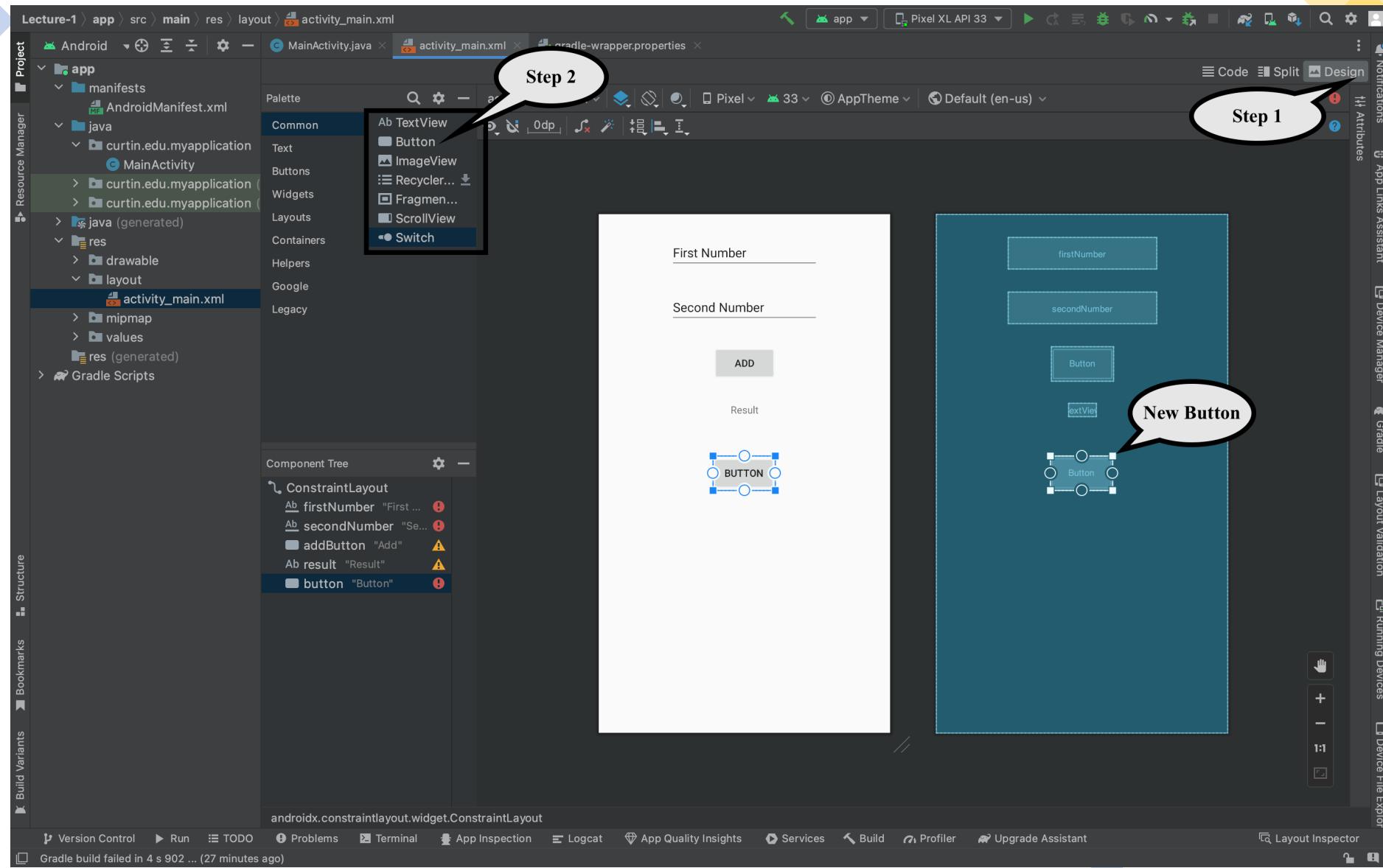
You may need to select a system image



The activity_main.xml has the view codes. Look at the Main Design and Blueprint



Drag and Drop two EditTexts, on Button and a TextView



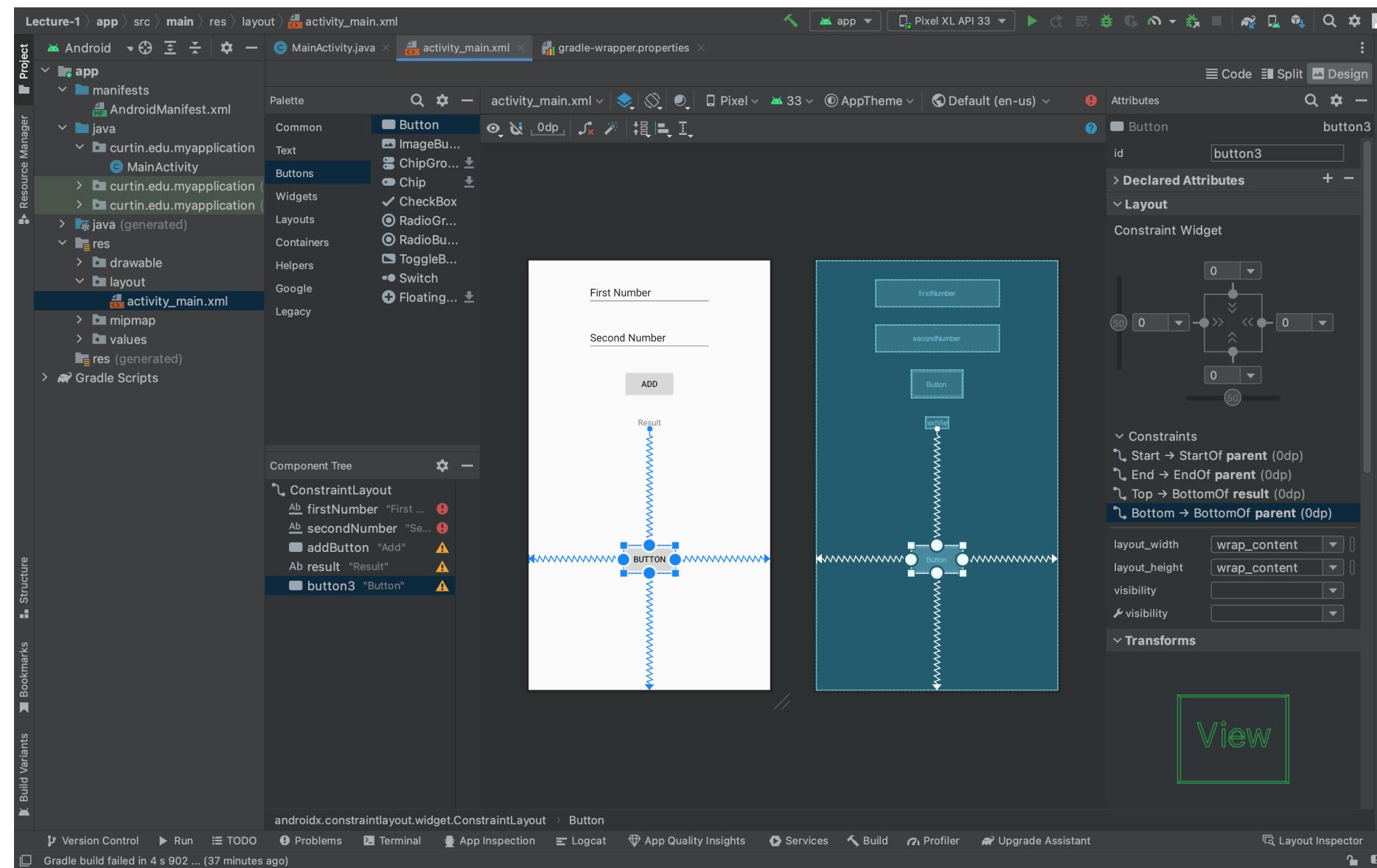
change the android:text, android:id and give your desired values. Id is for java code and text is what displays on the screen

The screenshot shows the Android Studio interface with the project 'Lecture-1' open. The 'activity_main.xml' layout file is selected in the Project structure. The layout XML code is displayed in the main editor:

```
<Button  
    android:id="@+id/addButton"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="32dp"  
    android:text="Add"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toBottomOf="@+id/secondNumber" />  
  
<TextView  
    android:id="@+id/result"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="32dp"  
    android:text="Result"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintStart_toStartOf="parent"  
    app:layout_constraintTop_toBottomOf="@+id/addButton" />  
  
<Button  
    android:id="@+id/button"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Button"  
    tools:layout_editor_absoluteX="161dp"  
    tools:layout_editor_absoluteY="341dp" />  
  
</androidx.constraintlayout.widget.ConstraintLayout>
```

The bottom right corner of the code editor has a red border, indicating a warning or error. To the right of the code editor is the 'Component Tree' panel, which shows the structure of the layout components. Below the code editor is the 'AndroidManifest.xml' file. The bottom of the screen shows the standard Android Studio navigation bar.

Add Constraints by dragging the left, right, top and bottom of the elements. It fits them in relation to the other views. For example, the button is constrained according to the parent at start, end, bottom and to result at top.



Now, in the MainActivity.Java, first create four variable for button, two inputs and the textView. Set a click listener for the button. Override the onClick method as follows: get the texts from two editTexts and then convert them to integer, add them, and finally set the text of the textView to show it on the screen

The screenshot shows the Android Studio interface with the project 'Lecture-1' open. The left sidebar displays the project structure, including the app module with its manifest, Java files, and resources. The main editor window shows the MainActivity.java code. The code defines a public class MainActivity that extends AppCompatActivity. It overrides the onCreate method to set the content view to activity_main. Inside the onCreate method, it finds views for an addButton, input1, input2, and result. It then sets an onClickListener for the addButton. The onClickListener's implementation retrieves the text from input1 and input2, converts them to integers, adds them together, and sets the result in the result textView. The code ends with a closing brace for the onCreate method and another for the class definition.

```
package curtin.edu.myapplication;
import ...;

public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Button addButton = findViewById(R.id.addButton);
        EditText input1 = findViewById(R.id.firstNumber);
        EditText input2 = findViewById(R.id.secondNumber);
        TextView result = findViewById(R.id.result);

        addButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                int i = Integer.parseInt(String.valueOf(input1.getText()));
                int j = Integer.parseInt(String.valueOf(input2.getText()));
                int r = i+j;
                result.setText(String.valueOf(r));
            }
        });
    }
}
```

The status bar at the bottom indicates a Gradle build failure: "Gradle build failed in 4 s 902 ... (40 minutes ago)". The bottom right corner shows the current time as 16:46 and file details as 13 chars, LF, UTF-8, 4 spaces.

Lecture-13 > app > src > main > java > curtin > edu > myapplication > MainActivity > onCreate

Project Structure Manager Resource Manager External Libraries Scratches and Consoles

MainActivity.java gradle-wrapper.properties build.gradle (:app)

Running Devices: Pixel XL API 33

Device Manager Notifications Gradle

Result

First Number
Second Number
ADD

6:35 Lecture-1

Run: app

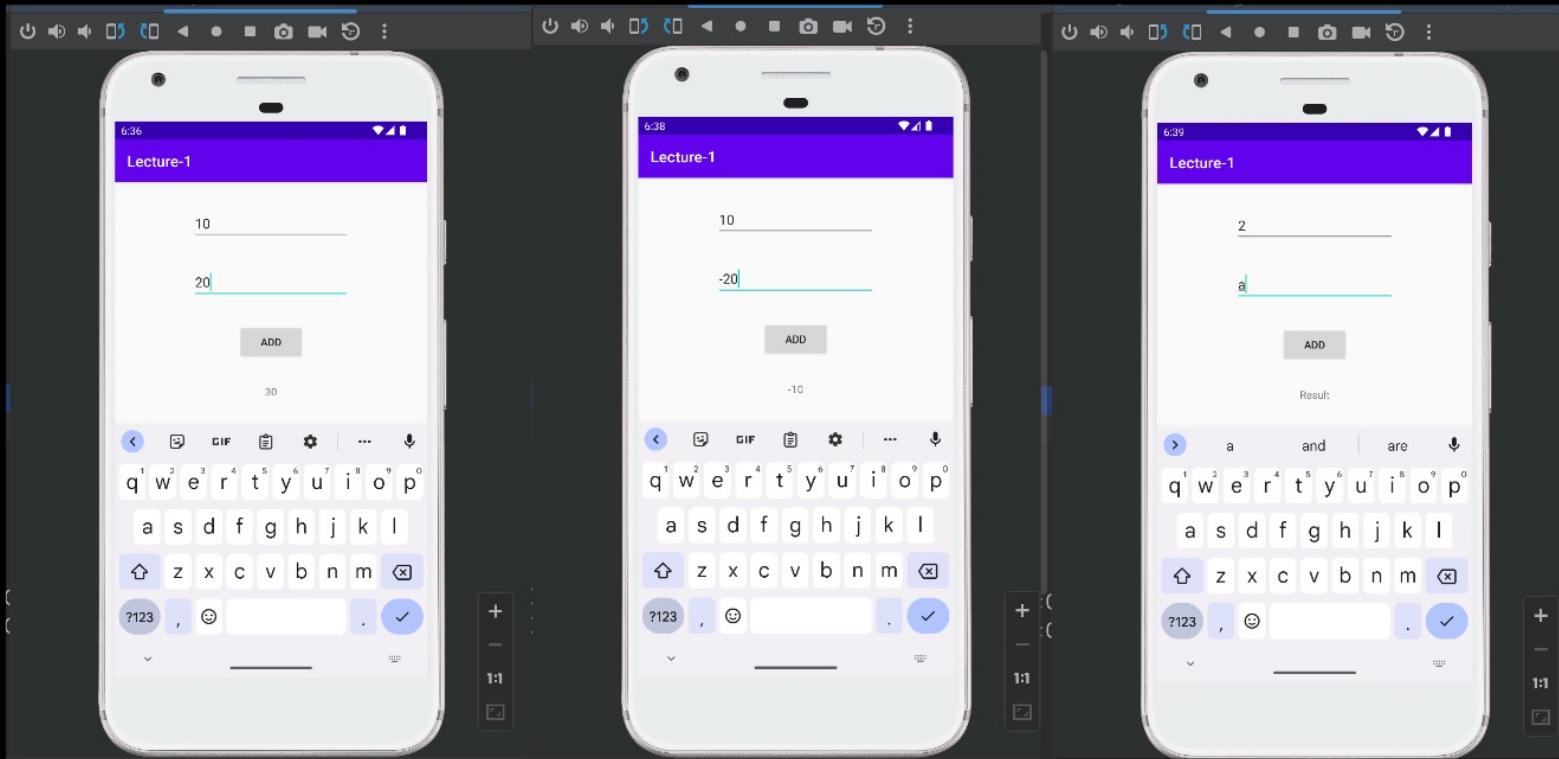
Bookmarks Build Variants

07/15 18:33:51: Launching 'app' on Pixel XL API 33.
Install successfully finished in 1 s 293 ms.
\$ adb shell am start -n "curtin.edu.myapplication/curtin.edu.myapplication.MainActivity" -a android.intent.action.MAIN -c android.intent.category.LAUNCHER
--splashscreen-show-icon
[Open logcat panel for emulator Pixel XL API 33](#)
Connected to process 6167 on device 'Pixel_XL_API_33 [emulator-5554]'.

Version Control Run TODO Problems Terminal Logcat App Quality Insights Services Profiler App Inspection Upgrade Assistant Layout Inspector

Launch succeeded (2 minutes ago) 20:1 (58 chars, 1 line break) LF UTF-8 4 spaces

The screenshot shows the Android Studio interface with the project 'Lecture-13 [My Application]' open. The code editor displays the MainActivity.java file, which contains Java code for handling the onCreate method of an AppCompatActivity. The code includes findViewById calls for an add button and two edit texts, and sets an onClickListener for the add button to perform addition and update a result TextView. The run tab shows successful launch logs for the Pixel XL API 33 emulator. The right side of the screen shows the emulator displaying a simple UI with fields for First Number and Second Number, an ADD button, and a Result field.



Expected Output [10 Marks]

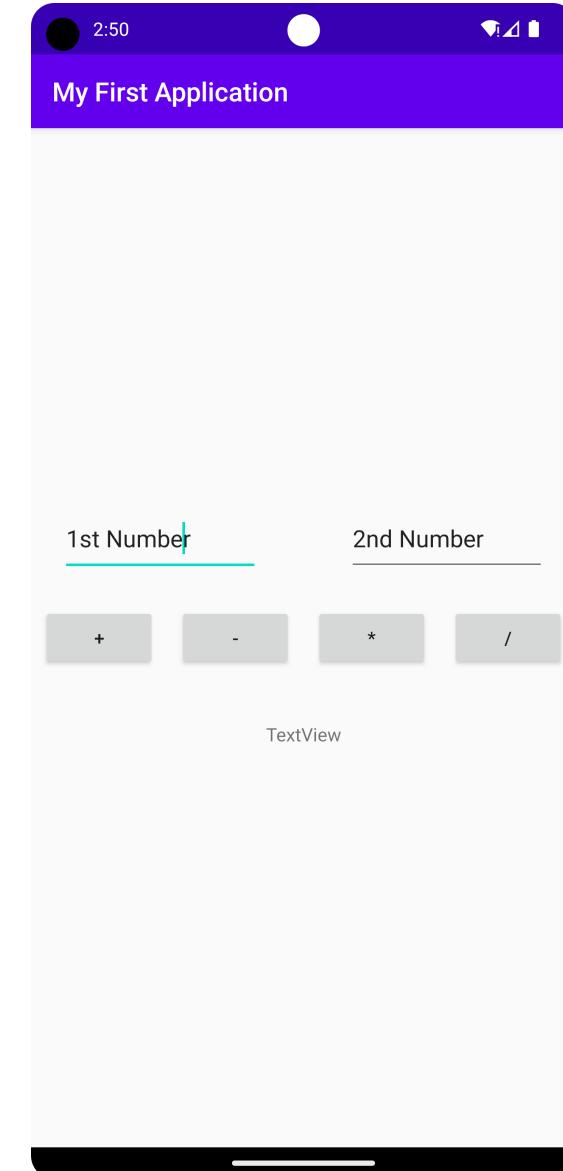
1. The calculator implements all four function for integer, double or mixed inputs

*hint: Play with Double.parseDouble

2. The calculator handles common exceptions and shows relevant information in the result (basically it does not crash)

* Think divide by zero issue

Marking guidelines: Three levels, i.e., full marks for full functional implementation, partial marks for partial functional implementation, and zero marks for no submission. You will also get partial marks if there is an effort to solve a particular functionality that is not executable.



Do not worry about how the app looks (the alignment of the buttons, input etc.). The focus is only the four calculator function

Tutorial 1-Due date August 27 Sunday 23:59 AWST ,10 Marks

It is recommended to submit early and get marked. Do not wait for the due date. Tutorial 1,2,3,4 have the same due date, i.e., 27 August Sunday 23:59 AWST

