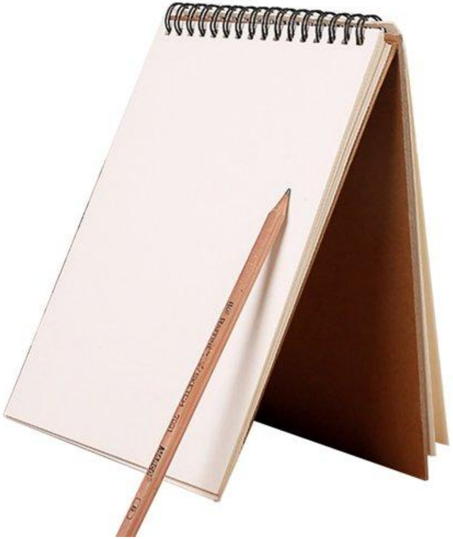
Basics of Android

University of Alberta CMPUT-301 (Winter 2023)

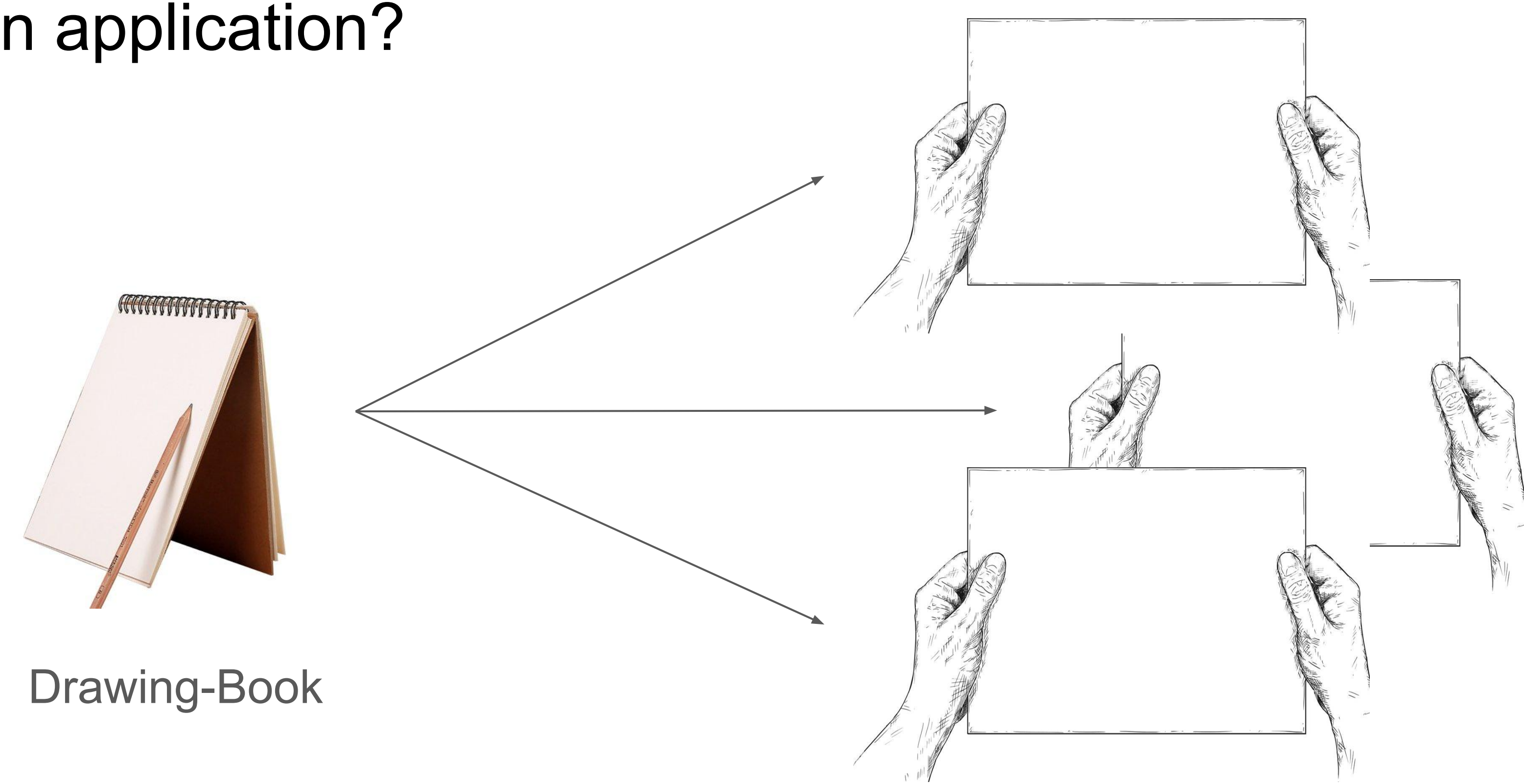
Ranit Ganguly (ranit@ualberta.ca)

What is an application?



Let’s think of an application as a Drawing-Book

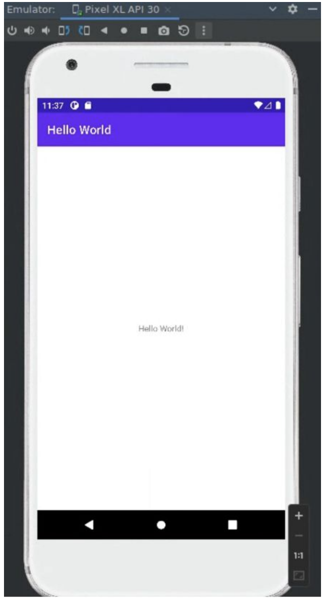
What is an application?



Drawing-Book

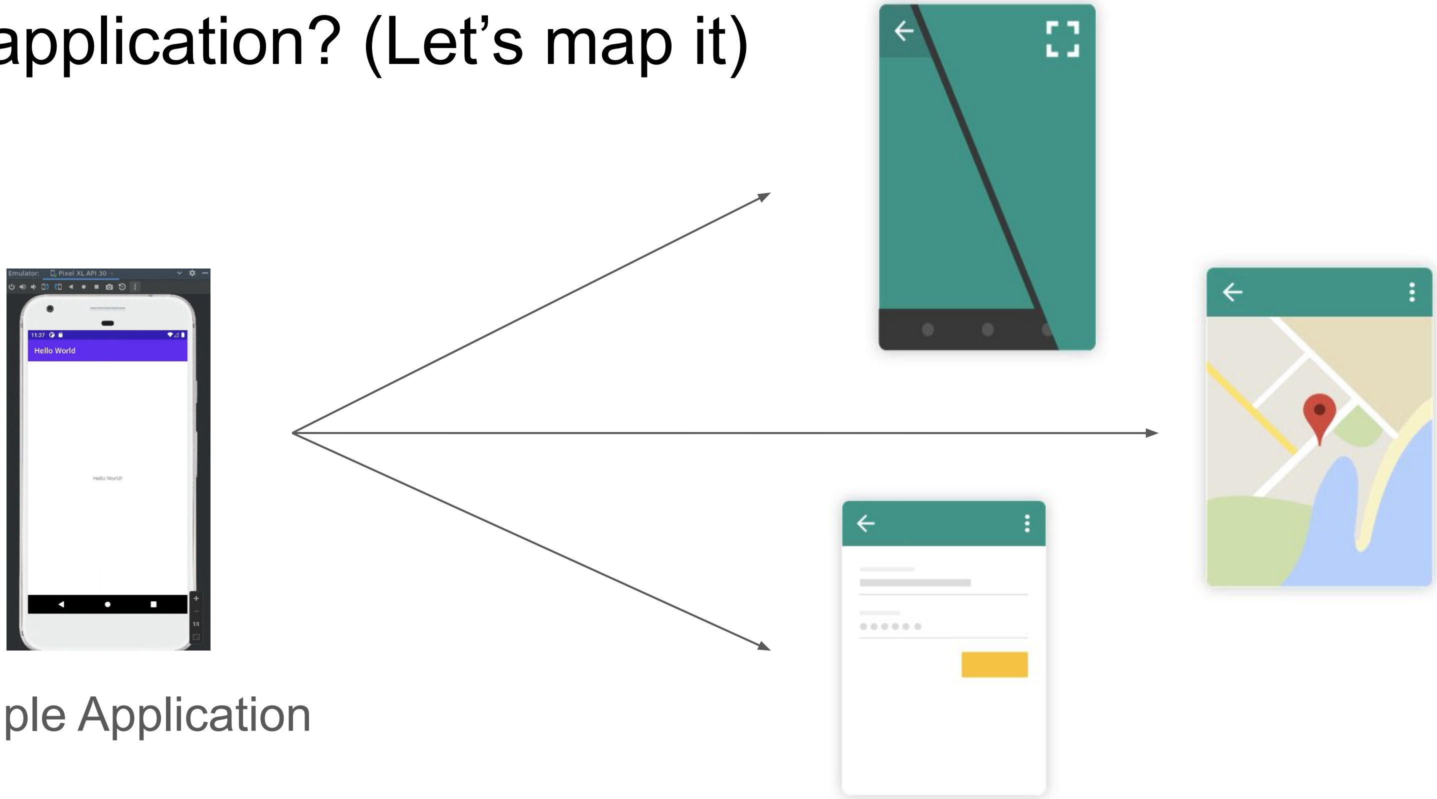
Collection of **PAGES**

What is an application? (Let’s map it)



Sample Android Application

What is an application? (Let’s map it)



Sample Application

Collection of several components
eg: **ACTIVITIES**

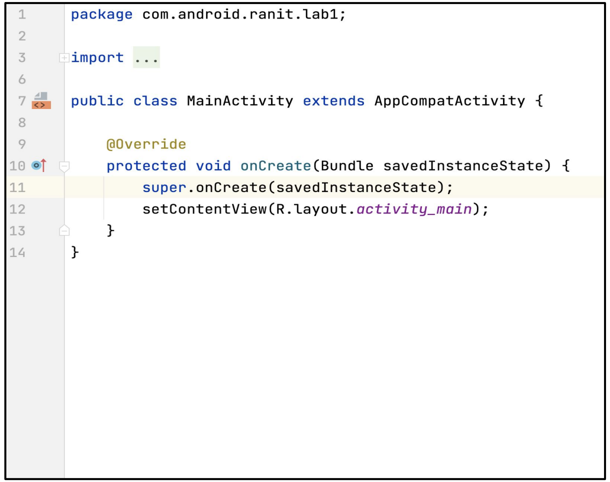
I. Activity

A **lifecycle-aware** component which provides the **window** in which the app draws the UI.

I. Activity

TWO components

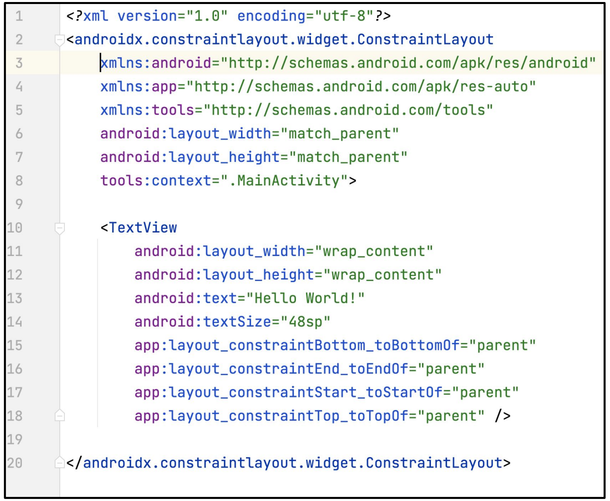
Java/ Kotlin file



```java
package com.android.ranit.lab1;

import ...

public class MainActivity extends AppCompatActivity {

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

- *MainActivity.java*

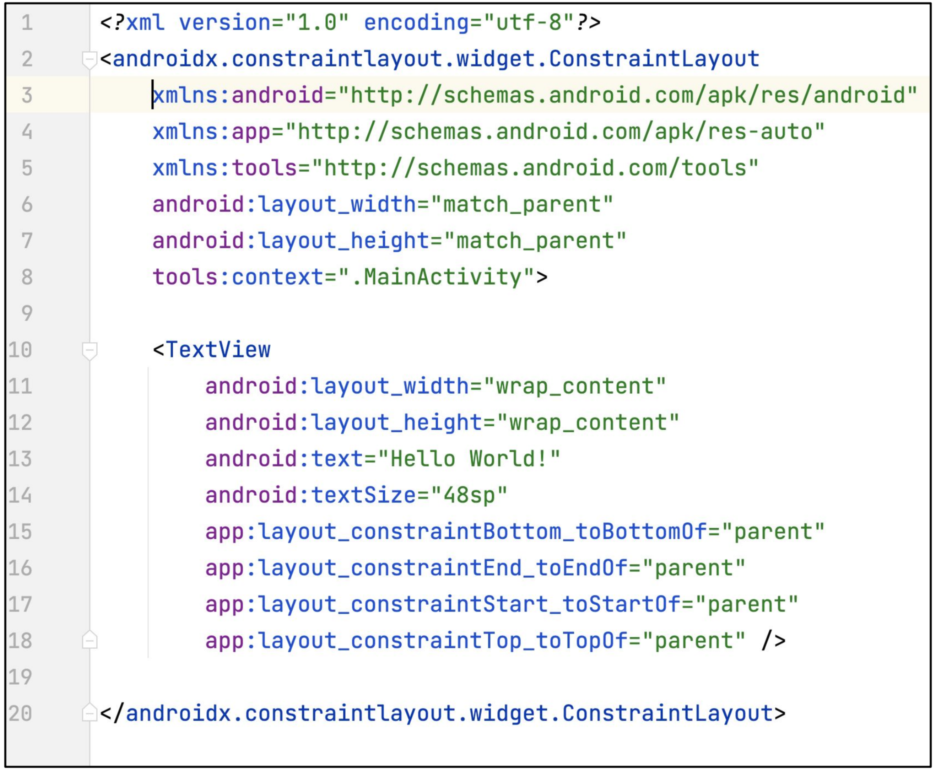
Layout file



```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:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        android:textSize="48sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

- *activity_main.xml*

I. Activity (Components)



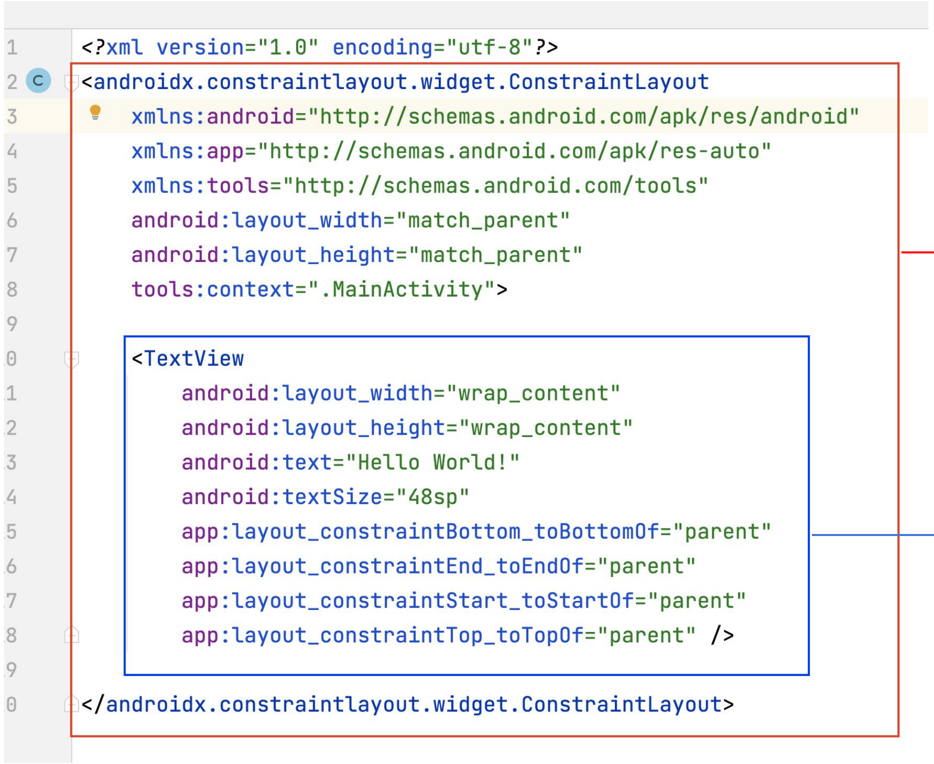
```xml
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        android:textSize="48sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

a. Layout file

[activity_main.xml]

Use **XML** to define **UI widgets** such as, TextView, EditText, Buttons etc. on the screen

I. Activity (Components)



```
<?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:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        android:textSize="48sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

a. Layout file

[activity_main.xml]

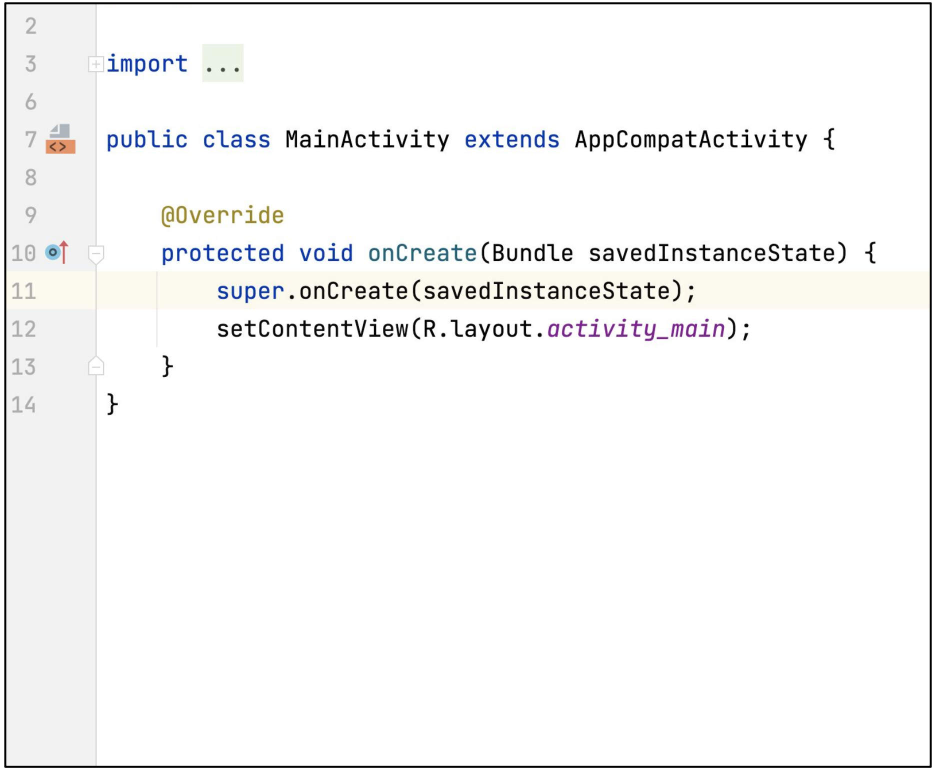
View-Groups

- Invisible containers holding views & viewGroups.
- Eg: Constraint, Linear, Frame

Views

- Simple rectangular boxes which responds to user's actions.
- Eg: Button, TextView etc.

I. Activity (Components)



```

import ...

public class MainActivity extends AppCompatActivity {

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

b. Java Class file

[MainActivity.java]

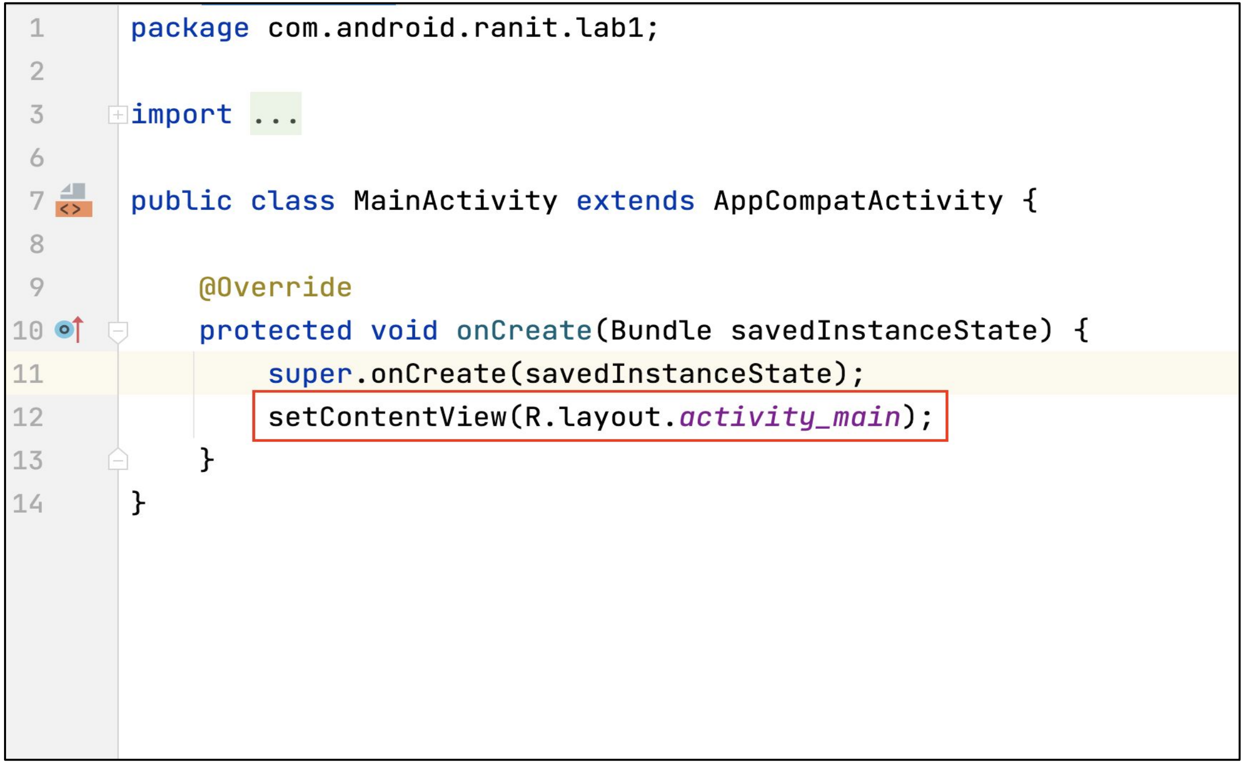
Define the **behaviour** and **logic** for the screen. *(eg: do some operation when a button is clicked)*

I. Activity (Components)

Question 1. How do we associate the **layout file** (activity_main.xml) with the Java/ Kotlin class file (MainActivity.java)?

I. Activity (Components)

Answer:

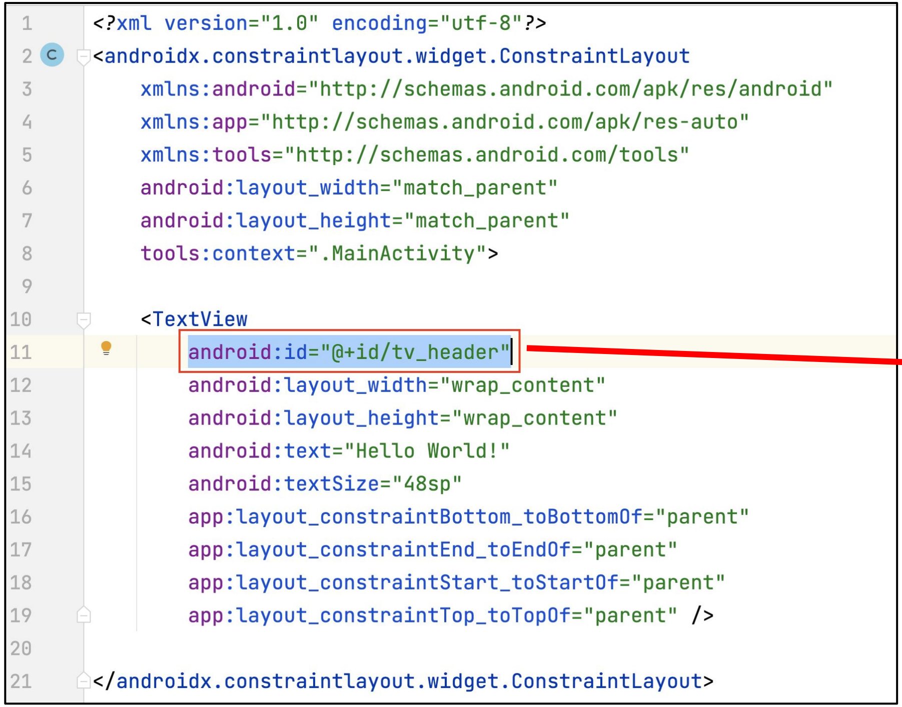


```java
package com.android.ranit.lab1;

import ...

public class MainActivity extends AppCompatActivity {

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

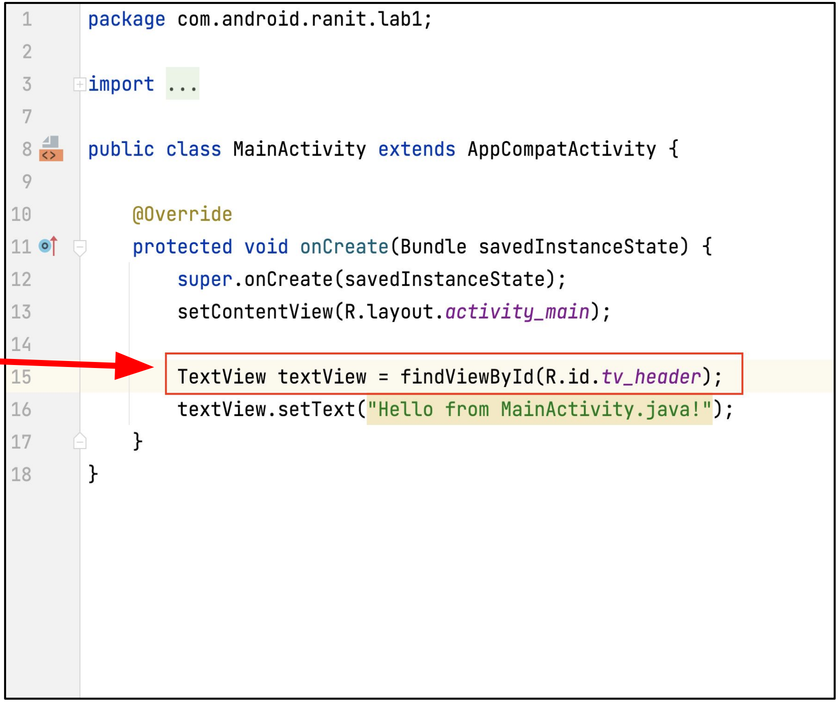
I. Activity (Components)

Question 2. How do we associate **a view** in the layout file (activity_main.xml) with the Java/ Kotlin class (MainActivity.java)?

I. Activity (Components)



```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/tv_header"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        android:textSize="48sp"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```



```java
package com.android.ranit.lab1;

import ...

public class MainActivity extends AppCompatActivity {

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

        TextView textView = findViewById(R.id.tv_header);
        textView.setText("Hello from MainActivity.java!");
    }
}
```

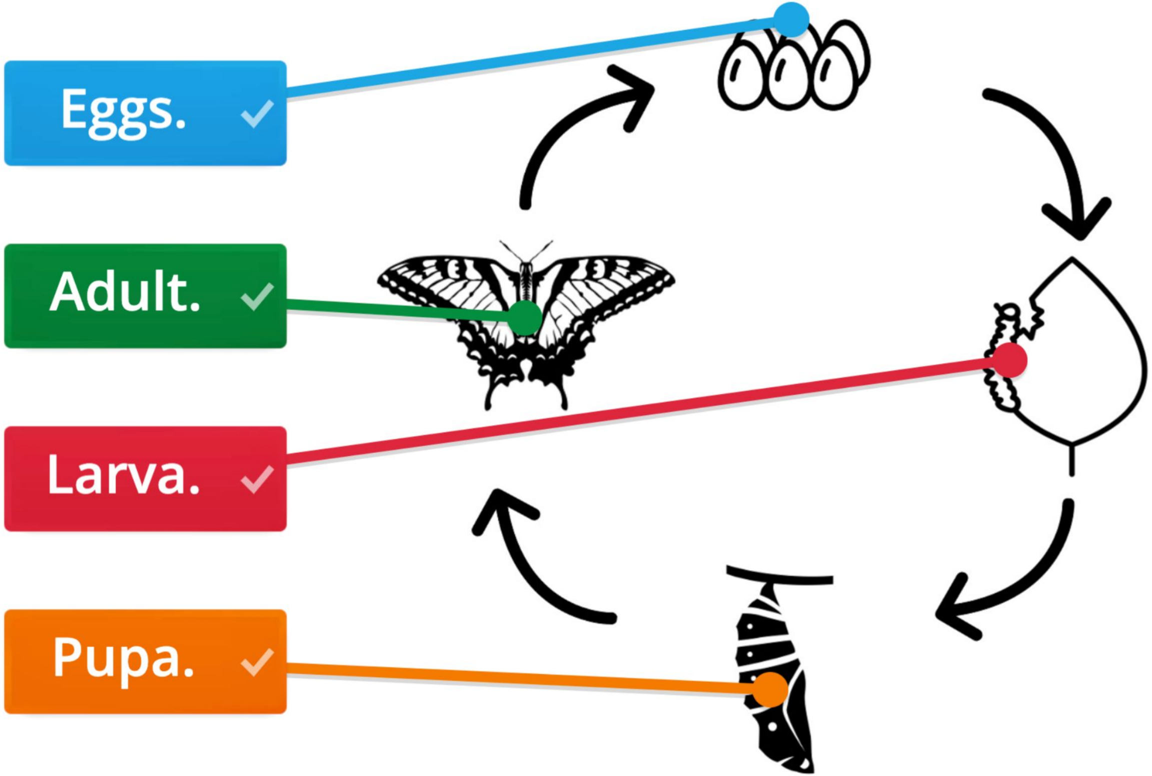
II. Activity Lifecycle

A **lifecycle-aware** component which provides the **window** in which the app draws the UI.

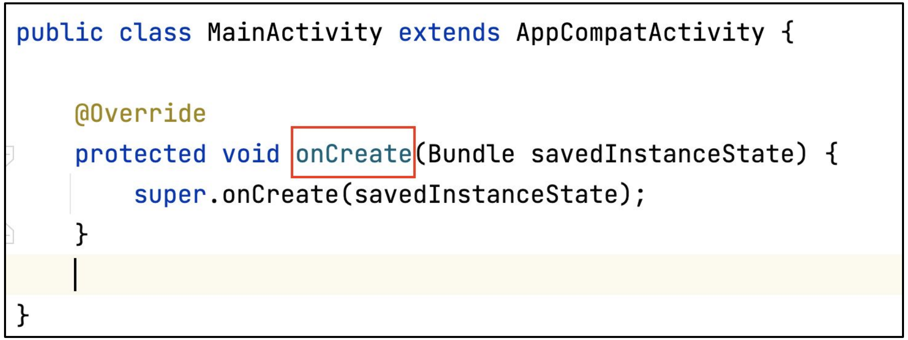
II. Activity Lifecycle

What is a **lifecycle**?

What is a lifecycle? (an example)



What is a lifecycle?



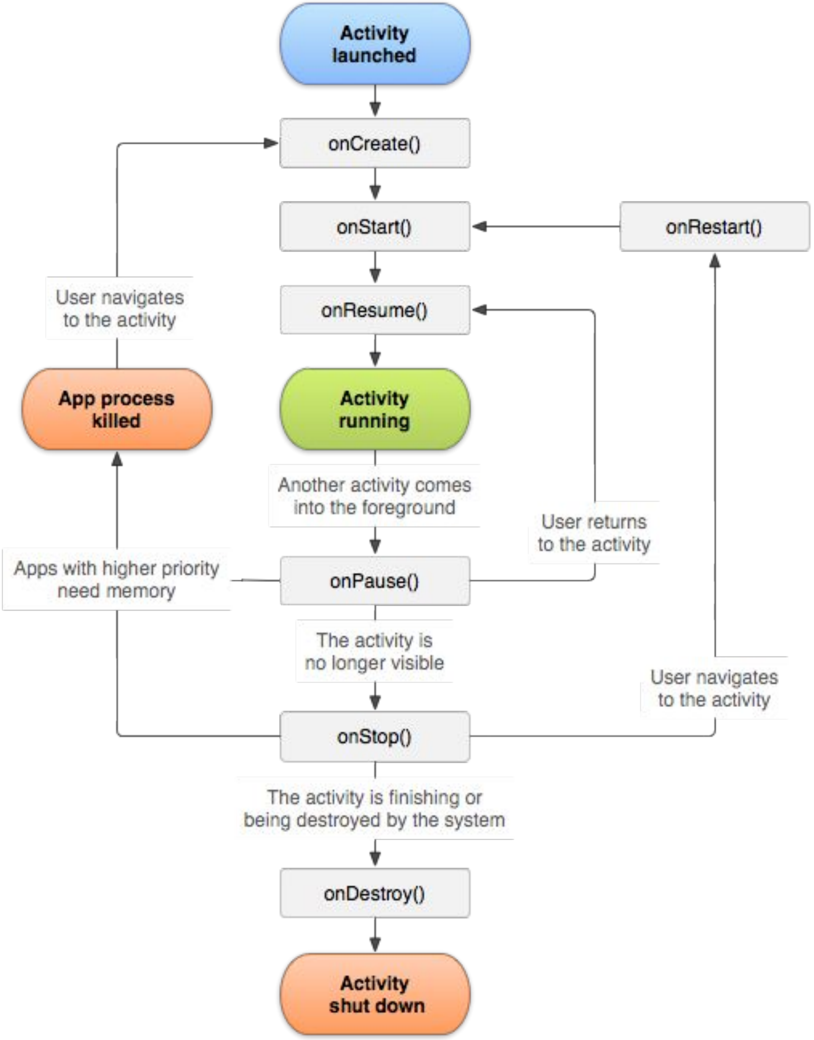
```
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }

}
```

Similarly, an **Activity** is a component defined in Android which has its **own lifecycle**.

Every stage in the Activity lifecycle is **called automatically by the Android OS** based on the **Current State of the Activity**.

II. Activity Lifecycle

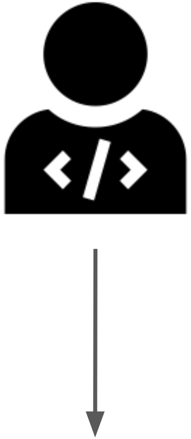
The Android OS calls these lifecycle methods when **Conditions** defined for each of these callbacks are triggered.


Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

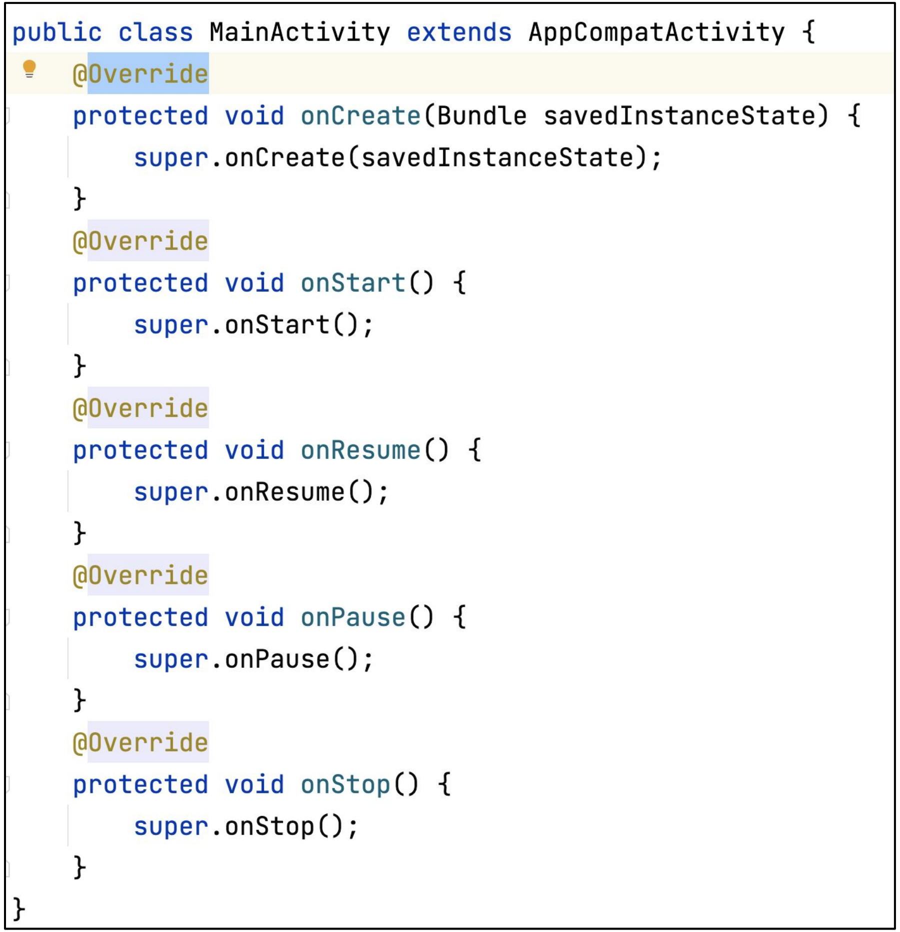
Link: Understanding Activity Lifecycle for Beginners

Link: What code should be written in each of the lifecycle methods?

II. Activity Lifecycle



DO NOT define these methods by yourself!
Just override these methods and add your own logic!

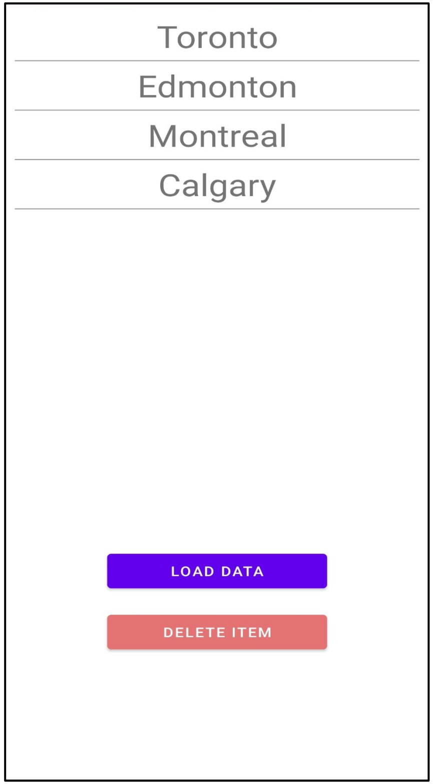


```
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }
    @Override
    protected void onStart() {
        super.onStart();
    }
    @Override
    protected void onResume() {
        super.onResume();
    }
    @Override
    protected void onPause() {
        super.onPause();
    }
    @Override
    protected void onStop() {
        super.onStop();
    }
}
```

II. Activity Lifecycle

DEMO

What’s Next?



Toronto

Edmonton

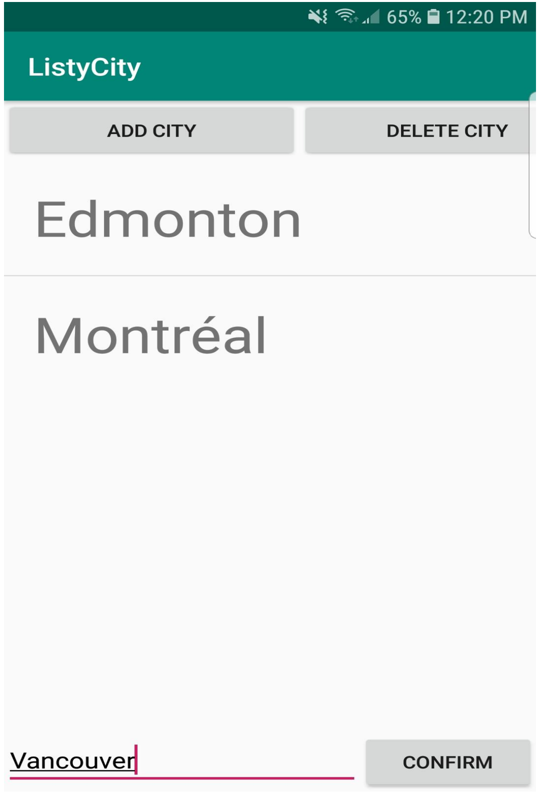
Montreal

Calgary

ListCity

1. Displaying a list of items(cities) using ListView/ListAdapter
2. How to add/ remove items from listAdapter

What’s Next?



Edmonton

Montréal

Vancouver

- **You** are expected to:
  a. Modify/Expand ListyCity to allow for the **addition of new cities** as well as the **deletion of existing ones** in the ListView.
  b. The specifics of the design of this functionality are up to you.

NOTE: This will help you to complete Assignment-1