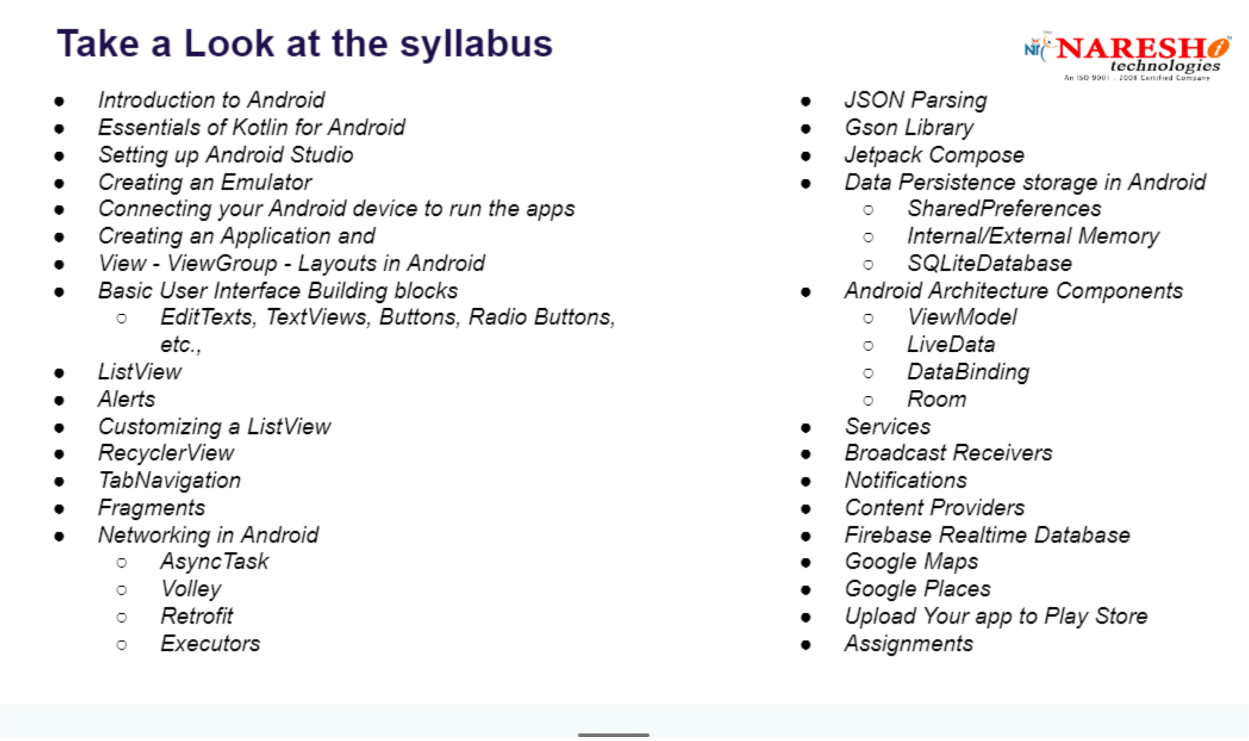
Kotlin Android Development - Naresh Tech - Batch 23 - Demo Notes

About the course:

- A 45 day course (1.5 hours each day).
- 20+ apps around the core concepts
- 5 days a week sessions.

What is the syllabus ?



Take a Look at the syllabus

- *Introduction to Android*
- *Essentials of Kotlin for Android*
- *Setting up Android Studio*
- *Creating an Emulator*
- *Connecting your Android device to run the apps*
- *Creating an Application and*
- *View - ViewGroup - Layouts in Android*
- *Basic User Interface Building blocks*
  - *EditTexts, TextViews, Buttons, Radio Buttons, etc.,*
- *ListView*
- *Alerts*
- *Customizing a ListView*
- *RecyclerView*
- *TabNavigation*
- *Fragments*
- *Networking in Android*
  - *AsyncTask*
  - *Volley*
  - *Retrofit*
  - *Executors*
- *JSON Parsing*
- *Gson Library*
- *Jetpack Compose*
- *Data Persistence storage in Android*
  - *SharedPreferences*
  - *Internal/External Memory*
  - *SQLiteDatabase*
- *Android Architecture Components*
  - *ViewModel*
  - *LiveData*
  - *DataBinding*
  - *Room*
- *Services*
- *Broadcast Receivers*
- *Notifications*
- *Content Providers*
- *Firebase Realtime Database*
- *Google Maps*
- *Google Places*
- *Upload Your app to Play Store*
- *Assignments*

How to Install Android Studio ?

1. First check the system requirements on this link
2. Download the latest version of android studio from here (Based on the operating system that you use, you will be able to download compatible android studio executable file)
3. Follow the instructions given in the first link and complete the installation.

Read My e-book here (in Java)

Link

How to create your first project using kotlin in android ?

Follow this codelab

Project structure in android studio

Please use this link

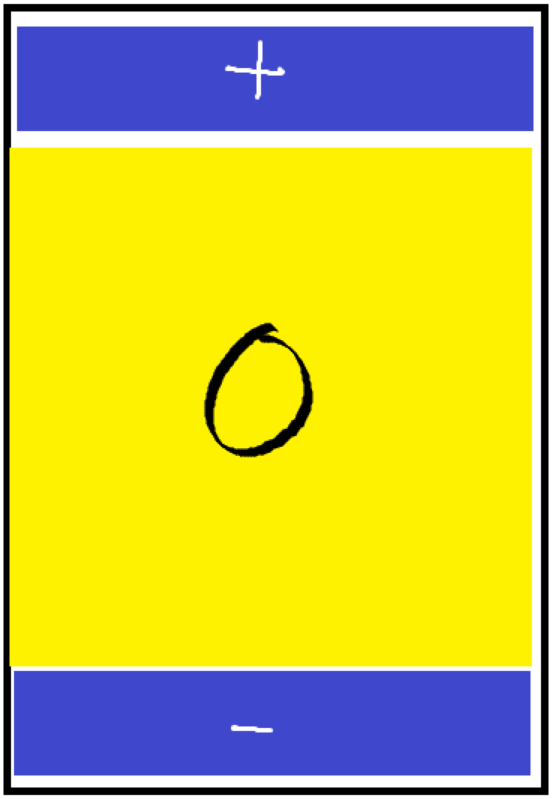
How to connect your Physical device to android studio ?

Please use this codelab

Closing android studio

File -> close project

First Project - Score Tracker



What is a **View** ?

Any UI component is called as a view in android

What is a **ViewGroup** ?

A View Group is also a view that can hold other views inside of it.

Basic Design Code:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
```

```
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:id="@+id/main"
   android:orientation="vertical"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   tools:context=".MainActivity">

   <Button
       android:text="+"
       android:layout_width="match_parent"
       android:layout_height="wrap_content"
       />

   <TextView
       android:text="0"
       android:textSize="100sp"
       android:textColor="#000000"
       android:background="#FFEB3B"
       android:layout_width="match_parent"
       android:layout_height="wrap_content"
       />

   <Button
       android:text="-"
       android:layout_width="match_parent"
       android:layout_height="wrap_content"
       />
```

```
</LinearLayout>
```

Demo Sessions

Day-1 https://youtu.be/AHWxzvlgIvg
Day-2 https://youtu.be/cYJjjBrUDYM
Day-3 https://youtu.be/yHjF9fWCcK0
Day-4 https://www.youtube.com/watch?v=3HX--ElhxqM&feature=youtu.be
Day-5 https://youtu.be/frEcfMtNyyU
Day-6 https://youtu.be/IBgpUFQTUSs
Day-7 https://youtu.be/PpdTXJ1Q-DE

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout

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:orientation="vertical"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   tools:context=".MainActivity">

   <Button
       android:text="+"
       android:onClick="incrementScore"
       android:layout_weight="1"
       android:layout_width="match_parent"
       android:layout_height="wrap_content"
       />
```

```
    <TextView
        android:layout_weight="10"
        android:text="0"
        android:gravity="center"
        android:textSize="100sp"
        android:textColor="#000000"
        android:background="#FFEB3B"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        />

    <Button
        android:layout_weight="1"
        android:text="-"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        />

</LinearLayout>
```

What is a **Snack Bar** ?

Refer to this link

Snackbars provide lightweight feedback about an operation. They show a brief message at the bottom of the screen on mobile and lower left on larger devices. Snack Bars appear above all other elements on screen and only one can be displayed at a time.

We basically have two files when it comes to a screen (Activity).

1. Xml
    a. This is used to display the ui components in some order
2. .kt file

a. This is used to handle all the logic

Score Tracker code that is functional

`MainActivity.kt`

```kotlin
package com.nareshit.scoretracker

import android.os.Bundle
import android.view.View
import android.widget.TextView
import android.widget.Toast
import androidx.activity.enableEdgeToEdge
import androidx.appcompat.app.AppCompatActivity
import androidx.core.view.ViewCompat
import androidx.core.view.WindowInsetsCompat
import com.google.android.material.snackbar.Snackbar

class MainActivity : AppCompatActivity() {

   // Declare an object for TextView
   lateinit var s:TextView
   var count:Int = 0

   override fun onCreate(savedInstanceState: Bundle?) {
       super.onCreate(savedInstanceState)
       setContentView(R.layout.activity_main)
       // Connect this object(.kt) to the view(.xml)
       s = findViewById(R.id.score)
   }

   // This function is called when the + button is
clicked.
```

```kotlin
    fun incrementScore(view: View) {
        count++
        s.setText(count.toString())
    }
    // This function is called when the - button is clicked.
    fun decrementScore(view: View) {
        count--
        s.setText("$count")
    }
}
```

`activity_main.xml`

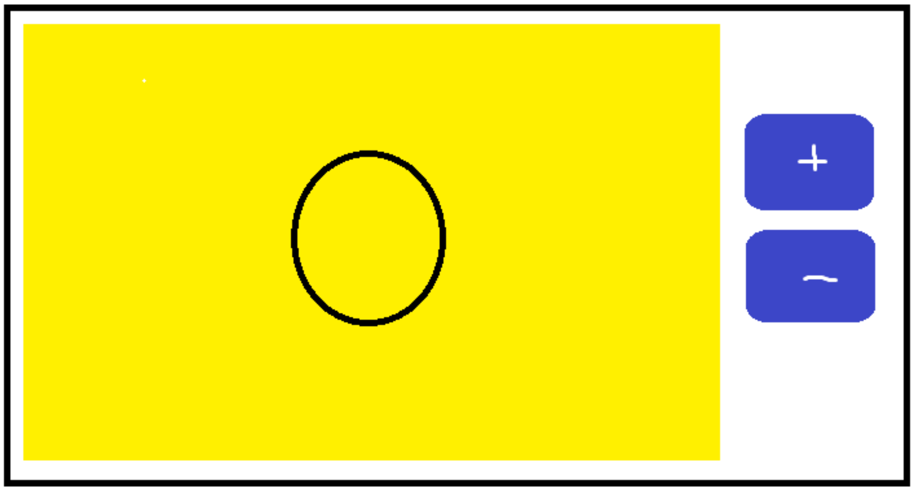
```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="vertical"
    android:layout_width="match_parent"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

    <Button
        android:text="+"
        android:onClick="incrementScore"
        android:layout_weight="1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        />

    <TextView
        android:id="@+id/score"
        android:layout_weight="10"
        android:text="0"
        android:gravity="center"
        android:textSize="100sp"
        android:textColor="#000000"
        android:background="#FFEB3B"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        />

    <Button
        android:onClick="decrementScore"
        android:layout_weight="1"
        android:text="-"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        />

</LinearLayout>
```

Assignments (for Home)

May 22, 2024

#1

The score tracker screen can be designed using the code shown in this section. Please design a separate screen for the landscape orientation of the same screen by seeing the picture down below.



Once done with the assignment, email me the code for activity_main.xml (land) to pavankreddy.t@gmail.com

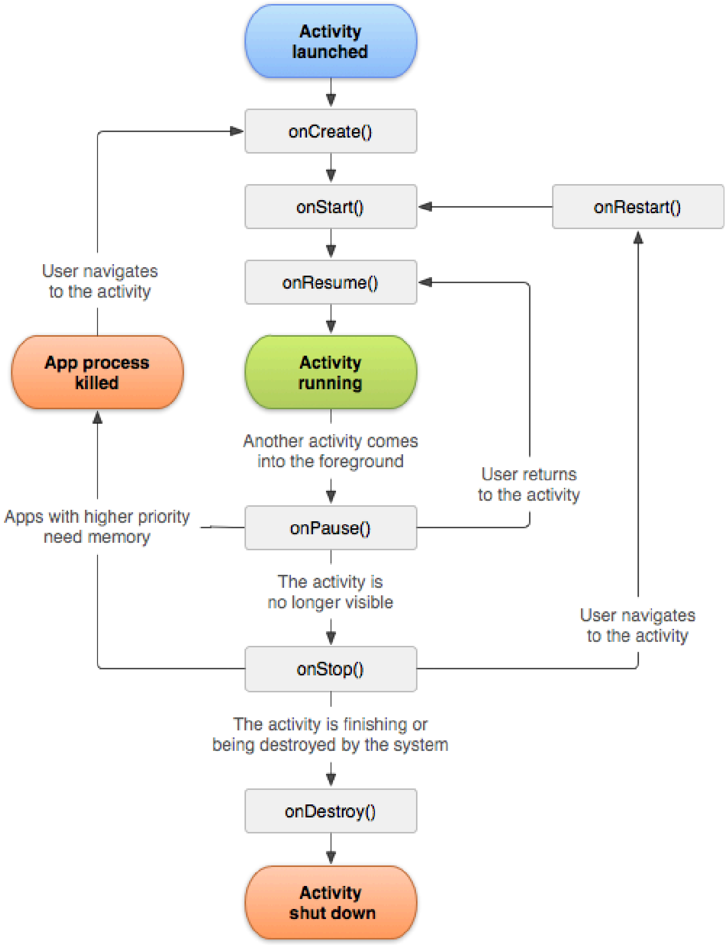
What is an Activity ?

An Activity is any single screen that the user can see in your application. Generally there will be multiple screens in any given application.
Refer more on activities here.

Activities have their own lifecycle.

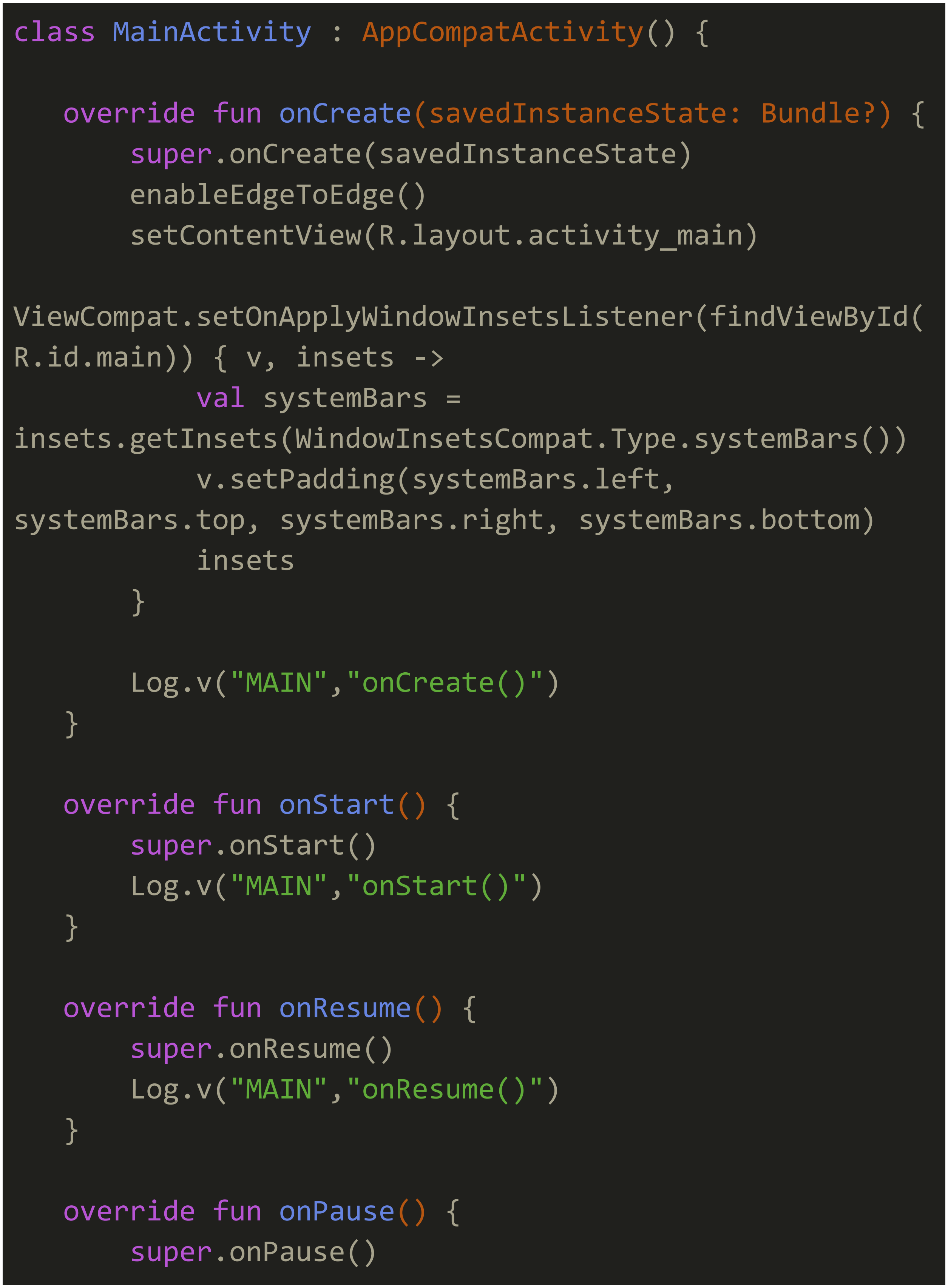
What is a lifecycle ?

Lifecycle means that from the time the activity is created till the time it is destroyed whatever the states that it goes through all come under lifecycle.


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

Understanding activity lifecycle

```
package com.nareshit.activitylifecycle

import android.os.Bundle
import android.util.Log
import androidx.activity.enableEdgeToEdge
import androidx.appcompat.app.AppCompatActivity
import androidx.core.view.ViewCompat
import androidx.core.view.WindowInsetsCompat
```



```
class MainActivity : AppCompatActivity() {

   override fun onCreate(savedInstanceState: Bundle?) {
       super.onCreate(savedInstanceState)
       enableEdgeToEdge()
       setContentView(R.layout.activity_main)

ViewCompat.setOnApplyWindowInsetsListener(findViewById(
R.id.main)) { v, insets ->
           val systemBars =
insets.getInsets(WindowInsetsCompat.Type.systemBars())
           v.setPadding(systemBars.left,
systemBars.top, systemBars.right, systemBars.bottom)
           insets
       }

       Log.v("MAIN","onCreate()")
   }

   override fun onStart() {
       super.onStart()
       Log.v("MAIN","onStart()")
   }

   override fun onResume() {
       super.onResume()
       Log.v("MAIN","onResume()")
   }

   override fun onPause() {
       super.onPause()
```

```
        Log.v("MAIN","onPause()")
    }

    override fun onStop() {
        super.onStop()
        Log.v("MAIN","onStop()")
    }

    override fun onDestroy() {
        super.onDestroy()
        Log.v("MAIN","onDestroy()")
    }

    override fun onRestart() {
        super.onRestart()
        Log.v("MAIN","onRestart()")
    }
}
```

How to persist data across the configuration change of the device (Changing orientation) ?

What is the configuration change ?

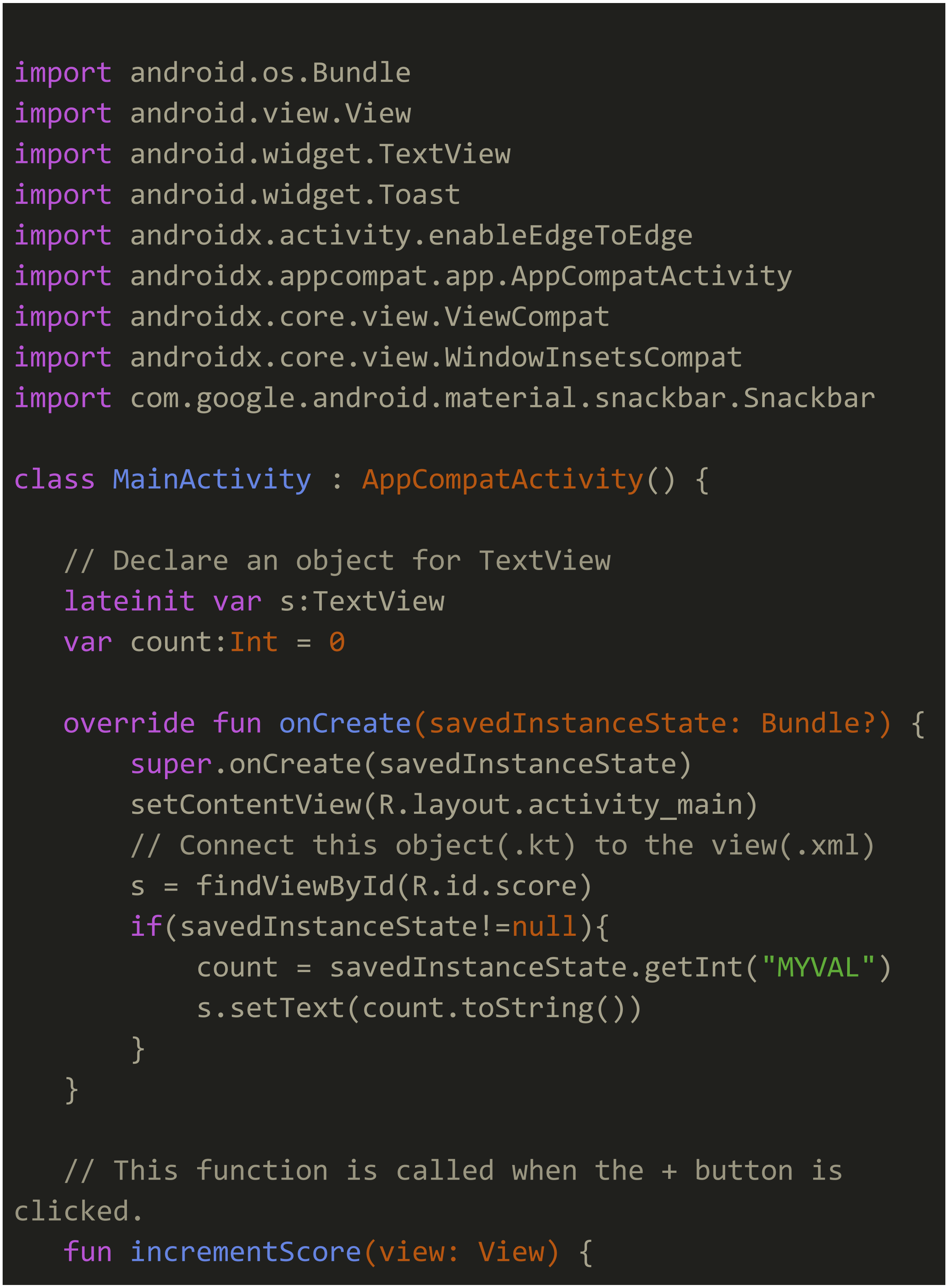
- Changing the orientation of the device
- Changing the locale(Language) of the Device/ App
- Learn More

Data Persistence upon configuration change in android?

- Handle Device configuration change

Changed code for handling configuration change so that the data is persisted.

```
package com.nareshit.scoretracker
```



```
import android.os.Bundle
import android.view.View
import android.widget.TextView
import android.widget.Toast
import androidx.activity.enableEdgeToEdge
import androidx.appcompat.app.AppCompatActivity
import androidx.core.view.ViewCompat
import androidx.core.view.WindowInsetsCompat
import com.google.android.material.snackbar.Snackbar

class MainActivity : AppCompatActivity() {

   // Declare an object for TextView
   lateinit var s:TextView
   var count:Int = 0

   override fun onCreate(savedInstanceState: Bundle?) {
       super.onCreate(savedInstanceState)
       setContentView(R.layout.activity_main)
       // Connect this object(.kt) to the view(.xml)
       s = findViewById(R.id.score)
       if(savedInstanceState!=null){
           count = savedInstanceState.getInt("MYVAL")
           s.setText(count.toString())
       }
   }

   // This function is called when the + button is
clicked.
   fun incrementScore(view: View) {
        count++
        s.setText(count.toString())
    }
    // This function is called when the - button is
clicked.
    fun decrementScore(view: View) {
        count--
        s.setText("$count")
    }

    // TODO 1: Override onSaveInstanceState(...)
function
    //  This function helps us save the current instance
of the activity
    //  so that the same instance can be used when the
activity gets re-created.
    override fun onSaveInstanceState(outState: Bundle) {
        super.onSaveInstanceState(outState)
        // Here I would to save count variable's value
for the next instance
        // You will be saving it in outstate Object.
        outState.putInt("MYVAL",count)
    }
}
```

How to make your app for Day Mode and Night modes ?

Step 1 : Create references for the colors you use in the project in colors.xml
Step 2 : create a separate colors.xml for night mode.

Download the completed project