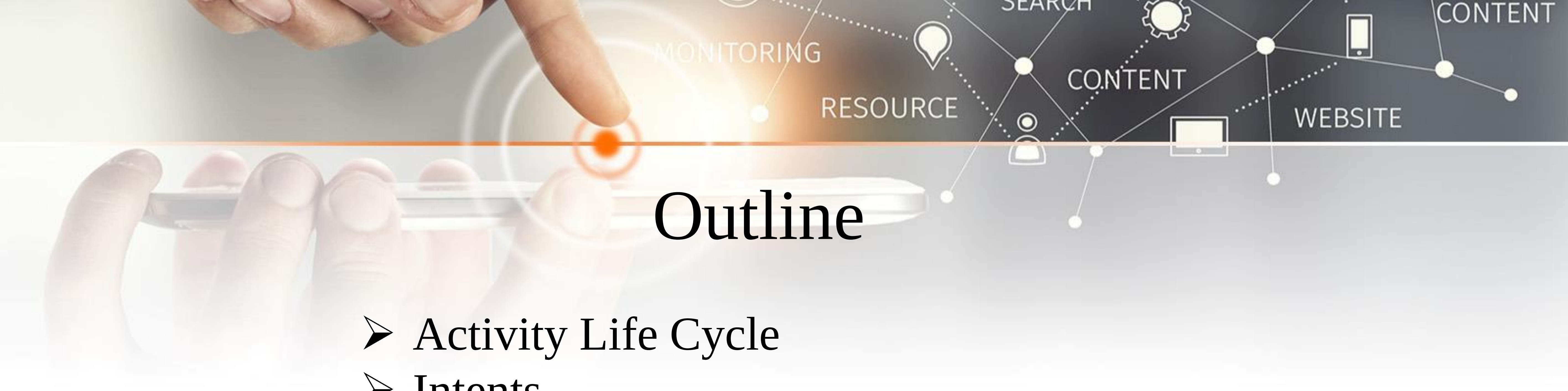
Activity Life Cycle & Intents

Asst. Prof. Monlica Wattana, Ph.D
Department of Computer Science,
Khon Kaen University

342 267 Mobile Device Programming



Outline

- Activity Life Cycle
- Intents
- Intents with Parcelable

Android Life Cycle

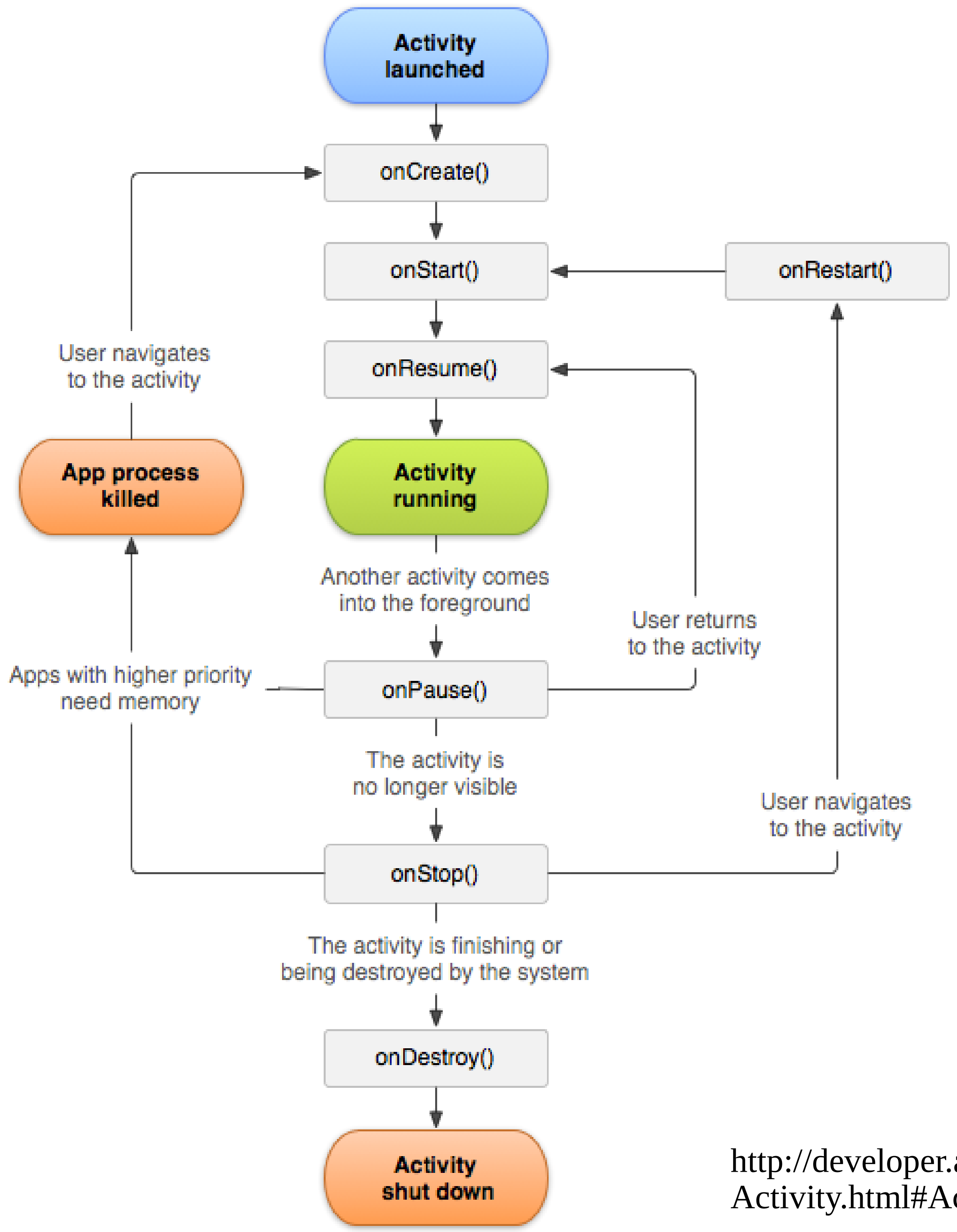
- Each application runs in its own process.
- Each activity of an app is run in the apps process
- Processes are started and stopped as needed to run an apps components.
- Processes may be killed to reclaim needed resources.
- Killed apps may be restored to their last state when requested by the user
- The steps that an application goes through from starting to finishing

Activity state

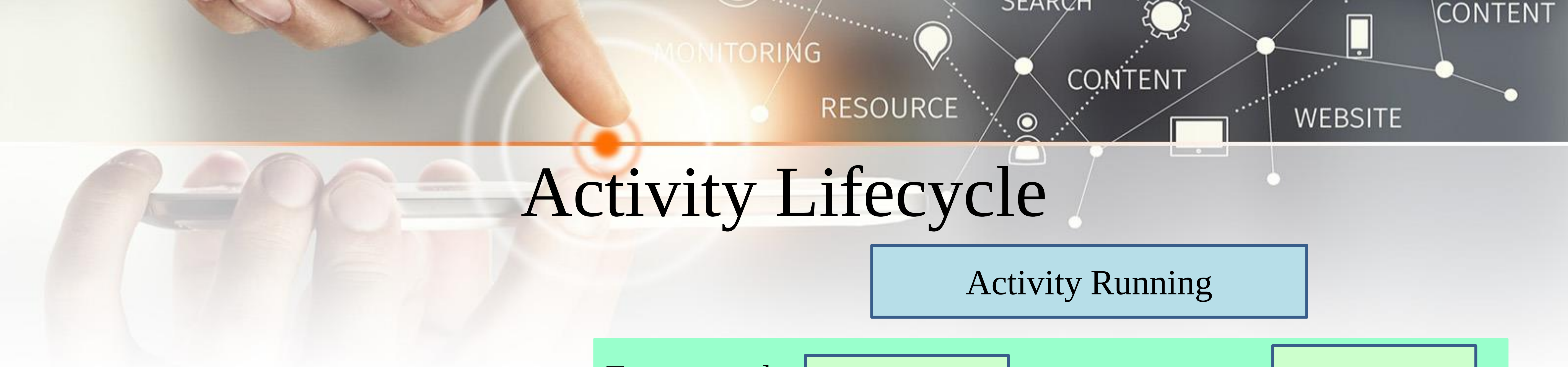
- An activity can be thought of as being in one of several states:
  - starting: In process of loading up, but not fully loaded.
  - running: Done loading and now visible on the screen.
  - paused: Partially obscured or out of focus, but not shut down.
  - stopped: No longer active, but still in the device's active memory.
  - destroyed: Shut down and no longer currently loaded in memory.

Activity Lifecycle

- Transitions between these states are represented by events that programmer can listen to in their activity code.
- The activity class has the following method **callbacks** to help programmer manage the app:
  - onCreate()
  - onStart()
  - onResume()
  - onPause()
  - onStop()
  - onRestart()
  - onDestroy()

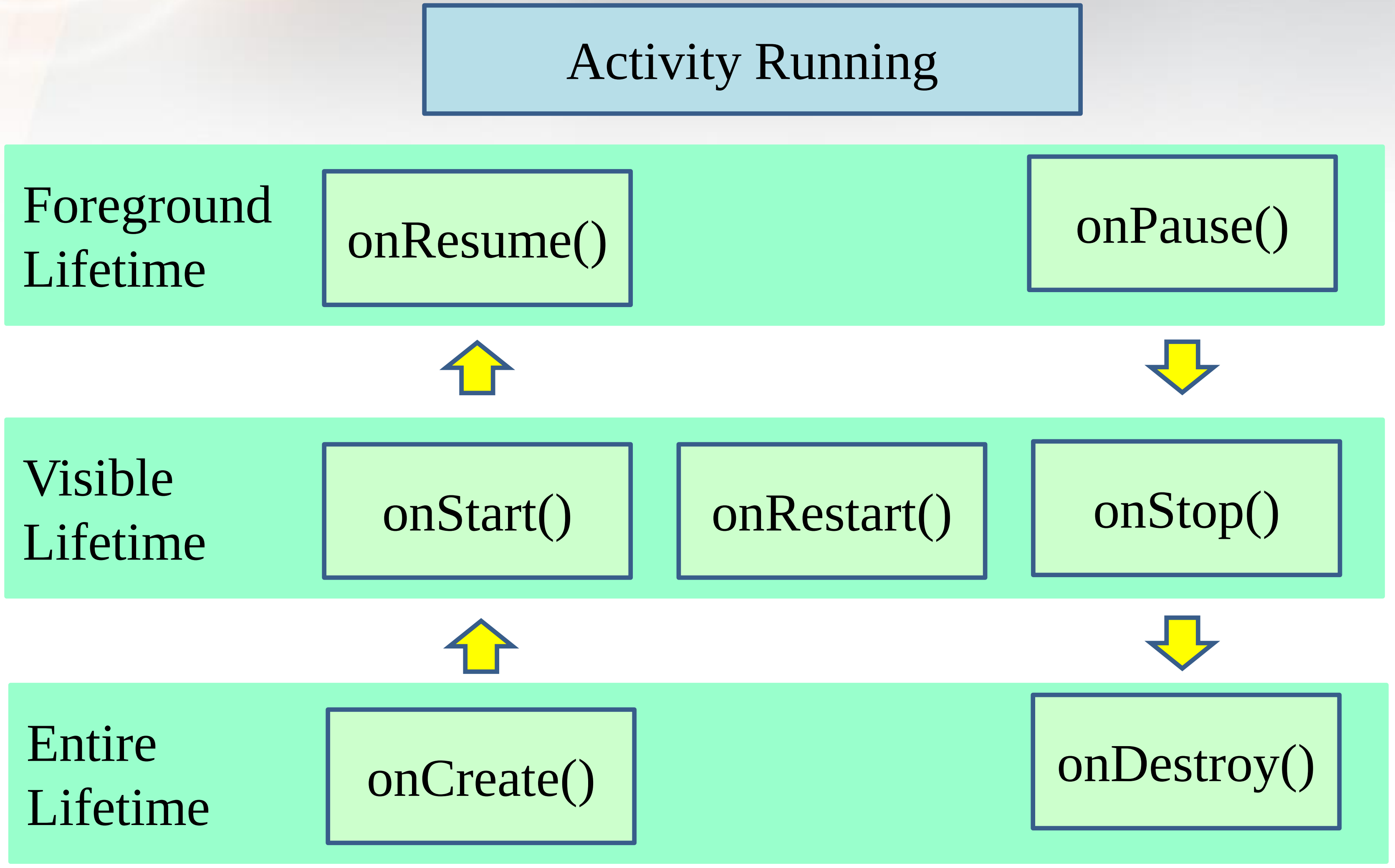

Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to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

http://developer.android.com/reference/android/app/Activity.html#ActivityLifecycle



Activity Lifecycle

3 key Loops in activity
- Entire Lifetime
- Visible Lifetime
- Foreground Lifetime



Activity Lifecycle

onCreate method

In onCreate, to create and set up the activity object, load any static resources like images, layouts, set up menus etc.

- After this, the Activity object exists

- think of this as the "constructor" of the activity

```
class MainActivity : AppCompatActivity() {
      ...
    override fun onCreate(savedInstanceState: Bundle?) {
              super.onCreate(savedInstanceState)    // always call super
              setContentView(R.layout.activity_main) // set up layout
                any other initialization code;
      }
 }
```

Activity Lifecycle

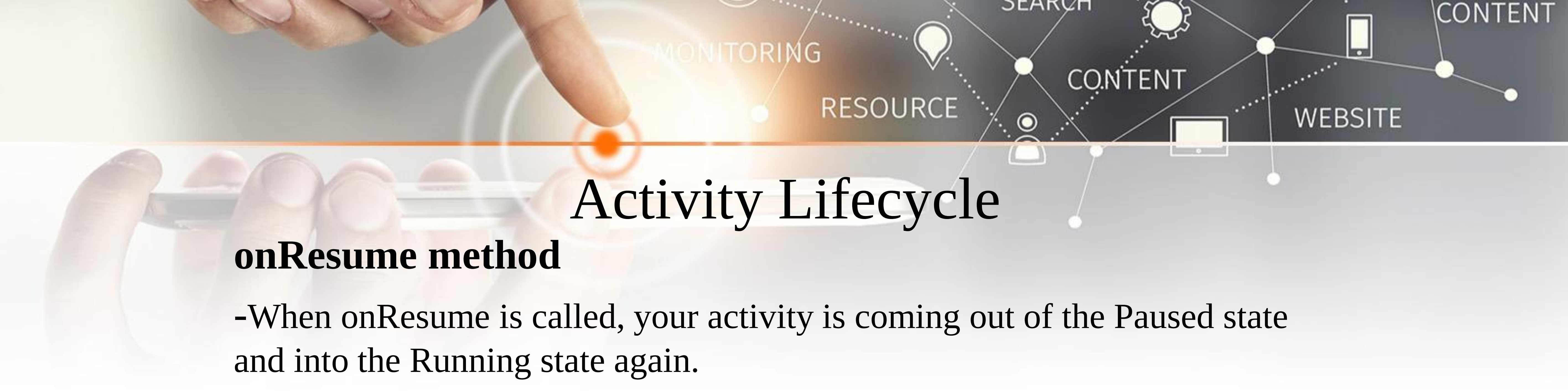
onPause method

- When onPause is called, the activity is still partially visible.

- May be temporary, or on way to termination.

  - Stop animations or other actions that consume CPU.
  - Commit unsaved changes (e.g. draft email).
  - Release system resources that affect battery life.

```
override fun onPause() {
        super.onPause() // always call super
        print("onPause")
    }
}
```



Activity Lifecycle

onResume method

-When onResume is called, your activity is coming out of the Paused state and into the Running state again.

- Also called when activity is first created/loaded!

- **Initialize resources** that you will release in onPause.
- **Start/resume animations** or other ongoing actions that should only run when activity is visible on screen.

```
override fun onResume() {
        super.onResume() // always call super
        if (myConnection == null) {
                myConnection = new ExampleConnect() // init.resources
                myConnection.connect();
        }
}
```

Activity Lifecycle

onStop method

-When onStop is called, your activity is no longer visible on the screen:

- User chose another app from Recent Apps window.
- User starts a different activity in your app.
- User receives a phone call while in your app.

- App might still be running, but that activity is not.

- onPause is always called before onStop.
- onStop performs heavy-duty shutdown tasks like writing to a database.

```
override fun onStop() {
        super.onStop()   // always call super
            ...
}
```

Activity Lifecycle

onStart and onRestart

- onStart is called every time the activity begins.

- onRestart is called when activity was stopped but is started again later (all but the first start).

  - Not as commonly used; favor onResume.
  - Re-open any resources that onStop closed.

```
override fun onStart() {
		 super.onStart(); // always call super
		...
}
```

```
override fun onRestart() {
		super.onRestart(); // always call super
		 ...
}
```

Activity Lifecycle

onDestroy method

-When onDestroy is called, your entire app is being shut down and unloaded from memory.

- Unpredictable exactly when/if it will be called.
- Can be called whenever the system wants to reclaim the memory used by your app.
- Generally, favor onPause or onStop because they are called in a predictable and timely manner.

```
override fun onDestroy() {
		super.onDestroy(); // always call super
		...
}
```

Testing activity states

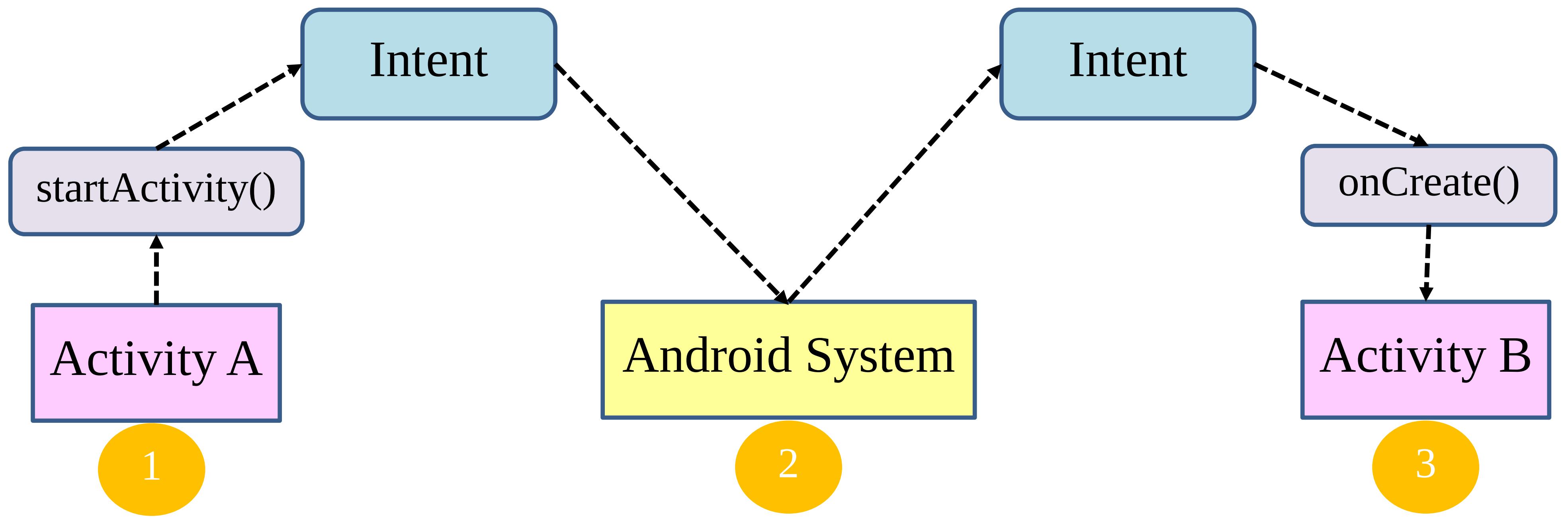
Use the LogCat system for logging messages when your app changes states:

- analogous to System.out.println debugging for Android apps
- appears in the LogCat console in Android Studio

```
override fun onStart() {
            super.onStart()
            Log.i(TAG, "onStart")
}
```

Intents

- When a new Activity is started, an **Intent** object is created and passed to that Activity
- The Intent object contains information about what the Activity is meant to do, and any data it needs in order to do it



Intents

Android intents are mainly used to:

- Start the service
- Launch an activity
- Display a web page
- Display a list of contacts
- Broadcast a message
- Dial a phone call etc.

Intents

Intents type: there are 2 type

1. Explicit intent
2. Implicit intent

finish() is used to destroy an activity and remove it from the stack.

Explicit intent

Intent statement:

```
val i = Intent(applicationContext, OtherActivity::class.java)
startActivity(i)
```

An explicit intent tells Android system to run specific component such as ActivityB

```
val i = Intent(applicationContext, ActivityB::class.java)
startActivity(i)
```

Implicit intent

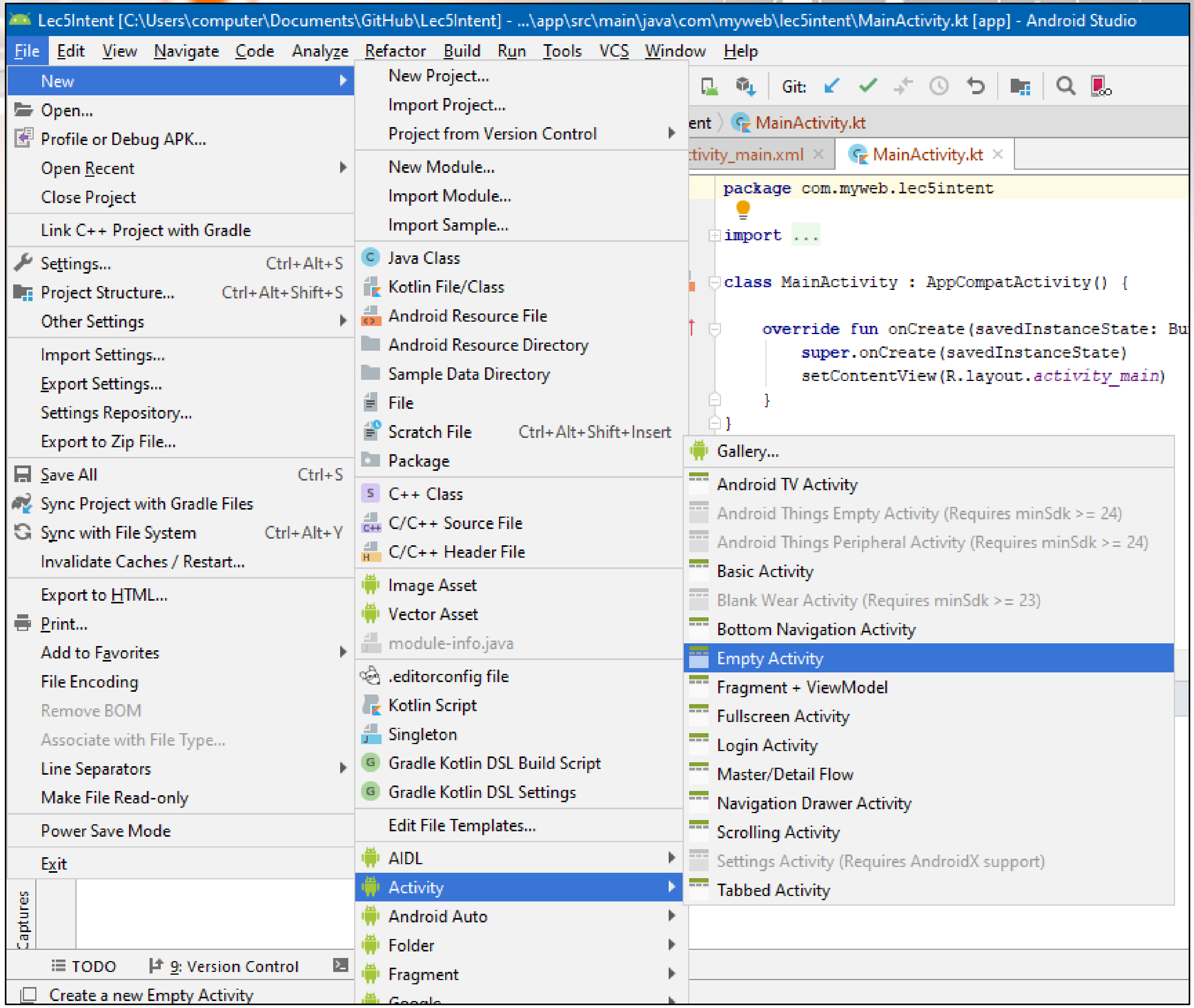
- Implicit intents specify the action which should be performed and optionally data which provides content for the action.
- If an implicit intent is sent to the Android system, it searches for all components which are registered for the specific action and the fitting data type.
- If only one component is found, Android starts this component directly.
- If several components are identified by the Android system, the user will get a selection dialog and can decide which component should be used for the intent

Implicit intent

- For example, the following tells the Android system to view a webpage. All installed web browsers should be registered to the corresponding intent data via an intent filter.

```
val i = Intent(Intent.ACTION_VIEW, Uri.parse("http://www.google.com"))
startActivity(i)
```

Add Activity



Lec5Intent [C:\Users\computer\Documents\GitHub\Lec5Intent] - ...\app\src\main\java\com\myweb\lec5intent\MainActivity.kt [app] - Android Studio

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

New
Open...
Profile or Debug APK...
Open Recent
Close Project
Link C++ Project with Gradle
Settings... Ctrl+Alt+S
Project Structure... Ctrl+Alt+Shift+S
Other Settings
Import Settings...
Export Settings...
Settings Repository...
Export to Zip File...
Save All Ctrl+S
Sync Project with Gradle Files
Sync with File System Ctrl+Alt+Y
Invalidate Caches / Restart...
Export to HTML...
Print...
Add to Favorites
File Encoding
Remove BOM
Associate with File Type...
Line Separators
Make File Read-only
Power Save Mode
Exit

New Project...
Import Project...
Project from Version Control
New Module...
Import Module...
Import Sample...
Java Class
Kotlin File/Class
Android Resource File
Android Resource Directory
Sample Data Directory
File
Scratch File Ctrl+Alt+Shift+Insert
Package
C++ Class
C/C++ Source File
C/C++ Header File
Image Asset
Vector Asset
module-info.java
.editorconfig file
Kotlin Script
Singleton
Gradle Kotlin DSL Build Script
Gradle Kotlin DSL Settings
Edit File Templates...
AIDL
Activity
Android Auto
Folder
Fragment
Google

Gallery...
Android TV Activity
Android Things Empty Activity (Requires minSdk >= 24)
Android Things Peripheral Activity (Requires minSdk >= 24)
Basic Activity
Blank Wear Activity (Requires minSdk >= 23)
Bottom Navigation Activity
Empty Activity
Fragment + ViewModel
Fullscreen Activity
Login Activity
Master/Detail Flow
Navigation Drawer Activity
Scrolling Activity
Settings Activity (Requires AndroidX support)
Tabbed Activity

Git:

ent 〉 MainActivity.kt

tivity_main.xml × MainActivity.kt ×

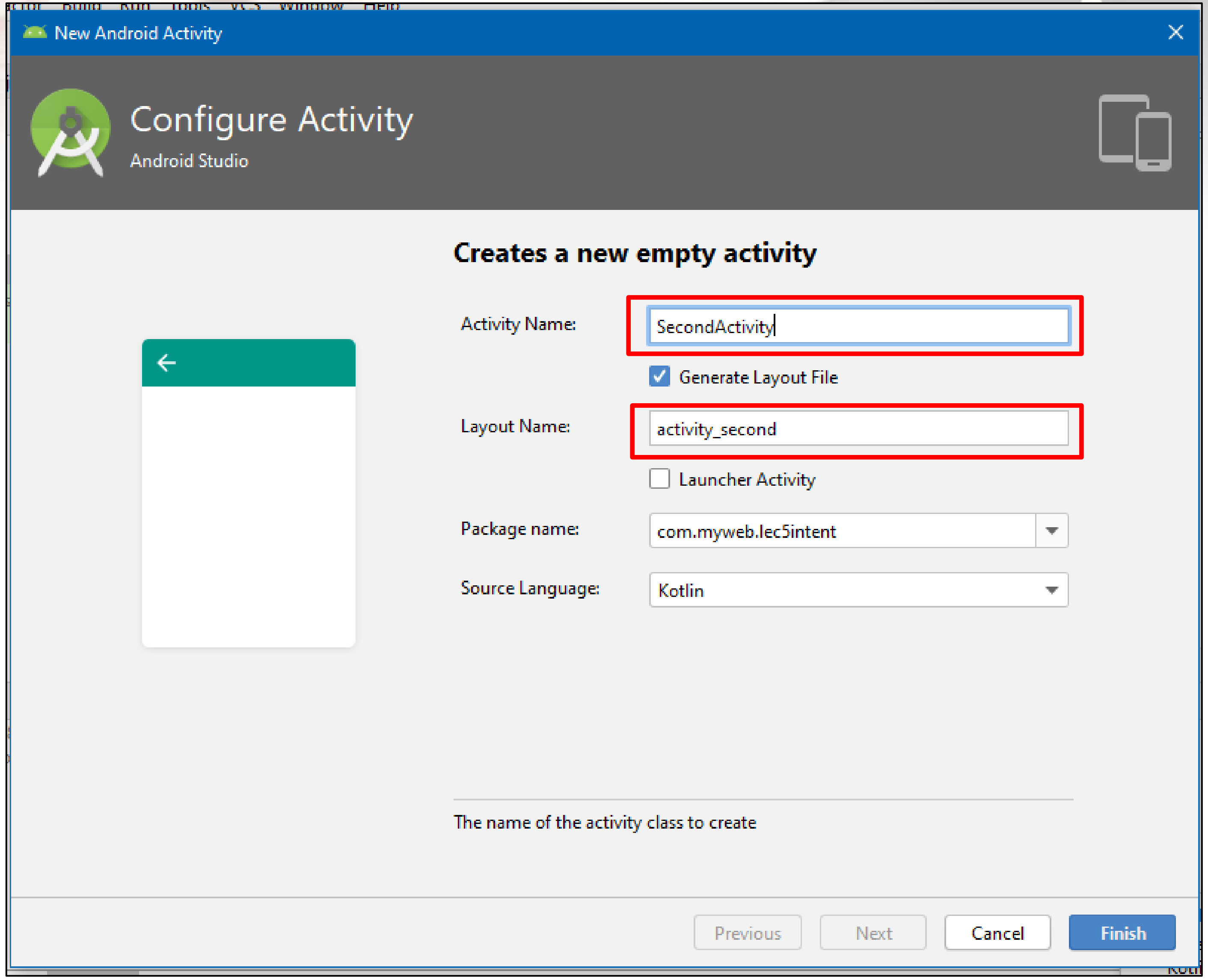
```
package com.myweb.lec5intent

import ...

class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bu
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
    }
}
```

Captures

TODO 9: Version Control

Create a new Empty Activity

Add Activity (cont.)



New Android Activity

Configure Activity

Android Studio

Creates a new empty activity

Activity Name: SecondActivity

[✓] Generate Layout File

Layout Name: activity_second

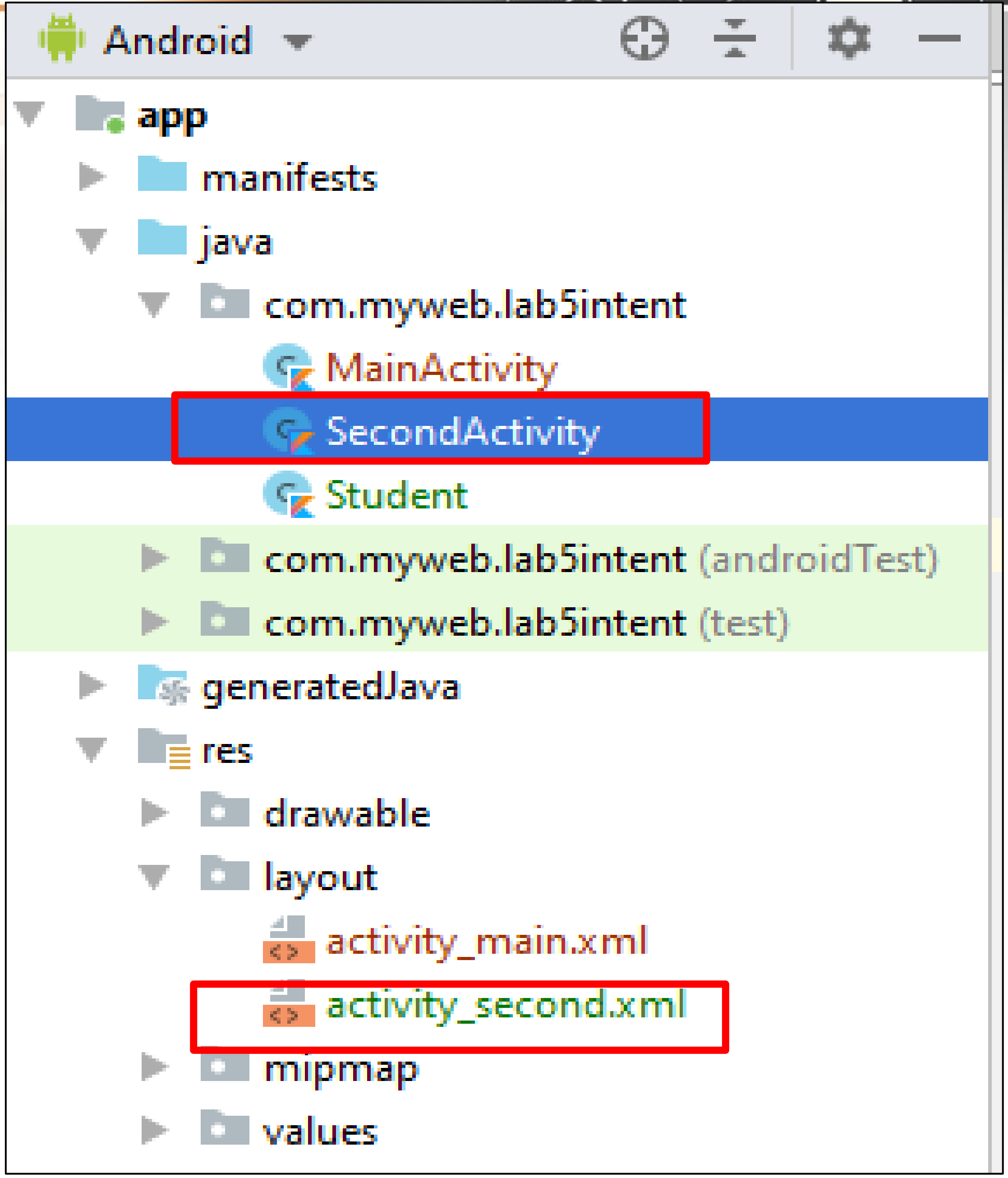
[ ] Launcher Activity

Package name: com.myweb.lec5intent

Source Language: Kotlin

The name of the activity class to create

Add Activity (cont.)



Android

- app
  - manifests
  - java
    - com.myweb.lab5intent
      - MainActivity
      - SecondActivity
      - Student
    - com.myweb.lab5intent (androidTest)
    - com.myweb.lab5intent (test)
  - generatedJava
  - res
    - drawable
    - layout
      - activity_main.xml
      - activity_second.xml
    - mipmap
    - values

Add Activity

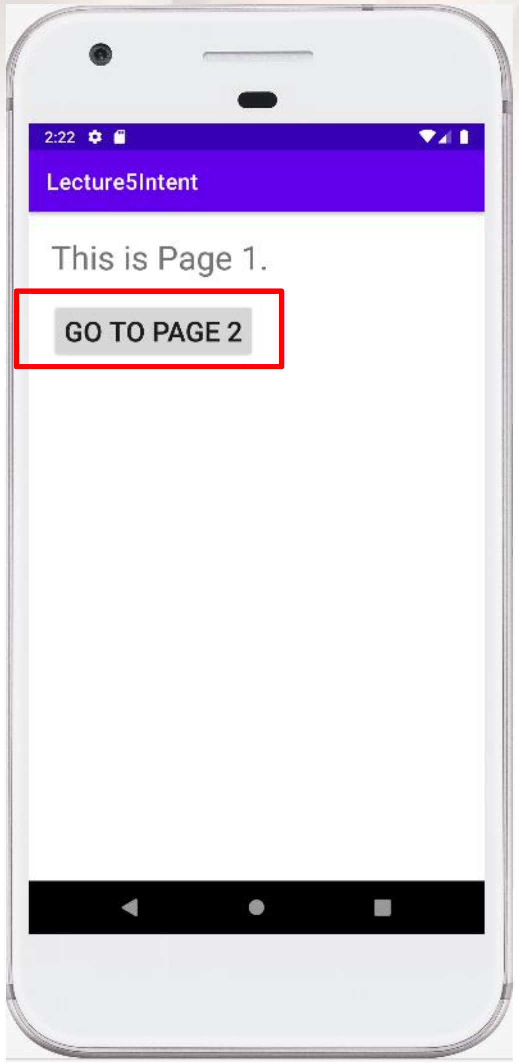
app>>manifests
>>AndroidManifest.xml

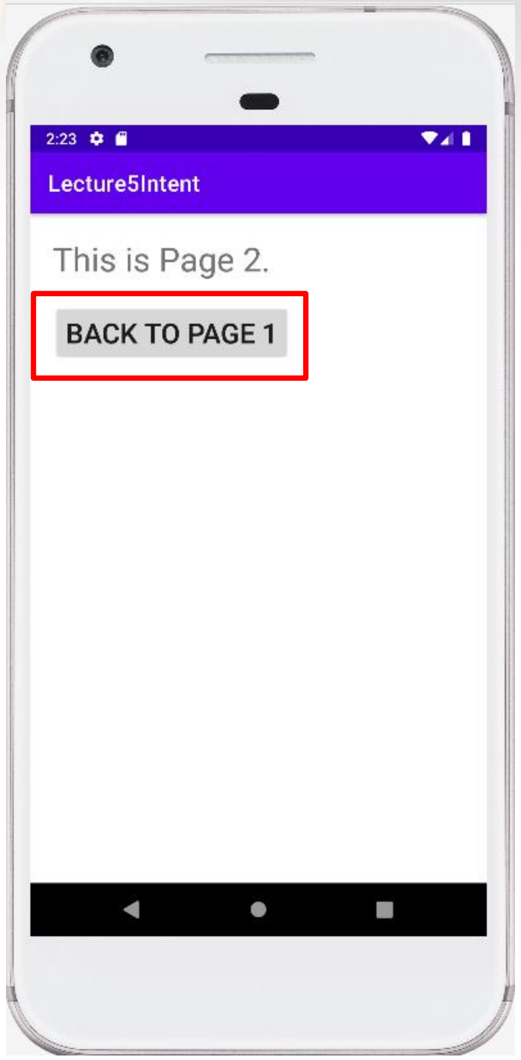
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.myweb.intenttest">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".SecondActivity"></activity>
    </application>

</manifest>
```

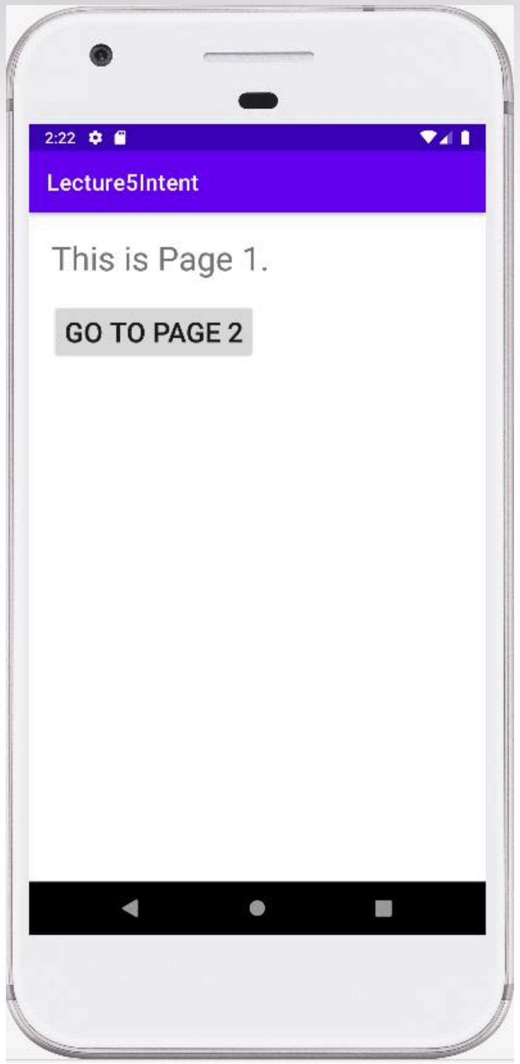
For Launcher Activity

Second Activity

Intents Example1



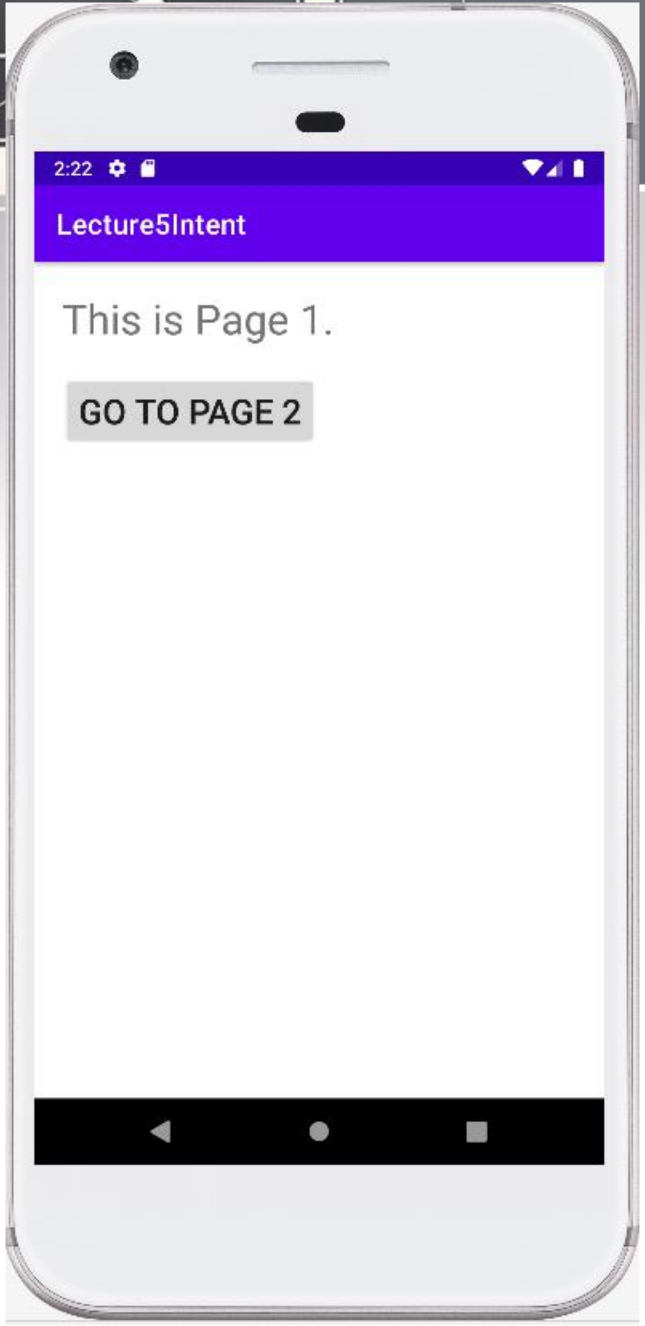




Intents Example1

activity_main.xml

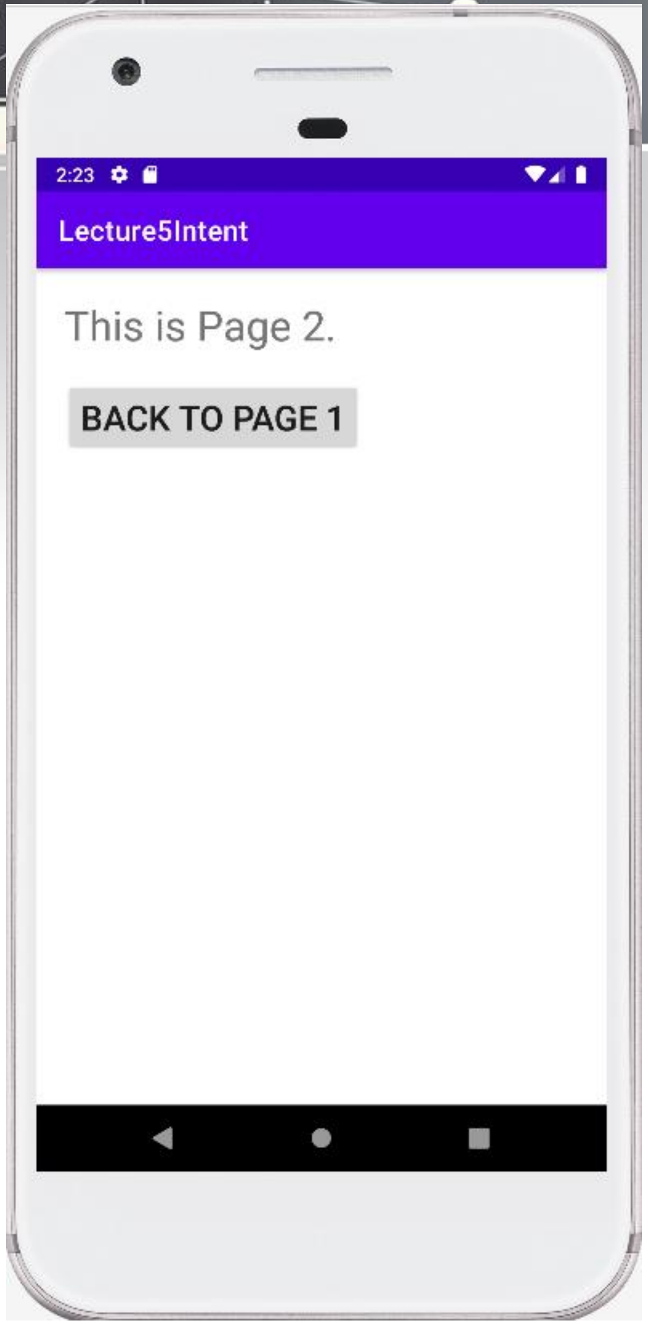
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
            xmlns:android="http://schemas.android.com/apk/res/android"
            xmlns:tools="http://schemas.android.com/tools"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            tools:context=".MainActivity"
            android:orientation="vertical">
  <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="This is Page 1."
            android:textSize= "30sp" />
  <androidx.appcompat.widget.AppCompatButton
          android:id="@+id/btnPage1"
          android:layout_width="wrap_content"
          android:layout_height="wrap_content"
          android:layout_marginTop="20dp"
          android:textSize="25sp"
          android:text="GO TO PAGE 2" />
</LinearLayout>
```



Intents Example1 (cont.)

activity_second.xml

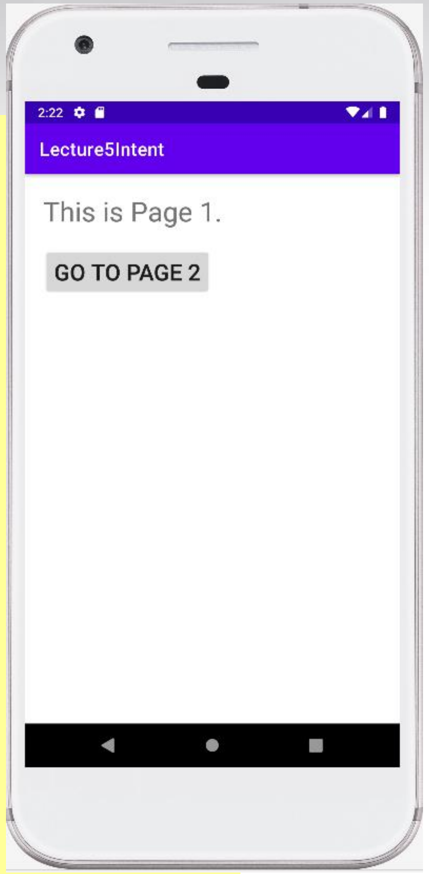
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:tools="http://schemas.android.com/tools"
      android:layout_width="match_parent"
      android:layout_height="match_parent"
      tools:context=".SecondActivity"
      android:orientation="vertical">
   <TextView
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:text="This is Page 2."
         android:textSize= "30sp"   />
   <androidx.appcompat.widget.AppCompatButton
          android:id="@+id/btnBack"
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:layout_marginTop="20dp"
         android:textSize="25sp"
         android:text="BACK TO PAGE 1"/>
</LinearLayout>
```

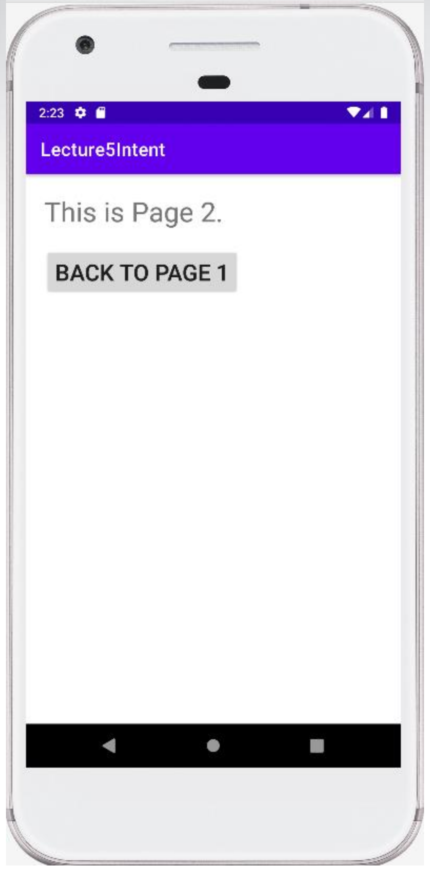


Intents Example1 (cont.)

MainActivity.kt

```
class MainActivity : AppCompatActivity() {
    private lateinit var  binding : ActivityMainBinding

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)

        //// Set Button to go activity_second
        binding.btnPage1.setOnClickListener() {
            val intent = Intent(this, SecondActivity::class.java)
            startActivity(intent)
        }
    }
}
```

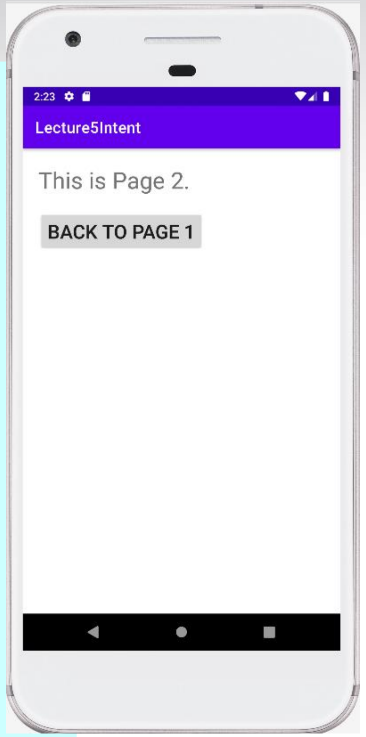


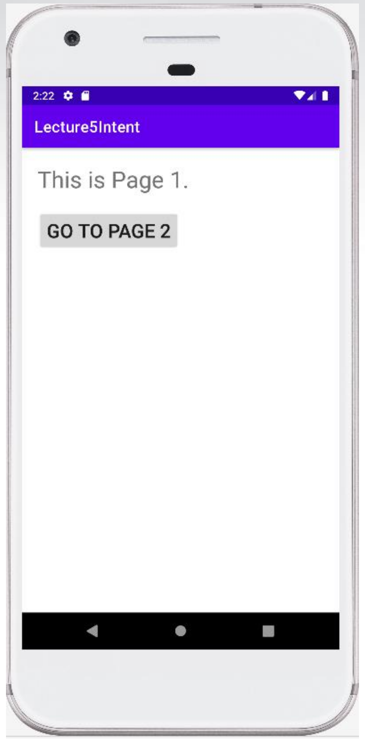


Intents Example1 (cont.)

SecondActivity.kt

```
class SecondActivity : AppCompatActivity() {
        private lateinit var  bindingSecond : ActivitySecondBinding
        override fun onCreate(savedInstanceState: Bundle?) {
            super.onCreate(savedInstanceState)

            bindingSecond = ActivitySecondBinding.inflate(layoutInflater)
            setContentView(bindingSecond.root)

            //// Set Button to go activity_main
            bindingSecond.btnBack.setOnClickListener() {
                val intent = Intent(this, MainActivity::class.java)
                startActivity(intent)
                finish() //Close SecondActivity
            }
        }
}
```


2:23
Lecture5Intent
This is Page 2.
BACK TO PAGE 1


2:22
Lecture5Intent
This is Page 1.
GO TO PAGE 2

Intents: Open Another App

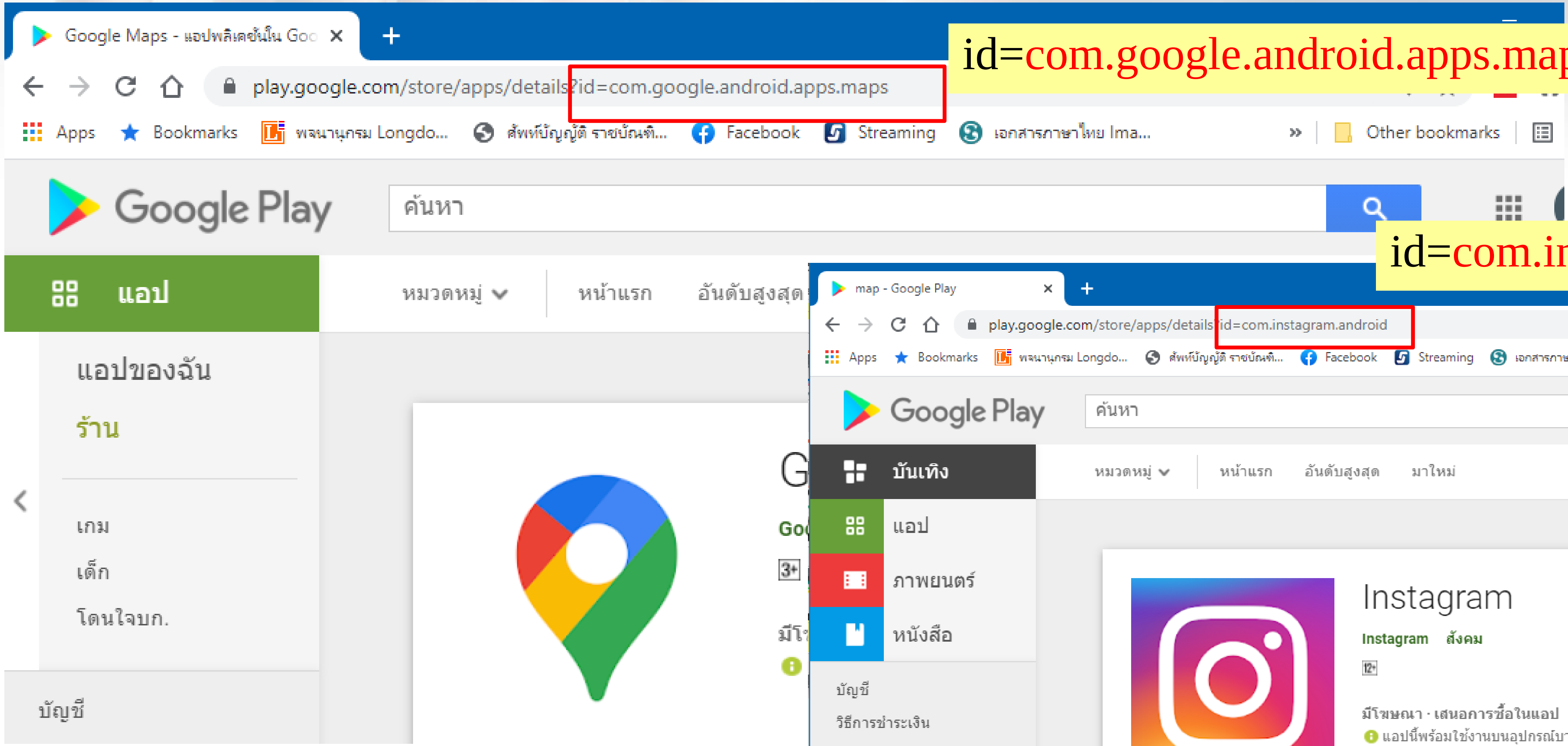
- Intents can start an activity in another app

```
val i : Intent? = packageManager.getLaunchIntentForPackage(packageName)
    i?.addCategory(Intent.CATEGORY_LAUNCHER)
    startActivity(i)
```

- No app on Mobile phone : NullPointerException

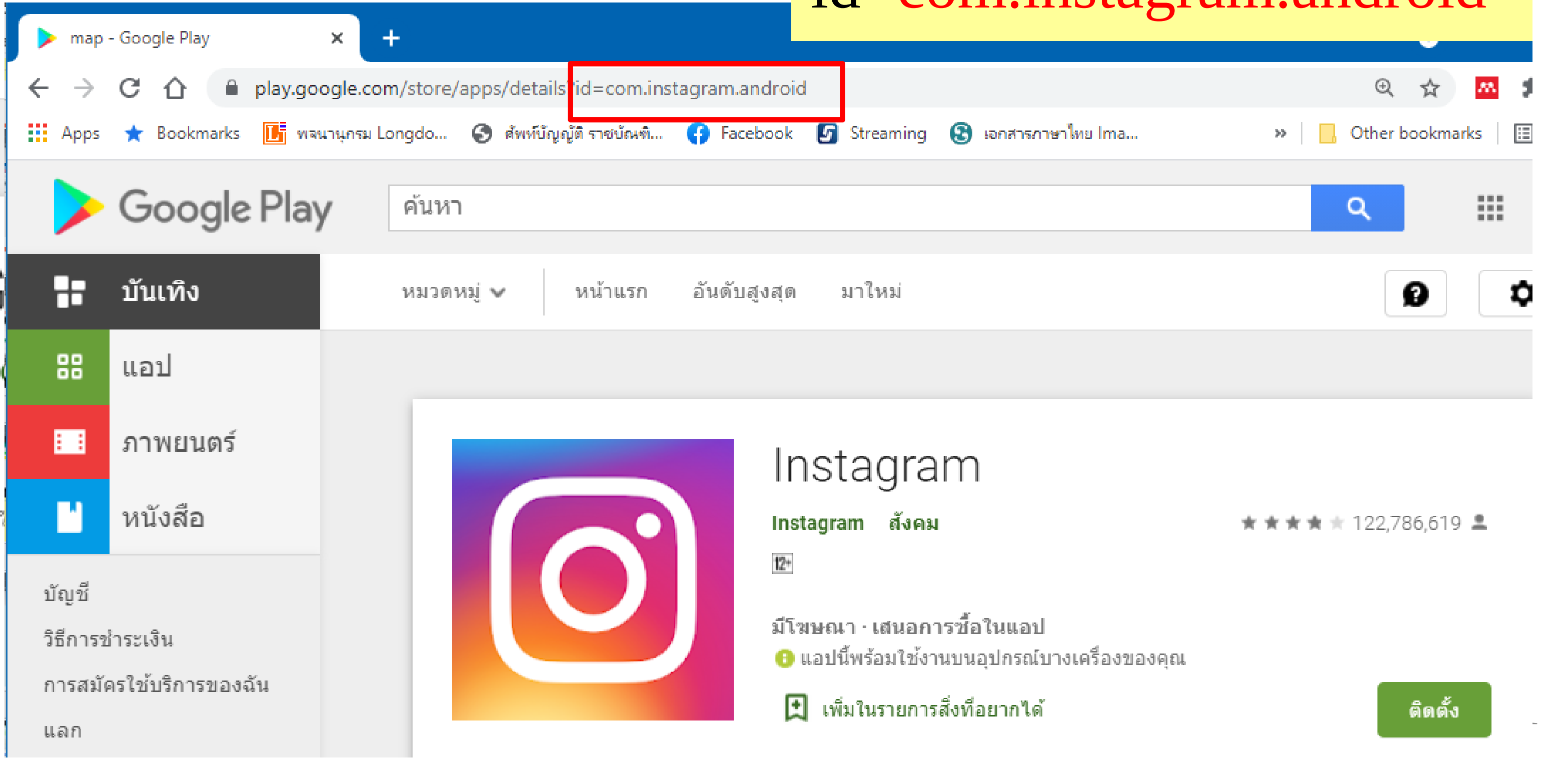
```
val i  = Intent(Intent.ACTION_VIEW)
    i.data = Uri.parse("https://play.google.com/store/apps/details?id=$packageName")
    startActivity(i)
```

Package Name



play.google.com/store/apps/details?id=com.google.android.apps.maps

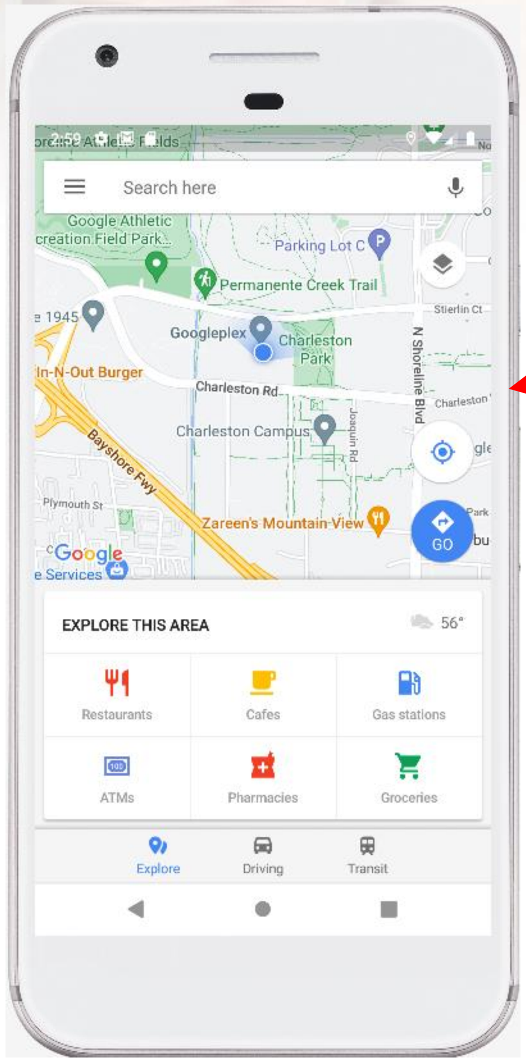
id=com.google.android.apps.maps

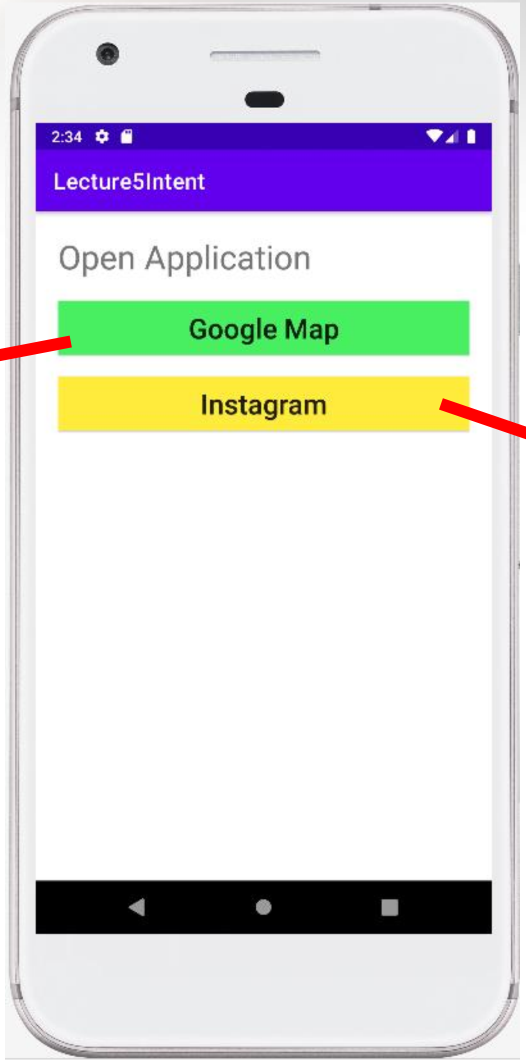


play.google.com/store/apps/details?id=com.instagram.android

id=com.instagram.android

Intents: Open Another App





2:34

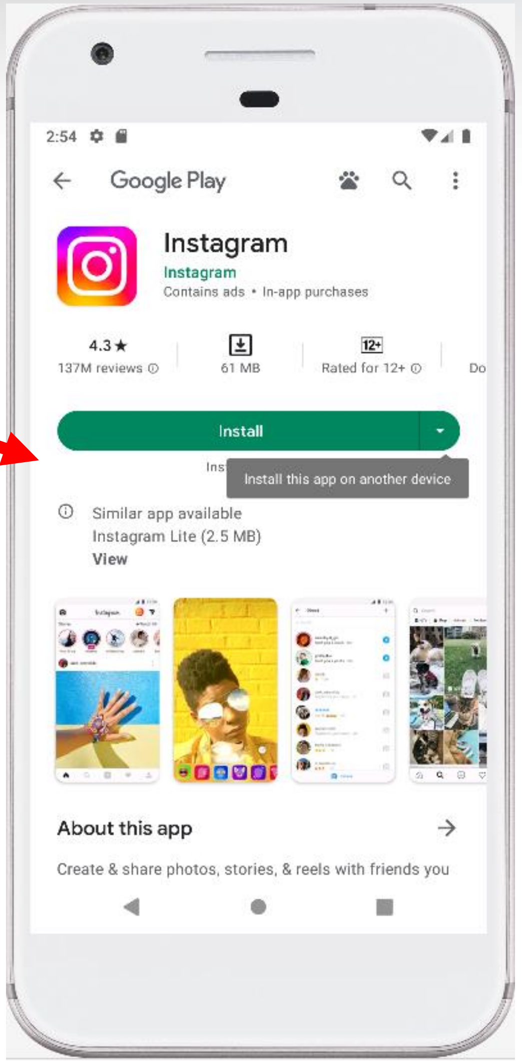
Lecture5Intent

Open Application

Google Map

Instagram

NullPointerException



2:54

Google Play

Instagram

Instagram

Contains ads • In-app purchases

4.3★ 137M reviews | 61 MB | Rated for 12+

Install

Install this app on another device

Similar app available

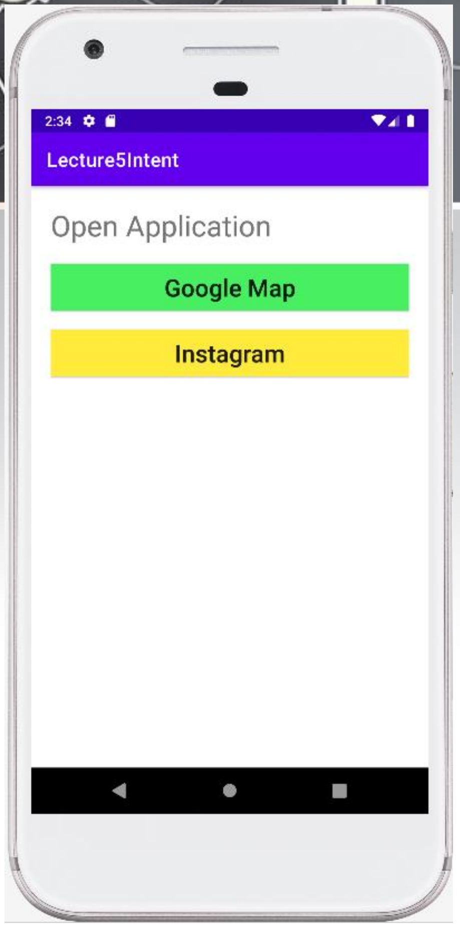
Instagram Lite (2.5 MB)

View

About this app

Create & share photos, stories, & reels with friends you

Intents: Open Another App



Open Application

Google Map

Instagram

Activity_main.xml

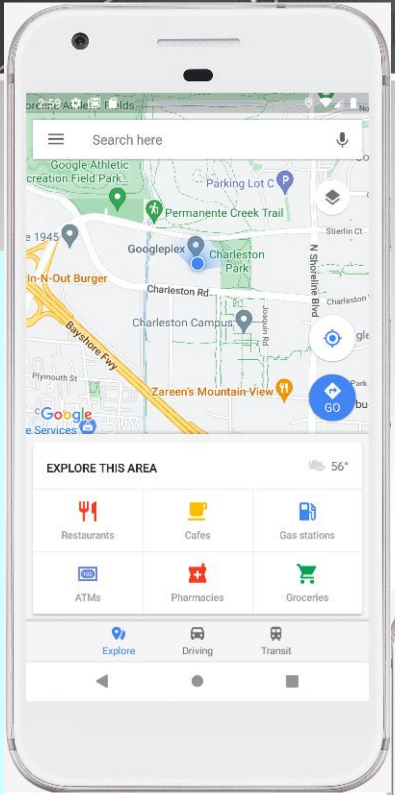
```
<?xml version="1.0"  encoding="utf-8"?>
<LinearLayout
      xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:tools="http://schemas.android.com/tools"
      android:layout_width="match_parent"
      android:layout_height="match_parent"
      tools:context=".MainActivity"
      android:orientation="vertical">
   <TextView
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:text="Open Application"
         android:textSize="30sp"
         android:layout_marginBottom="20dp" />
   <androidx.appcompat.widget.AppCompatButton
         android:id="@+id/btnMap"
         android:text="Google Map"
         android:layout_width="match_parent“
         android:layout_height="wrap_content"
         android:background="#48ef60"
         android:textSize="25sp"
         android:textAllCaps="false" />
```

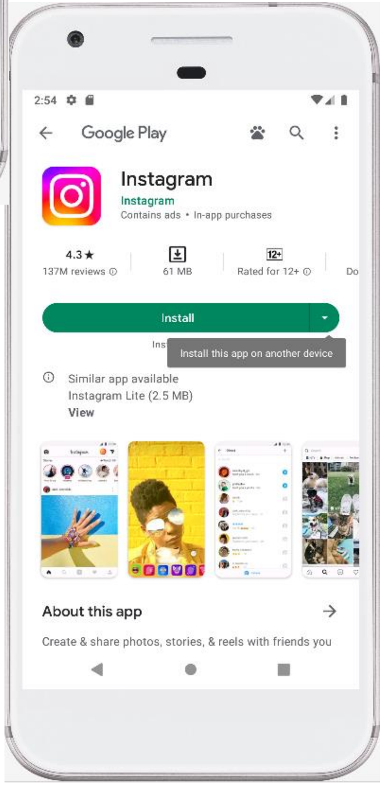
```
<androidx.appcompat.widget.AppCompatButton
          android:id="@+id/btnInstagram"
          android:text="Instagram"
          android:layout_width="match_parent"
          android:layout_height="wrap_content"
          android:background="#FFEB3B"
          android:layout_marginTop="20dp"
          android:textSize="25sp"
          android:textAllCaps="false" />
</LinearLayout>
```

Intents: Open Another App

MainActivity.kt

```
class SecondActivity : AppCompatActivity() {
        private lateinit var  binding : ActivityMainBinding
        override fun onCreate(savedInstanceState: Bundle?) {
            super.onCreate(savedInstanceState)

            binding= ActivityMainBinding.inflate(layoutInflater)
            setContentView(binding.root)

            binding.btnMap.setOnClickListener {
              val  packageName  = "com.google.android.apps.maps"
              startApp(packageName)
            }
          binding.btnInstagram.setOnClickListener {
             val  packageName = "com.instagram.android"
             startApp(packageName)
          }
}
```

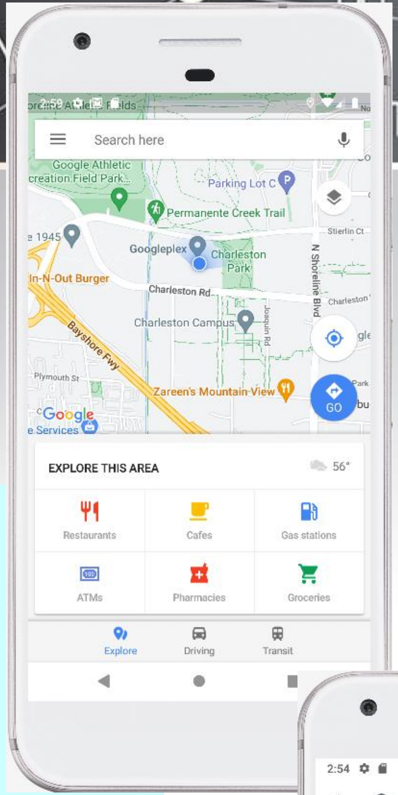


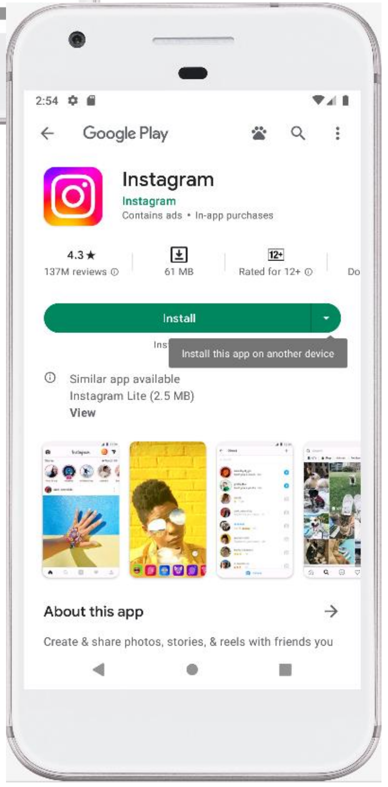


Intents: Open Another App

MainActivity.kt (cont)

```
fun startApp(packageName:String){
        try {
           val i : Intent? = packageManager.getLaunchIntentForPackage (packageName)
            i?.addCategory(Intent.CATEGORY_LAUNCHER)
            startActivity(i)
        } catch (e : NullPointerException ) {
            val i = Intent(Intent.ACTION_VIEW)
            i.data = Uri.parse("https://play.google.com/store/apps/details?id=$packageName")
            startActivity(i)
        }
    }
}
```



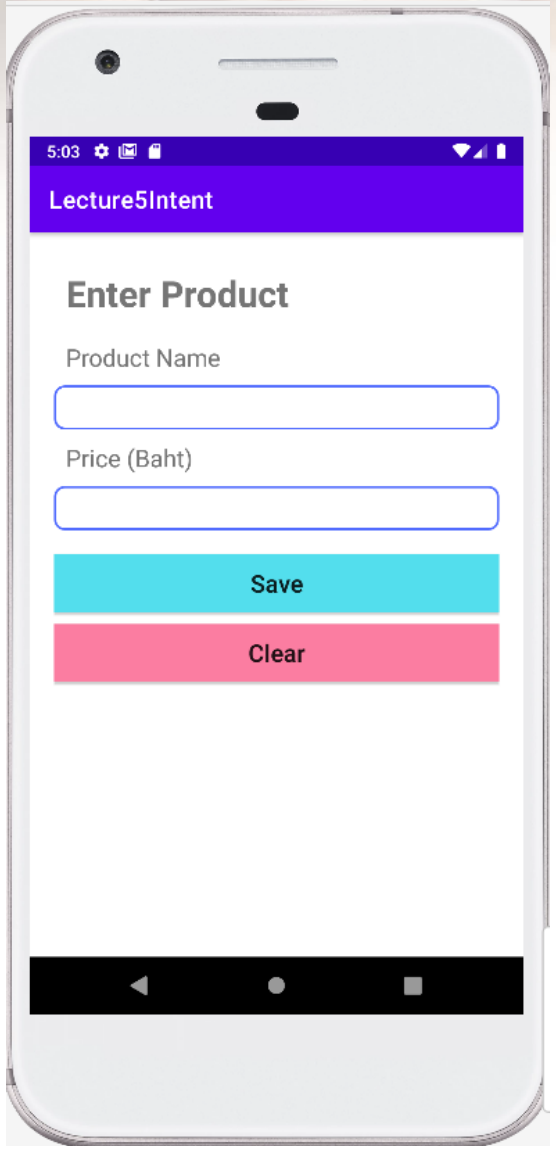


Intents with Parameter

- To pass data onto the new activities we use key value pairs inside the function putExtra, putStringArrayListExtra etc.
- putExtra generally passes the basic types such as Int, Float, Char, Double, Boolean, String along with

```
val i = Intent(this, OtherActivity::class.java)
i.putExtra("keyString", "Androidly String data")
startActivity(i)
```

Intents with Parameter



5:03

Lecture5Intent

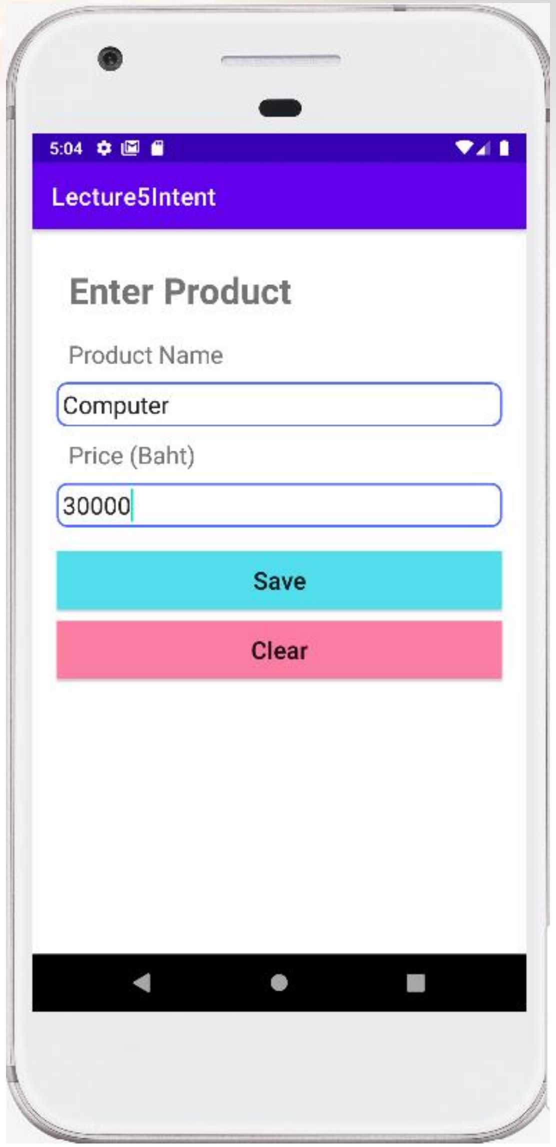
Enter Product

Product Name

Price (Baht)

Save

Clear



5:04

Lecture5Intent

Enter Product

Product Name

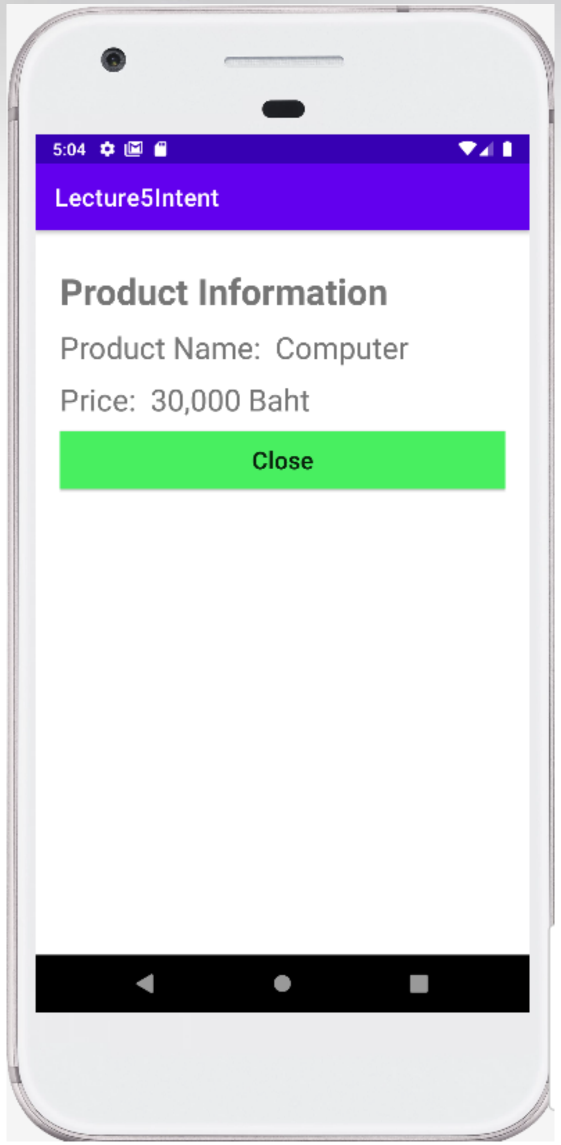
Computer

Price (Baht)

30000

Save

Clear



5:04

Lecture5Intent

Product Information

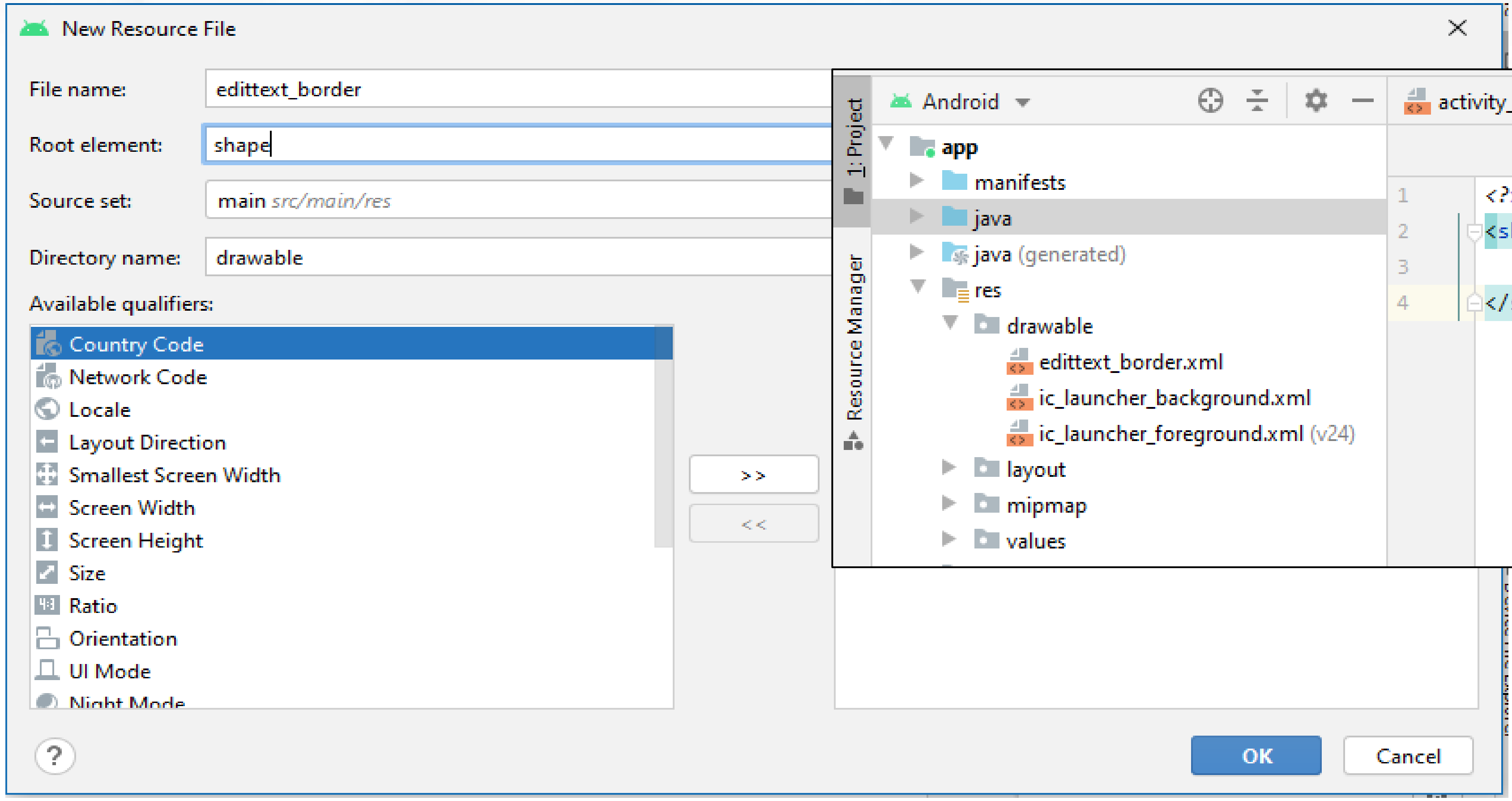
Product Name: Computer

Price: 30,000 Baht

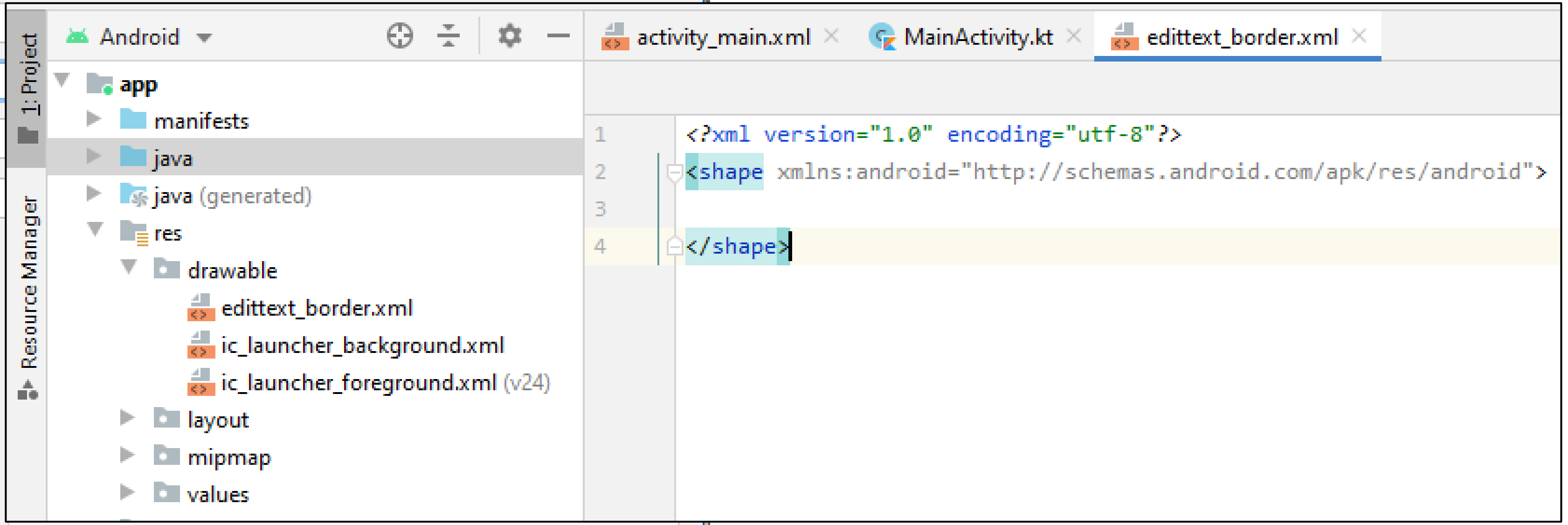
Close

EditText Border

- drawable folder> New > Drawable Resource File



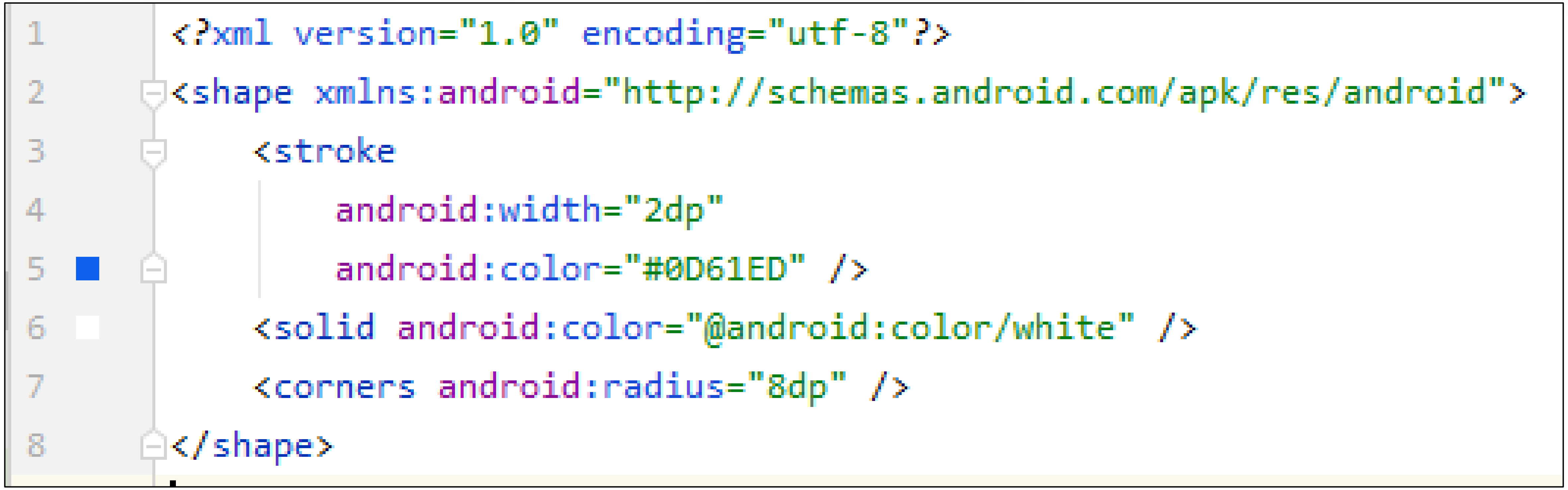
File name: edittext_border
Root element: shape
Source set: main src/main/res
Directory name: drawable



```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">

</shape>
```

EditText Border

- res/drawable



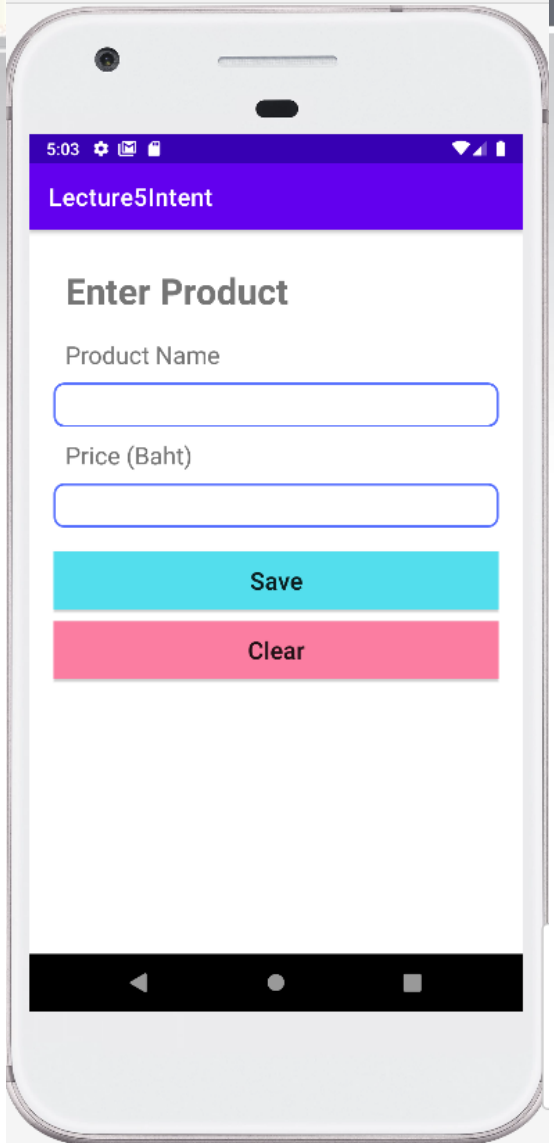
```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
    <stroke
        android:width="2dp"
        android:color="#0D61ED" />
    <solid android:color="@android:color/white" />
    <corners android:radius="8dp" />
</shape>
```

Intent with Parameter

```
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="20dp"
    android:orientation="vertical">
     <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Enter Product"
        android:textStyle="bold"
        android:textSize="30sp"
        android:padding="10dp"/>
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Product Name "
        android:textSize="20sp"
        android:padding="10dp"/>
```

activity_main.xml

```
<androidx.appcompat.widget.AppCompatEditText
        android:id="@+id/edtName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="text"
        android:textSize="20sp"
        android:padding="5dp"
        android:background="@drawable/edittext_border"/>
```

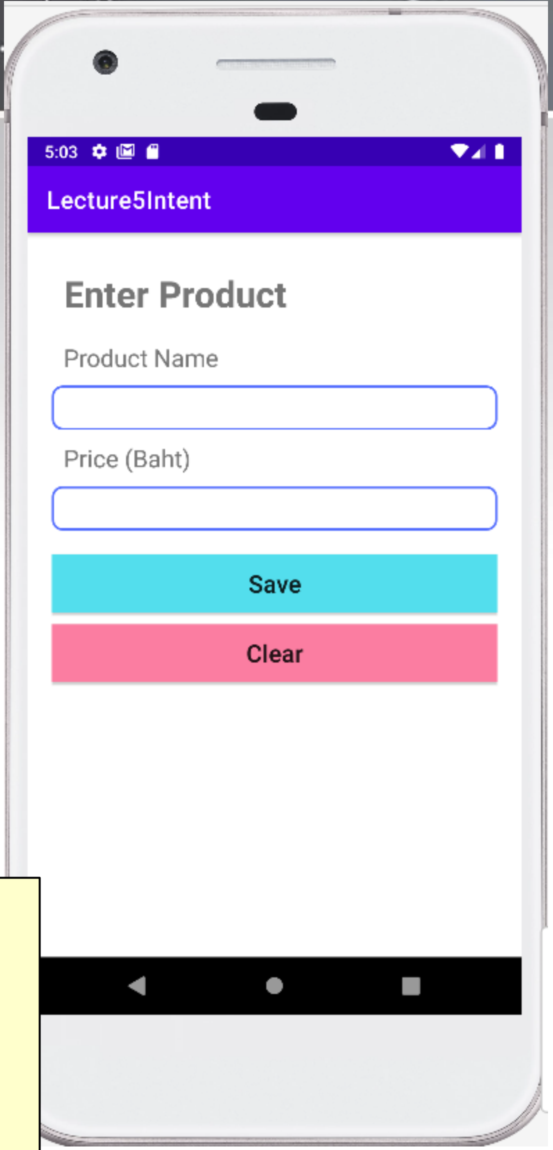


Intent with Parameter

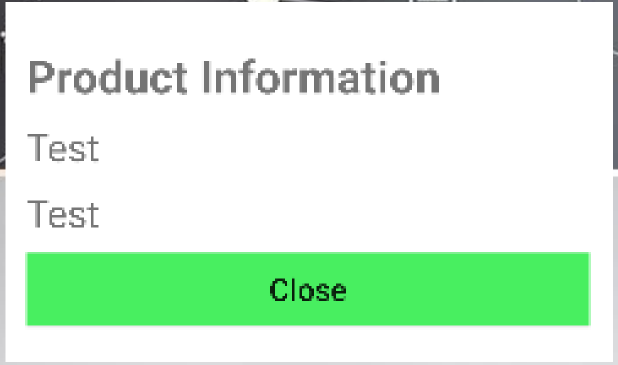
```
<TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Price (Baht) "
      android:textSize="20sp"
      android:padding="10dp"/>
  <androidx.appcompat.widget.AppCompatEditText
      android:id="@+id/edtPrice"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:inputType="numberDecimal"
      android:textSize="20sp"
      android:padding="5dp"
      android:background="@drawable/edittext_border"/>
<androidx.appcompat.widget.AppCompatButton
      android:id="@+id/btnSave"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_marginTop="20dp"
      android:background="#53deed"
      android:text="Save"
      android:textSize="20sp"
      android:textAllCaps="false" />
```

activity_main.xml (cont)

```
<androidx.appcompat.widget.AppCompatButton
      android:id="@+id/btnClear"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:layout_marginTop="10dp"
      android:background="#FB7DA1"
      android:text="Clear"
      android:textSize="20sp"
      android:textAllCaps="false" />
</LinearLayout>
```



Intent with Parameter


Product Information

Test

Test

activity_second.xml

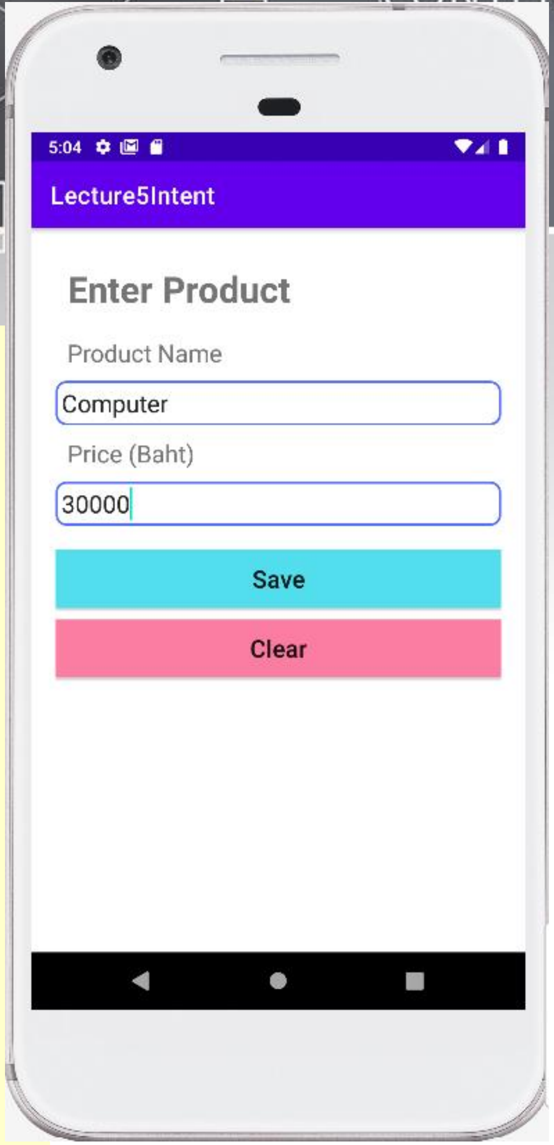
```
<?xml version="1.0"  encoding="utf-8"?>
<LinearLayout  …..
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        android:padding="20dp"
        tools:context=".ShowActivity">
<TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:textSize="30sp"
        android:textStyle="bold"
        android:text="Product Information"/>
<TextView
        android:id="@+id/txtName"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:textSize="25sp"
        android:text="Test"/>
```

```
<TextView
        android:id="@+id/txtPrice"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:textSize="25sp"
        android:text="Test"/>
<androidx.appcompat.widget.AppCompatButton
        android:id="@+id/btnClose"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:background="#48ef60"
        android:text="Close"
        android:textAllCaps="false"
        android:textSize="20sp"/>
</LinearLayout>
```

Intent with Parameter

MainActivity.kt

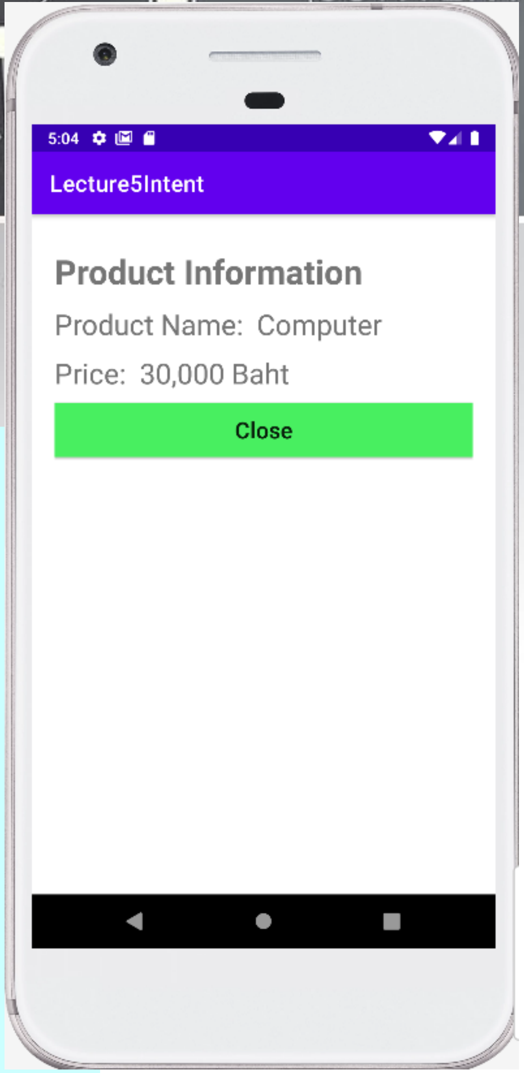
```
class MainActivity : AppCompatActivity() {
        private lateinit var  binding : ActivityMainBinding
        override fun onCreate(savedInstanceState: Bundle?) {
            super.onCreate(savedInstanceState)

            binding = ActivityMainBinding.inflate(layoutInflater)
            setContentView(binding.root)

            binding.btnSave.setOnClickListener {
                val mName = binding.edtName.text.toString()
                val mPrice  = binding.edtPrice.text.toString().toInt()
                val i = Intent(applicationContext, SecondActivity::class.java)
                 i.putExtra("pName", mName )
                 i.putExtra("pPrice", mPrice)
                 startActivity(i)
              }
            binding.btnClear.setOnClickListener {
                binding.edtName.text?.clear()
                binding.edtPrice.text?.clear()
            }
}}
```


5:04
Lecture5Intent
Enter Product
Product Name
Computer
Price (Baht)
30000

Intent with Parameter

SecondActivity.kt

```
class ShowActivity : AppCompatActivity() {

    private lateinit var  bindingSecond : ActivitySecondBinding
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        bindingShow = ActivitySecondBinding.inflate(layoutInflater)
        setContentView(bindingSecond.root)

        val mName = intent.getStringExtra("pName")
        val mPrice = intent.getIntExtra("pPrice",0)
        bindingSecond.txtName.text = "Product Name:  $mName"
        bindingSecond.txtPrice.text = "Price:  ${String.format("%,d",  mPrice)} Baht"

        bindingSecond.btnClose.setOnClickListener() {
            finish()
        }
} }
```


5:04

Lecture5Intent

Product Information

Product Name: Computer

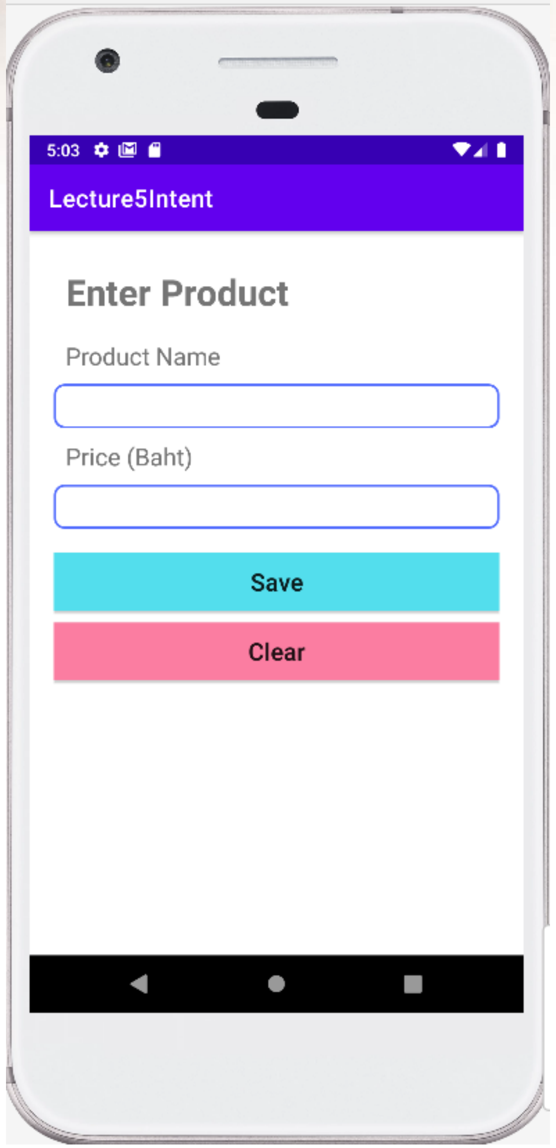
Price: 30,000 Baht

Close

Intents with Object: Parcelable

- Parcelable is interface for passing data as an object between android application components.
- Parcelable support : Android Extensions plugin now includes an automatic Parcelable implementation generator.

Intents with Object: Parcelable


5:03

Lecture5Intent

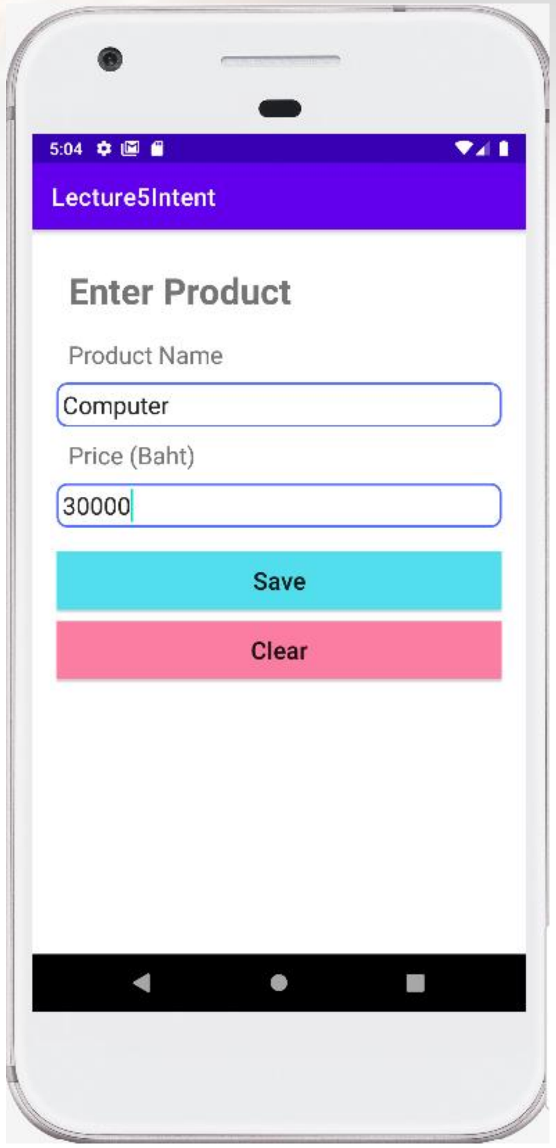
Enter Product

Product Name

Price (Baht)

Save

Clear


5:04

Lecture5Intent

Enter Product

Product Name

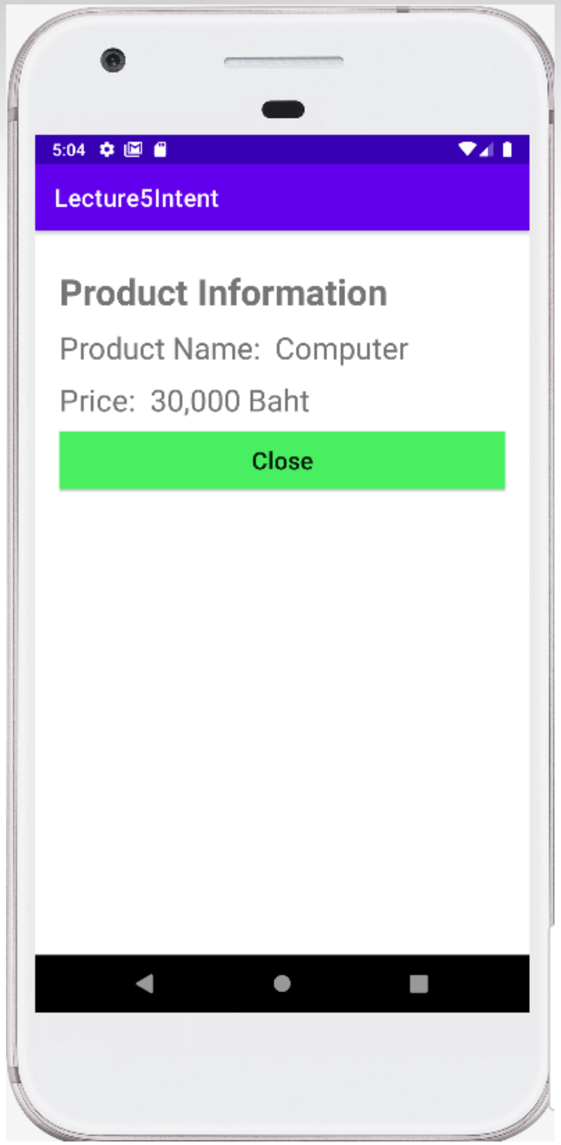
Computer

Price (Baht)

30000

Save

Clear


5:04

Lecture5Intent

Product Information

Product Name: Computer

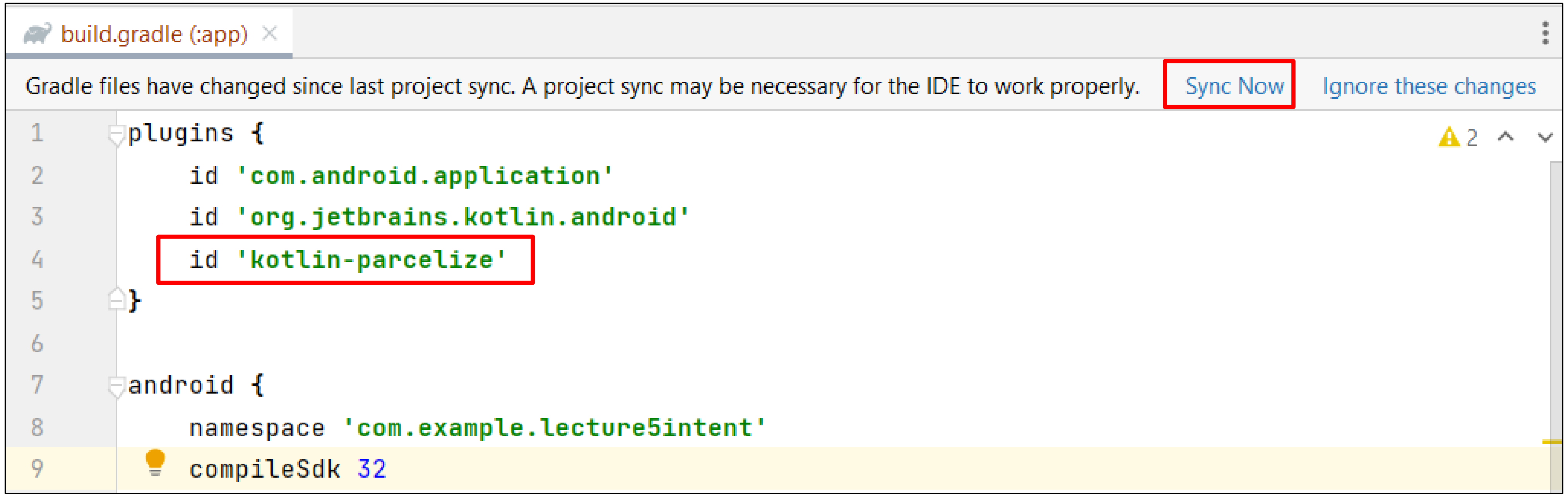
Price: 30,000 Baht

Close

Intents with Object : Parcelable

- Add to plugin to build.gradle File and Sync

id 'kotlin-parcelize'

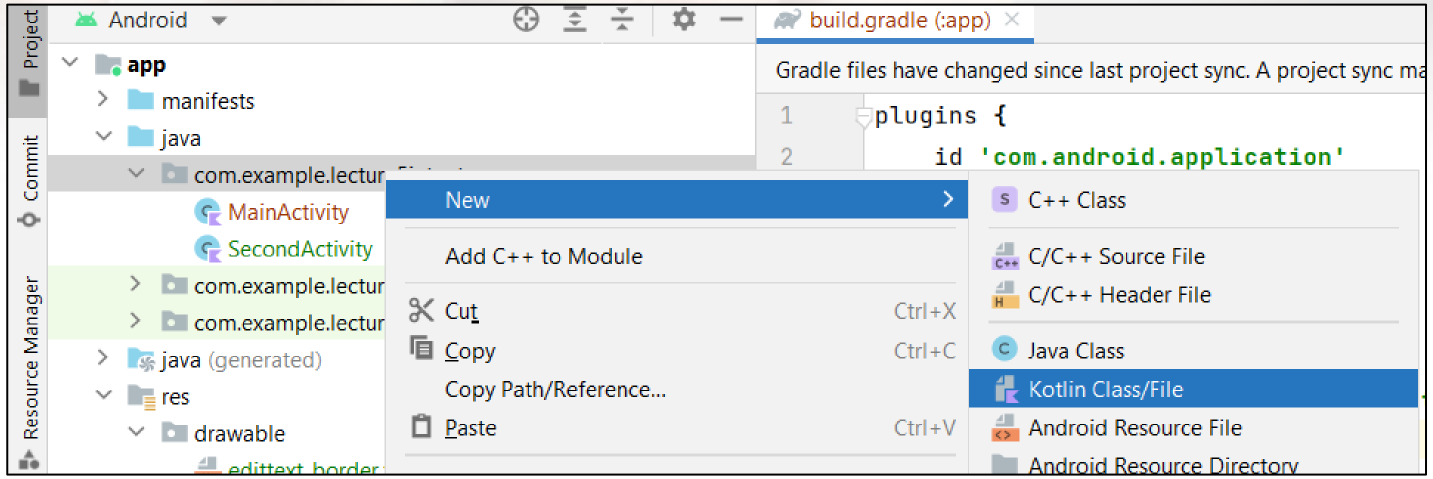


Gradle files have changed since last project sync. A project sync may be necessary for the IDE to work properly. Sync Now Ignore these changes

```
plugins {
    id 'com.android.application'
    id 'org.jetbrains.kotlin.android'
    id 'kotlin-parcelize'
}

android {
    namespace 'com.example.lecture5intent'
    compileSdk 32
```

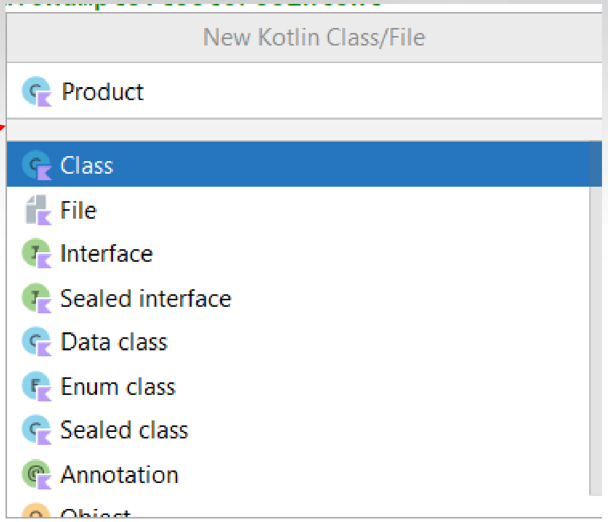
Intents with Object : Parcelable

- Create Data Class : Product

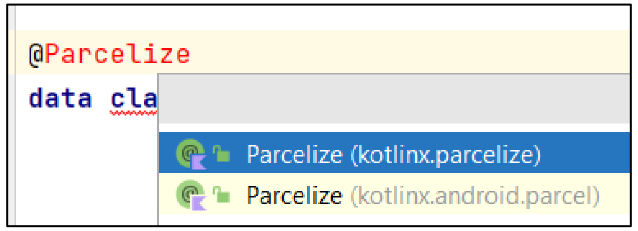


Gradle files have changed since last project sync. A project sync ma

```
plugins {
    id 'com.android.application'
```

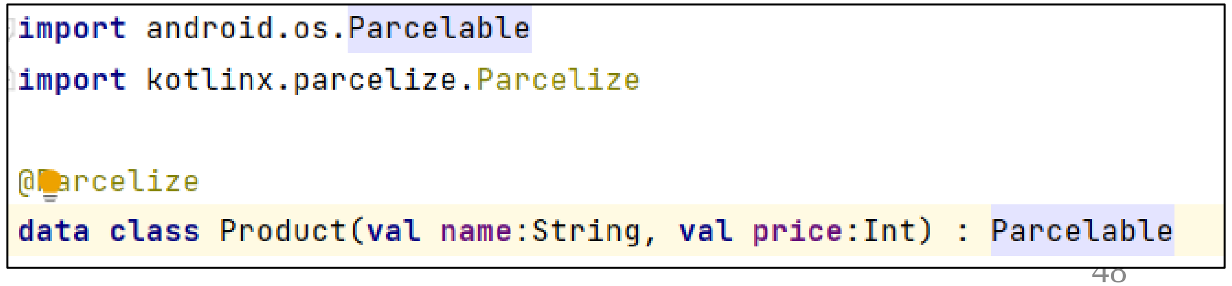


@Parcelize
data class Product(val name: String, val price: Int) : Parcelable


Parcelize (kotlinx.parcelize)
Parcelize (kotlinx.android.parcel)

```
@Parcelize
data cla
```

'kotlinx-parcelize'

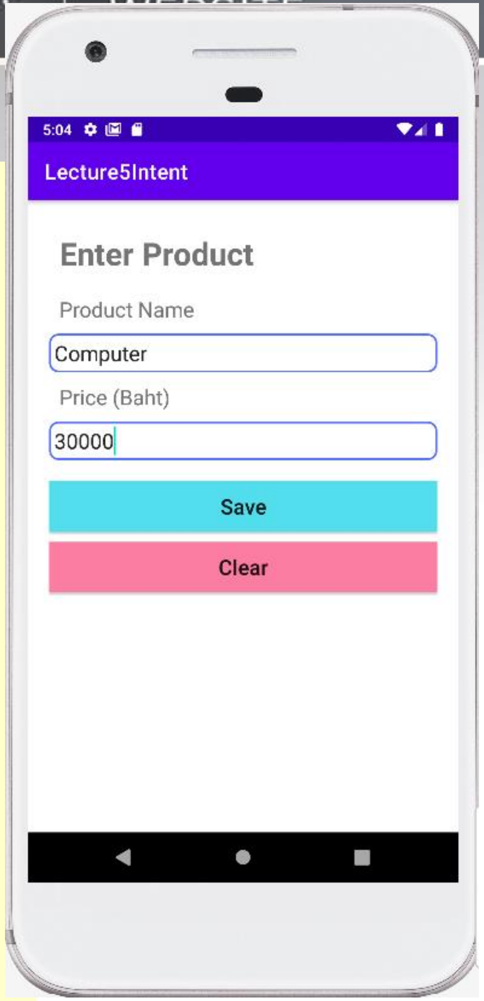


```
import android.os.Parcelable
import kotlinx.parcelize.Parcelize

@Parcelize
data class Product(val name:String, val price:Int) : Parcelable
```

Intents with Parcelable

MainActivity.kt

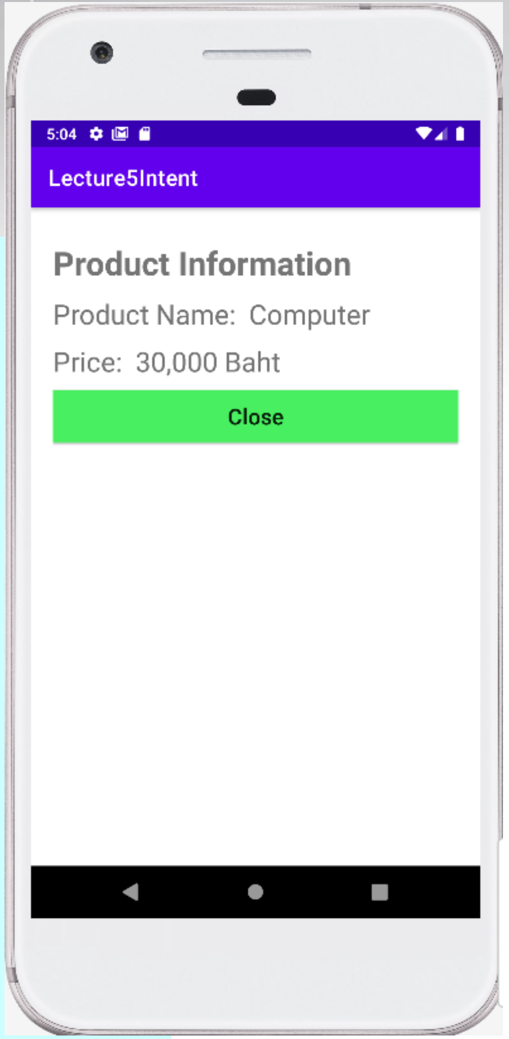
```
class MainActivity : AppCompatActivity() {

    private lateinit var  binding : ActivityMainBinding
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        binding = ActivityMainBinding.inflate(layoutInflater)
        setContentView(binding.root)

        binding.btnSave.setOnClickListener {
        val mName = binding.edtName.text.toString()
        val mPrice = binding.edtPrice.text.toString().toInt()
        val i = Intent(applicationContext, SecondActivity::class.java)
            i.putExtra("productData", Product(mName, mPrice))
         startActivity(i)
    }
         binding.btnClear.setOnClickListener {
             binding.edtName.text?.clear()
             binding.edtPrice.text?.clear()
    }
}}
```



Intents with Parcelable

ShowActivity.kt

```
class ShowActivity : AppCompatActivity() {
 private lateinit var  bindingSecond : ActivitySecondBinding

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)

        bindingSecond = ActivitySecondBinding.inflate(layoutInflater)
        setContentView(bindingSecond.root)

        var data = intent.extras
        var newProduct = data?.getParcelable<Product>("productData")
        bindingSecond.txtName.text = "Product Name:  ${newProduct?.name}"
        bindingSecond.txtPrice.text = "Price:  ${String.format("%,d",newProduct?.price)} Baht"

        bindingSecond.btnClose.setOnClickListener() {
            finish()
        }
    }}
```


5:04

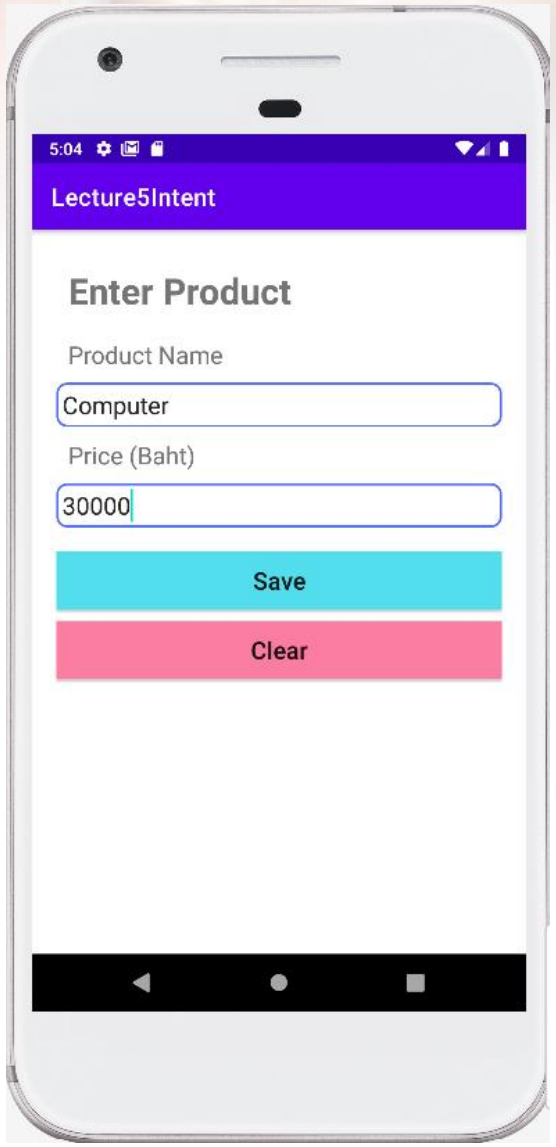
Lecture5Intent

Product Information

Product Name: Computer

Price: 30,000 Baht

Intents with Parcelable


5:04

Lecture5Intent

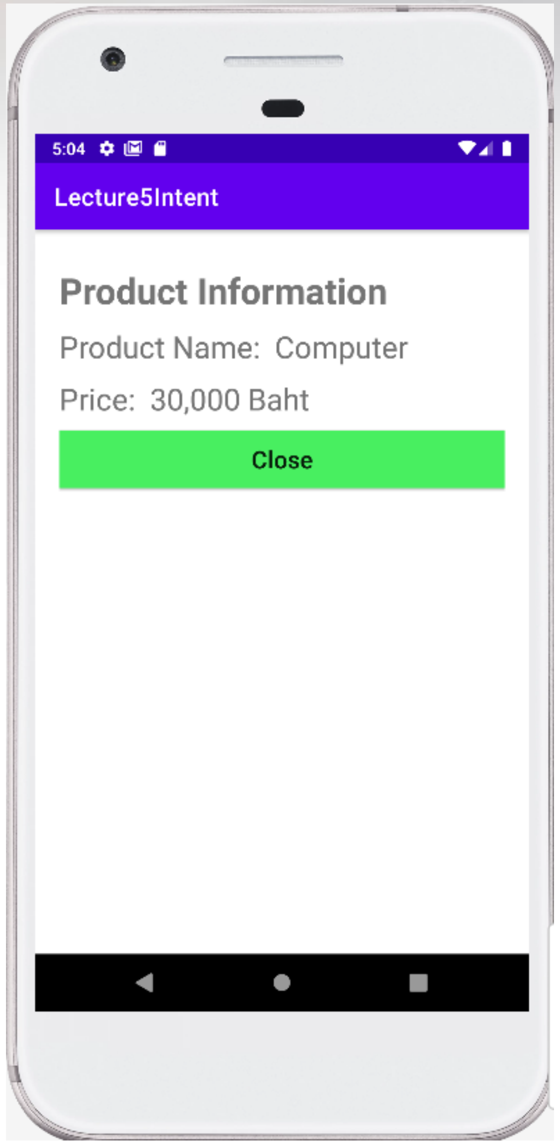
Enter Product

Product Name

Computer

Price (Baht)

30000

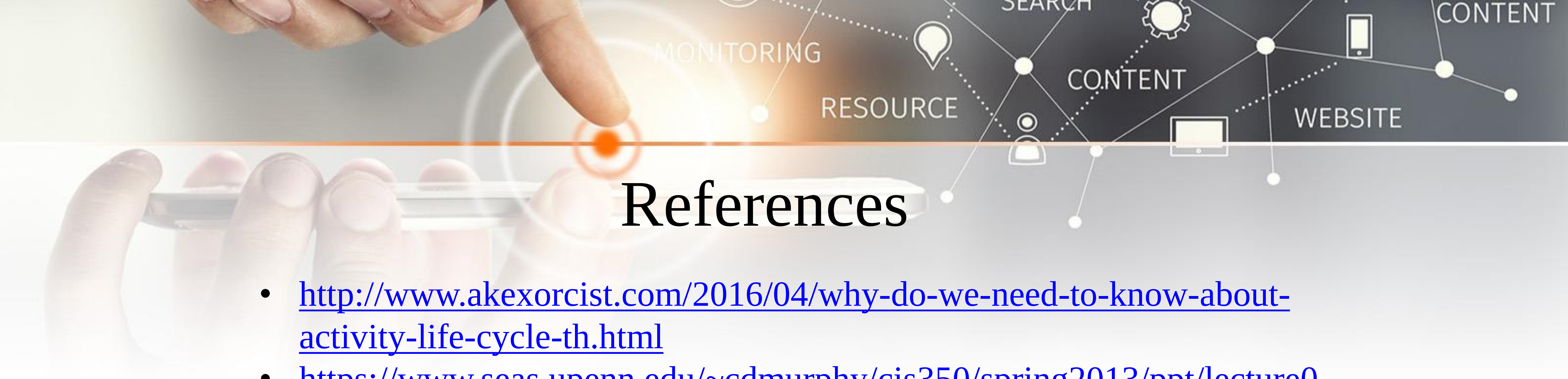

5:04

Lecture5Intent

Product Information

Product Name: Computer

Price: 30,000 Baht



References

- http://www.akexorcist.com/2016/04/why-do-we-need-to-know-about-activity-life-cycle-th.html
- https://www.seas.upenn.edu/~cdmurphy/cis350/spring2013/ppt/lecture06-android.ppt
- https://developer.android.com/guide/components/activities/activity-lifecycle
- http://www.artit-k.com/android-activity-lifecycle/
- http://www.cs.binghamton.edu/~steflik/cs328/AndroidLifeCycle.ppt
- http://ksuweb.kennesaw.edu/~kqian/cs4322/Intent-ppt.pptx
- https://tutorial.eyehunts.com/android/parcelable-android-pass-data-between-activities-kotlin-parcelize/