Android Applications

Dominic Duggan
Stevens Institute of Technology

1

ANDROID APPLICATIONS

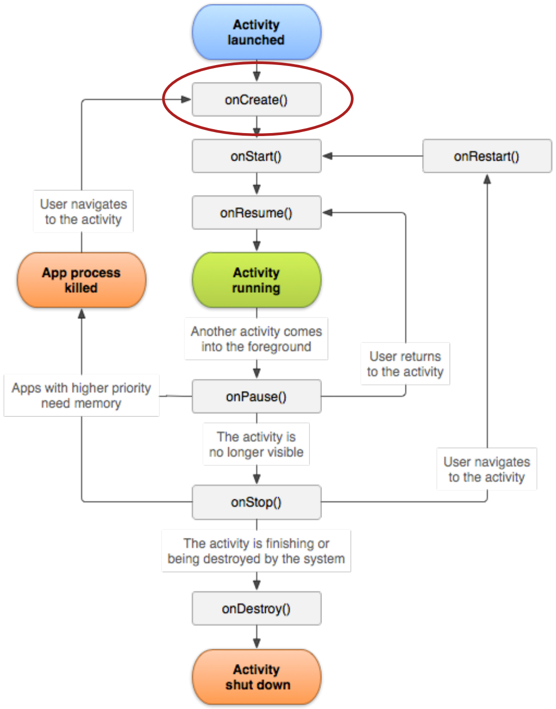
2

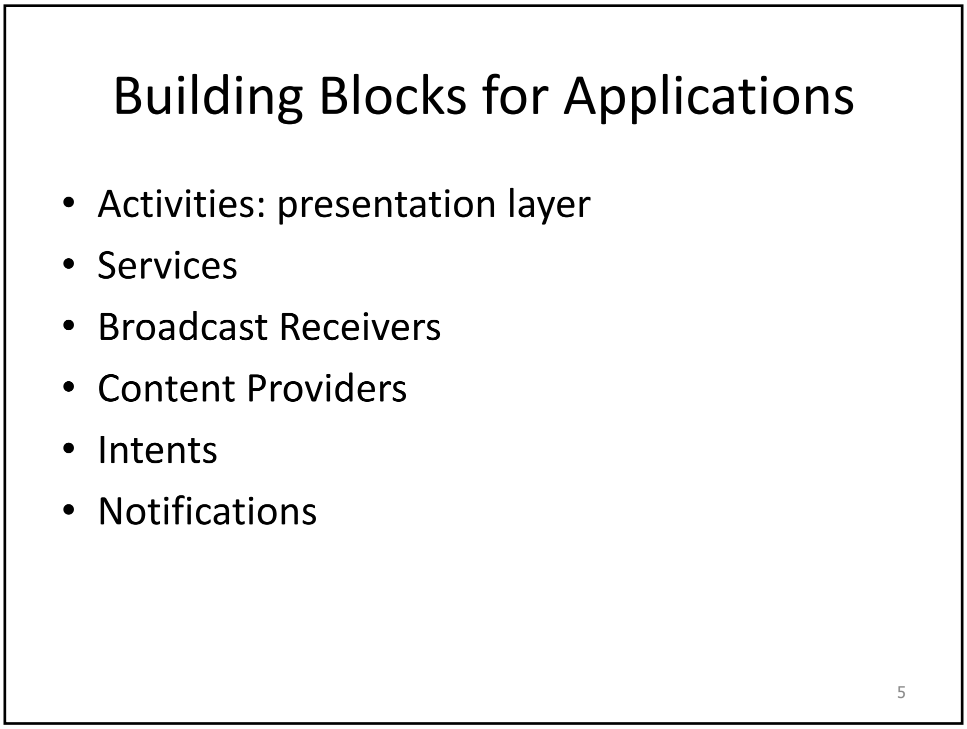
2

Android Basics

- (Mostly) Three Categories of Applications
  - Foreground activities
    - Suspended when not visible
    - E.g. games, mashups
  - Background services
    - E.g. call screening, SMS auto-responders
  - Intermittent activity
    - E.g. media player

Activity Life Cycle


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

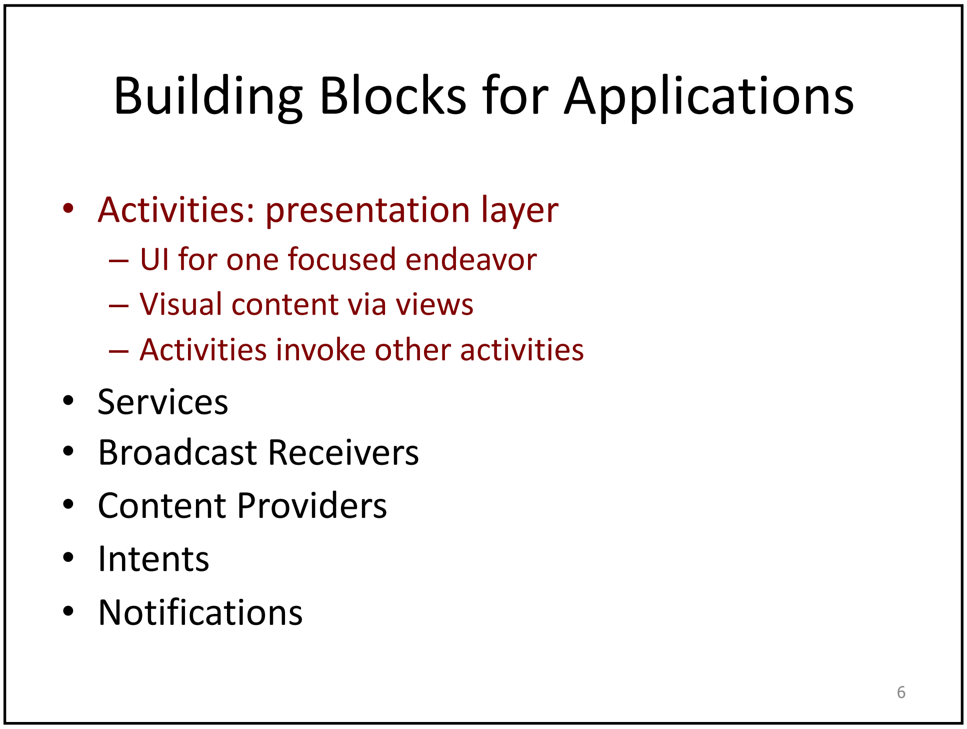


Building Blocks for Applications

- Activities: presentation layer
- Services
- Broadcast Receivers
- Content Providers
- Intents
- Notifications

5

5

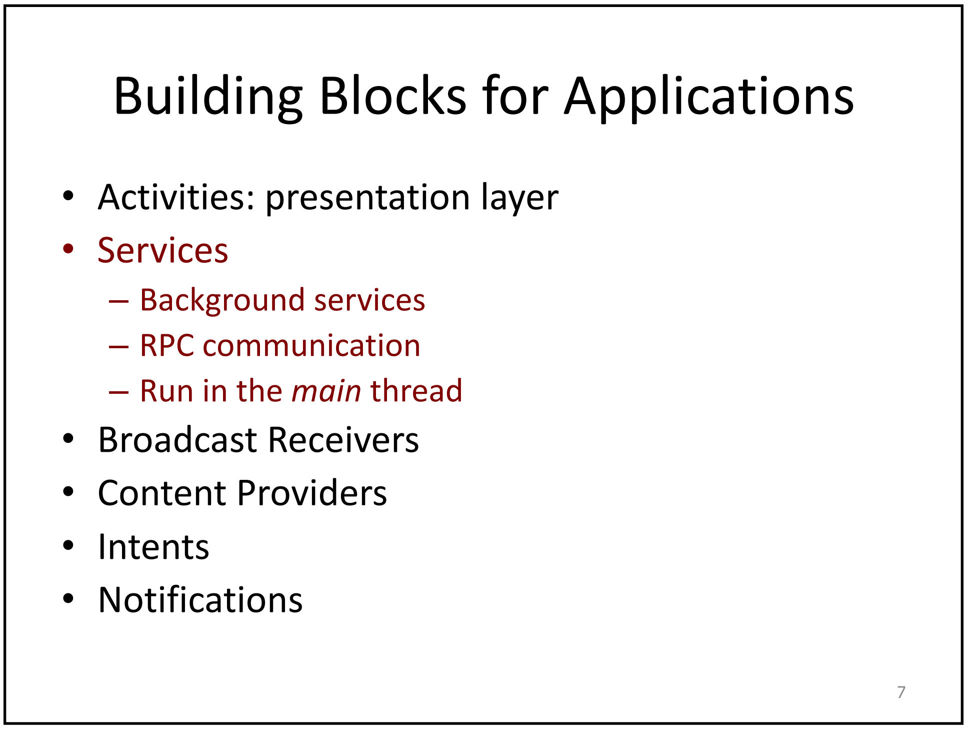


Building Blocks for Applications

- Activities: presentation layer
  - UI for one focused endeavor
  - Visual content via views
  - Activities invoke other activities
- Services
- Broadcast Receivers
- Content Providers
- Intents
- Notifications

6

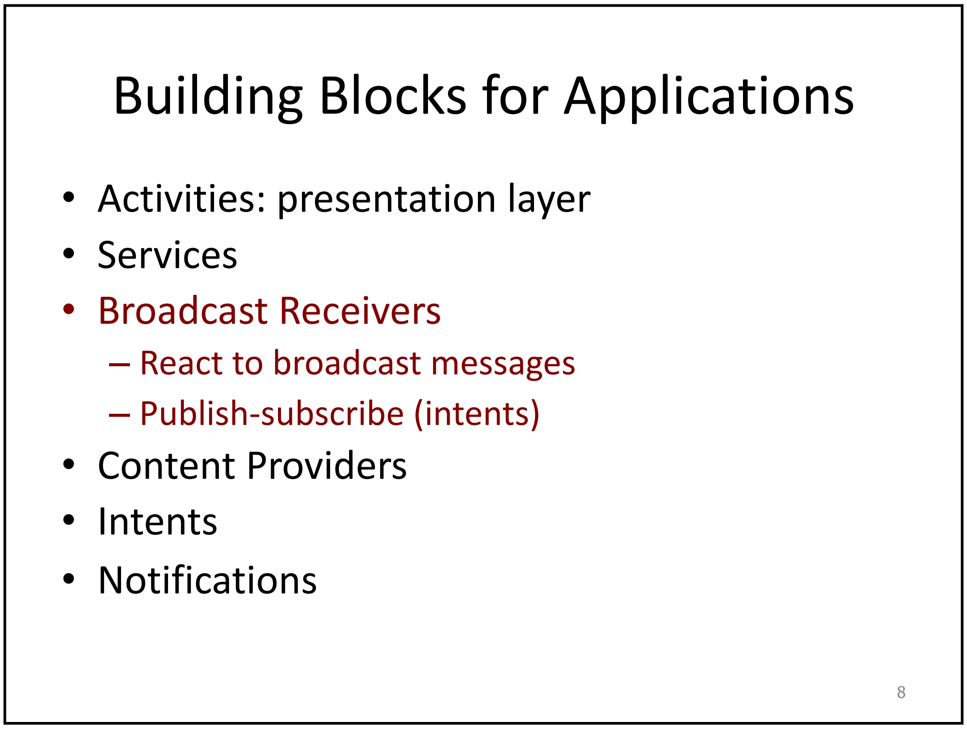
6



Building Blocks for Applications

- Activities: presentation layer
- Services
  - Background services
  - RPC communication
  - Run in the *main* thread
- Broadcast Receivers
- Content Providers
- Intents
- Notifications

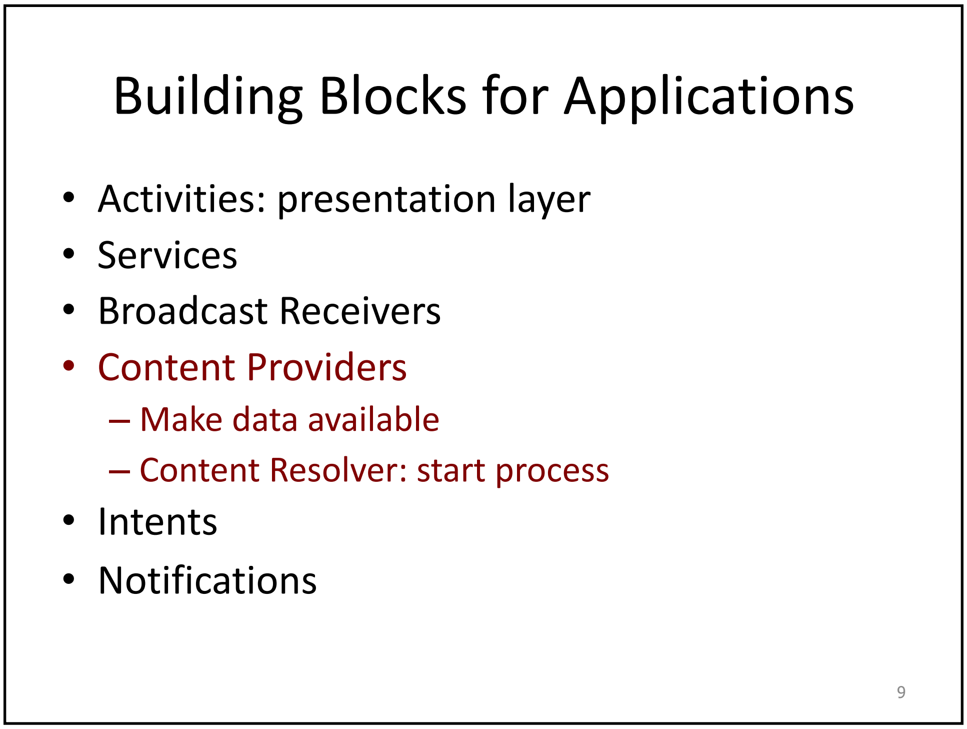
7



Building Blocks for Applications

- Activities: presentation layer
- Services
- Broadcast Receivers
  - React to broadcast messages
  - Publish-subscribe (intents)
- Content Providers
- Intents
- Notifications

8

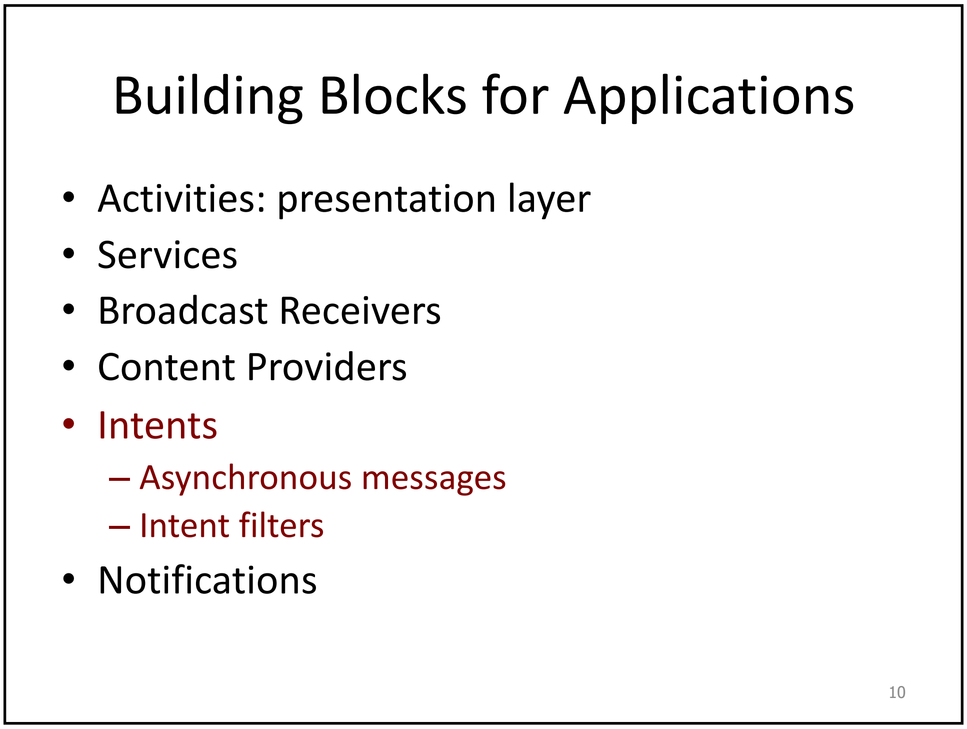


Building Blocks for Applications

- Activities: presentation layer
- Services
- Broadcast Receivers
- Content Providers
  - Make data available
  - Content Resolver: start process
- Intents
- Notifications

9

9

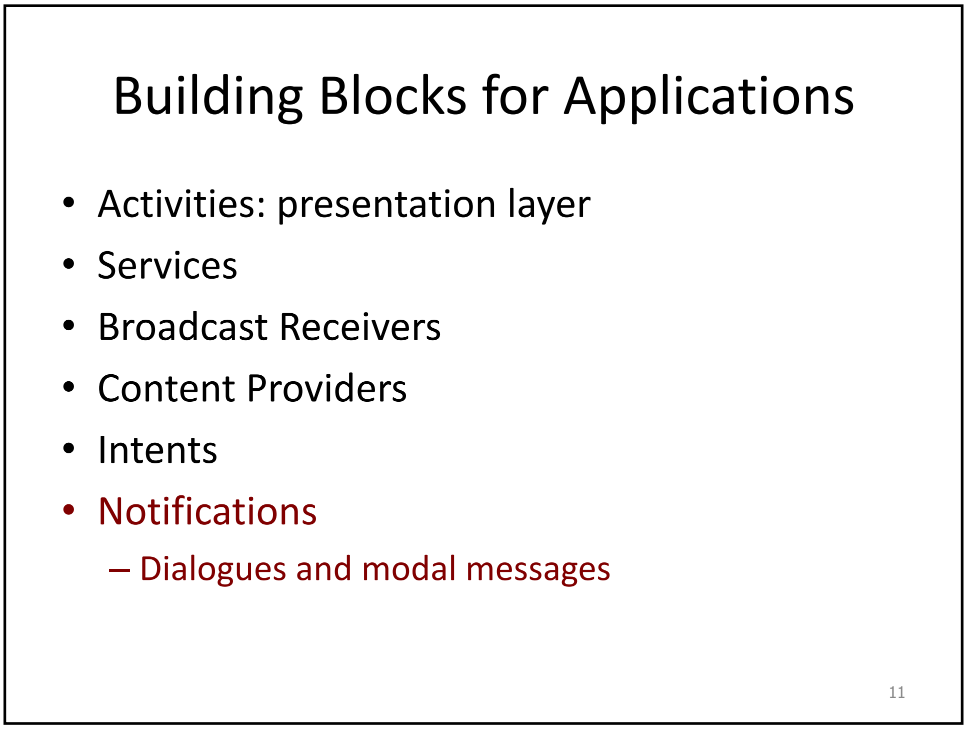


Building Blocks for Applications

- Activities: presentation layer
- Services
- Broadcast Receivers
- Content Providers
- Intents
  - Asynchronous messages
  - Intent filters
- Notifications

10

10



Building Blocks for Applications

- Activities: presentation layer
- Services
- Broadcast Receivers
- Content Providers
- Intents
- Notifications
  - Dialogues and modal messages

11

APPLICATION MANIFEST

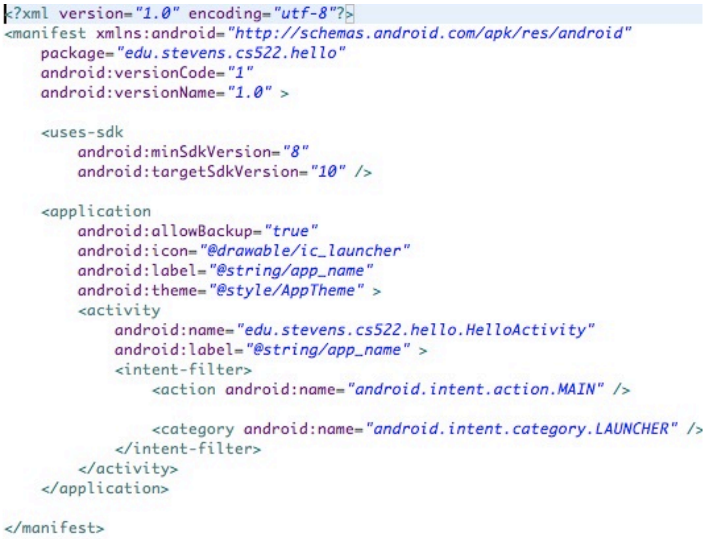
12

Application Manifest

- Stored in root of project hierarchy
  - AndroidManifest.xml
  - Define structure and metadata of application
  - Nodes for each of the components

```
<manifest
  xmlns:android=“http://schemas.android.com/apk/res/android”
  package=”edu.stevens.cs522.hello">
…
</manifest>
```

Application Manifest



```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="edu.stevens.cs522.hello"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="10" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="edu.stevens.cs522.hello.HelloActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

</manifest>
```

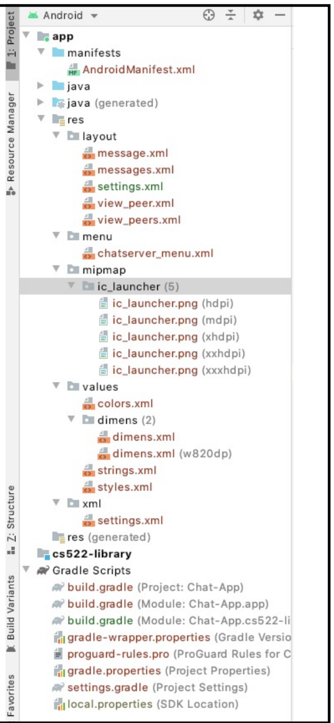
Application Manifest

- Manifest node tags
  - Application: container for…
    - Activity: specify intent filter
    - Service
    - Provider
    - Receiver: global broadcast receiver
  - Uses-permission:
    - Must be granted during installation
  - Permission
    - Declare to restrict access to components in app
  - instrumentation

15

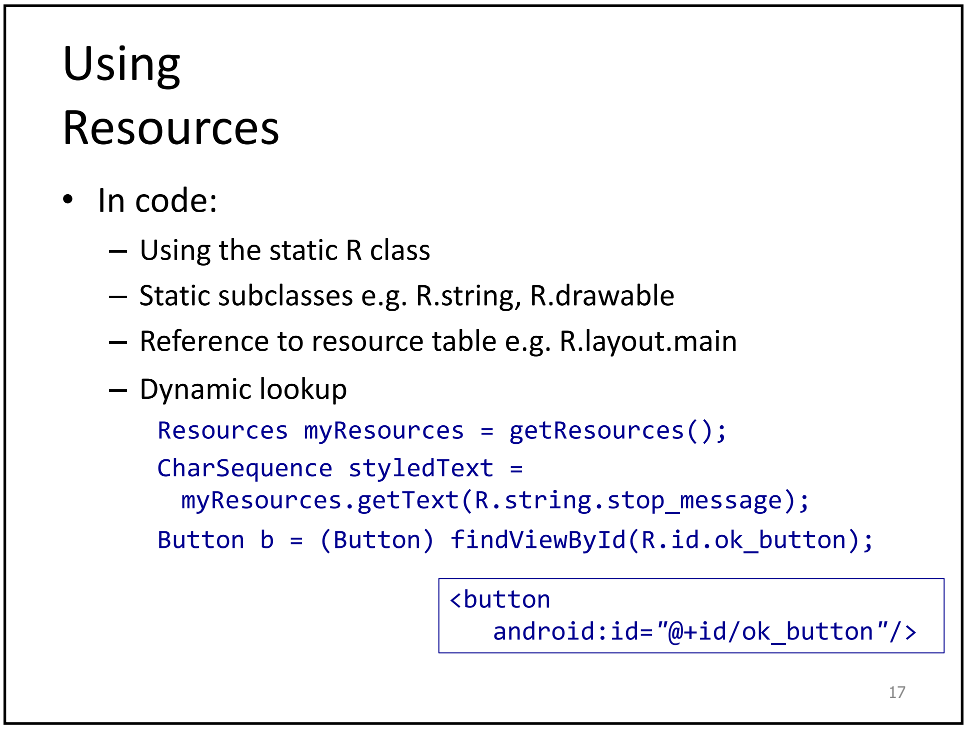
Resources

- External resources
  - Values
    - Strings, colors, dimensions, string or integer arrays
  - Styles and themes
    - Colors and fonts
  - Drawables
    - Bitmaps and (stretchable) images
  - Layouts
    - UI specified statically in XML
    - Android best practice



```
app
  manifests
    AndroidManifest.xml
  java
  java (generated)
  res
    layout
      message.xml
      messages.xml
      settings.xml
      view_peer.xml
      view_peers.xml
    menu
      chatserver_menu.xml
    mipmap
      ic_launcher (5)
        ic_launcher.png (hdpi)
        ic_launcher.png (mdpi)
        ic_launcher.png (xhdpi)
        ic_launcher.png (xxhdpi)
        ic_launcher.png (xxxhdpi)
    values
      colors.xml
      dimens (2)
        dimens.xml
        dimens.xml (w820dp)
      strings.xml
      styles.xml
    xml
      settings.xml
  res (generated)
cs522-library
Gradle Scripts
  build.gradle (Project: Chat-App)
  build.gradle (Module: Chat-App.app)
  build.gradle (Module: Chat-App.cs522-li
  gradle-wrapper.properties (Gradle Versio
  proguard-rules.pro (ProGuard Rules for C
  gradle.properties (Project Properties)
  settings.gradle (Project Settings)
  local.properties (SDK Location)
```

16



Using Resources

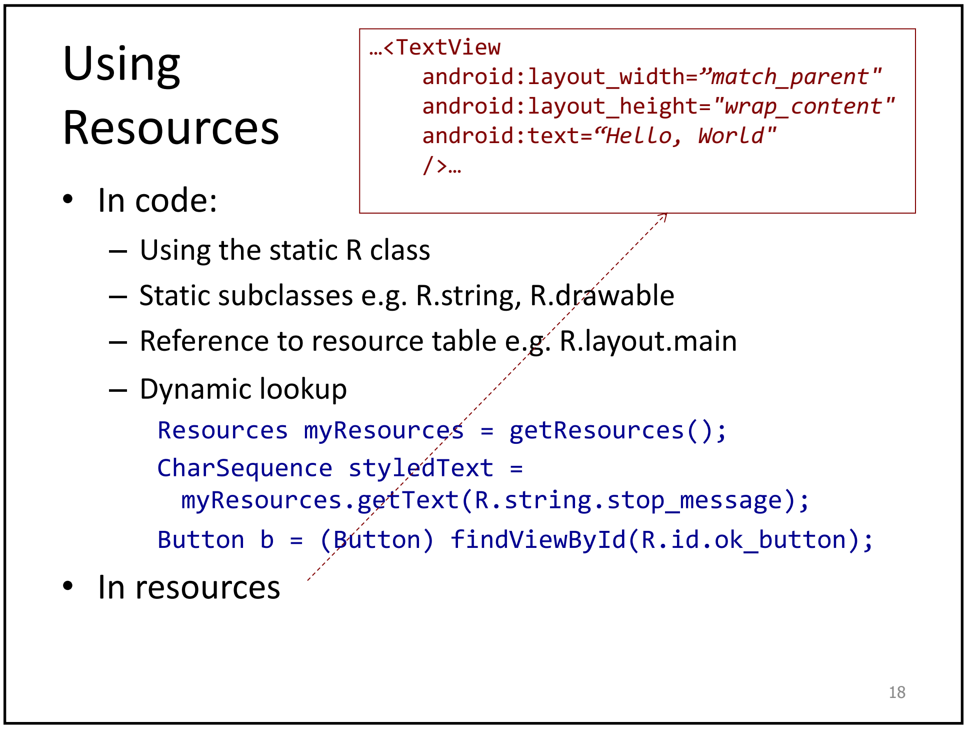
- In code:
  - Using the static R class
  - Static subclasses e.g. R.string, R.drawable
  - Reference to resource table e.g. R.layout.main
  - Dynamic lookup

```
Resources myResources = getResources();
CharSequence styledText =
  myResources.getText(R.string.stop_message);
Button b = (Button) findViewById(R.id.ok_button);
```

```
<button
   android:id="@+id/ok_button"/>
```

17

17



Using Resources

```
…<TextView
    android:layout_width=”match_parent"
    android:layout_height="wrap_content"
    android:text=“Hello, World"
    />…
```

- In code:
  - Using the static R class
  - Static subclasses e.g. R.string, R.drawable
  - Reference to resource table e.g. R.layout.main
  - Dynamic lookup

```
Resources myResources = getResources();
CharSequence styledText =
  myResources.getText(R.string.stop_message);
Button b = (Button) findViewById(R.id.ok_button);
```

- In resources

18

18

Using Resources

```
…<TextView
    android:layout_width=”match_parent"
    android:layout_height="wrap_content"
    android:text=“@string/hello_message”
    />…
```

- In code:
  - Using the static R class
  - Static subclasses e.g. R.string, R.drawable
  - Reference to resource table e.g. R.layout.main
  - Dynamic lookup

```
Resources myResources = getResources();
CharSequence styledText =
  myResources.getText(R.string.stop_message);
Button b = (Button) findViewById(R.id.ok_button);
```

- In resources

Activity Node in the Application Manifest

Entry point for a task

```
<activity android:label=“@string/app_name”
          android:name=“.MyActivity”>
  <intent-filter>
    <action
        android:name=“android.intent.action.MAIN”/>
    <category
      android:name=“android.intent.category.LAUNCHER”/>
  </intent-filter>
</activity>
```

Icon in the application launcher