



ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ  
HAROKOPIO UNIVERSITY



Τμήμα Πληροφορικής  
και Τηλεματικής

# Mobile Application Development

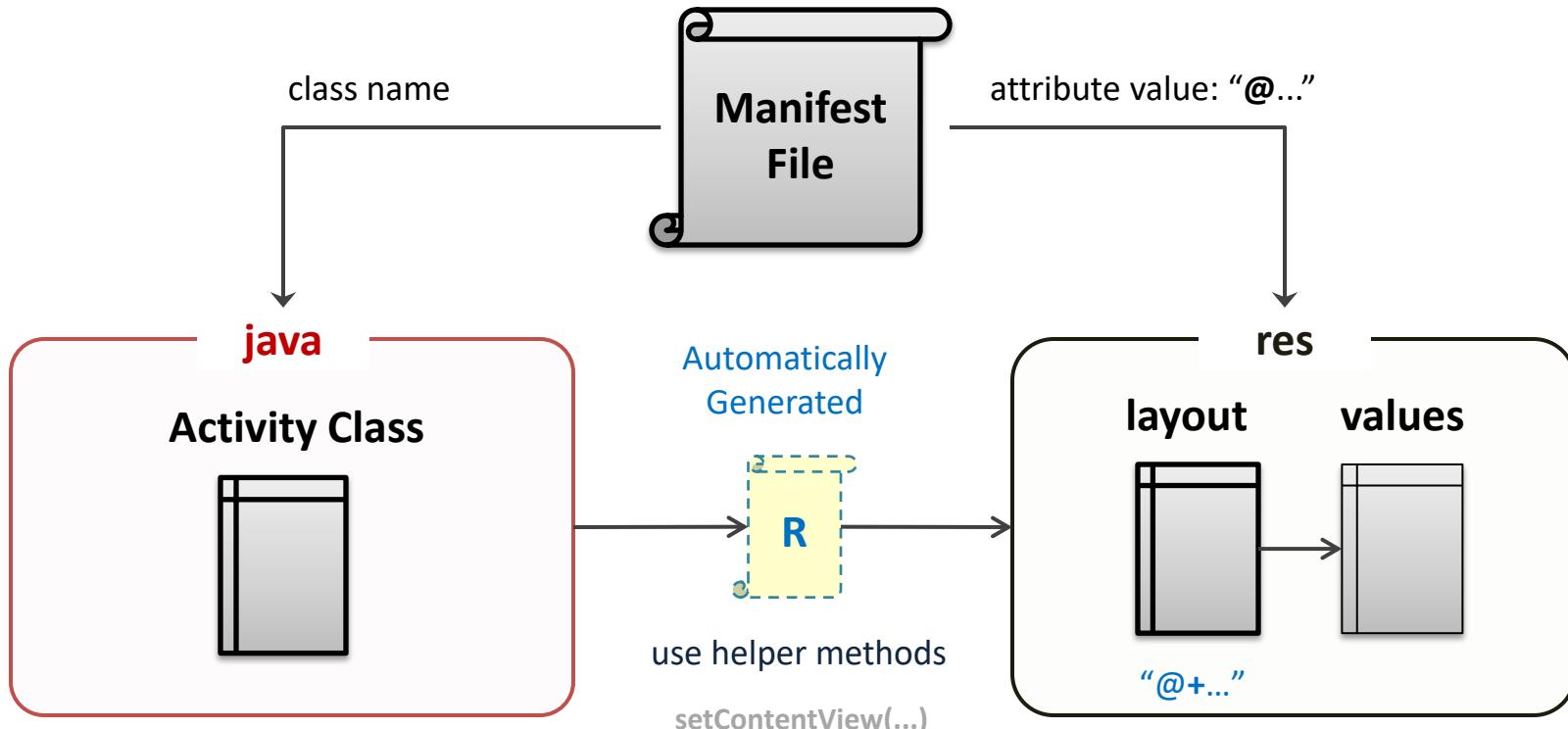
## Lab 1

Dr. Efthymios Chondrogiannis



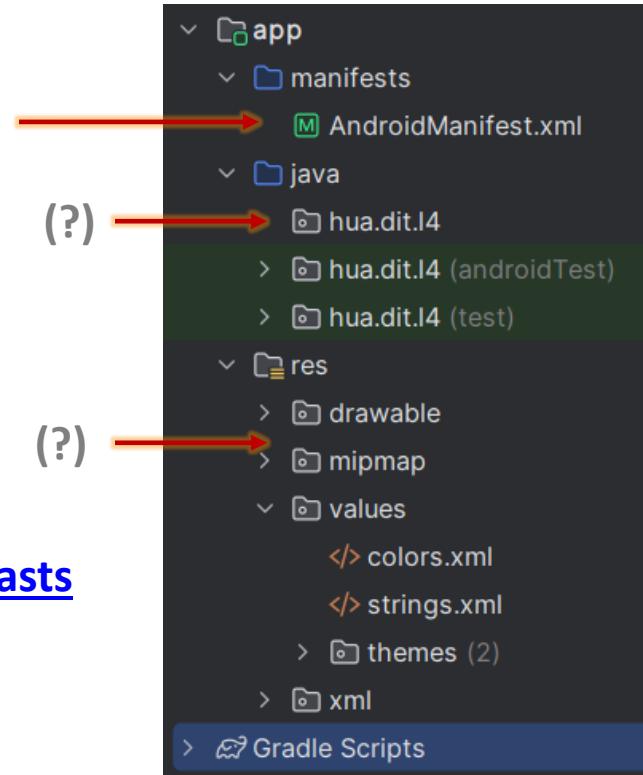
*Mobile Application Development*

# Android Project – Main Entities



# Android App Dev from Scratch

- New project – **No Activity**
- App UI – **Resource XML File(s)**
  - Layout Folder and XML File
- App Logic – **Java Code**
  - «Main» Activity – App Entry Point
    - Set UI ... Layout XML ... using R class (variable) (?)
  - Handle UI Events
    - Log Data and/or Show Messages using Toasts
- App Description – **Manifest XML**
  - Application Parameters & Main Activity



# Lab 1 Tasks

- **Task 1:** Study Activity Lifecycle (including what happens when configuration changes)
- **Task 2:** Create Activity Layout XML using different Views and View Groups
- **Task 3:** Link two (or more) Activities using Intents and Intent Filters

# Task 1: Study Activity Lifecycle

```
public class Activity extends ApplicationContext {  
    protected void onCreate(Bundle savedInstanceState);  
    protected void onStart();  
    protected void onRestart();  
    protected void onResume();  
    protected void onPause();  
    protected void onStop();  
    protected void onDestroy();  
}
```

For testing purposes:

- Override the six callback methods
- Add Log Messages
- Deploy & Run
- Check Log Messages

Activity: <https://developer.android.com/reference/android/app/Activity>

# Configuration Changes

- The system **recreates an Activity** when a configuration change occurs – also recreates the UI with the new configuration
  - e.g., Rotating the device, Changing the font size, or app language
- **Prevent automatic activity recreation**
  - Add the configuration type to the `<activity>` entry in your `AndroidManifest.xml`

```
<activity android:name=".MyActivity"
    android:configChanges="orientation|screenSize|screenLayout|keyboardHidden"
    android:label="@string/app_name">
```

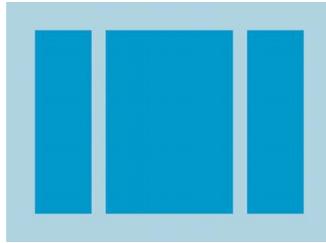
- Override the **onConfigurationChanged()** callback method in the Activity Class

```
public void onConfigurationChanged(Configuration newConfig)
```

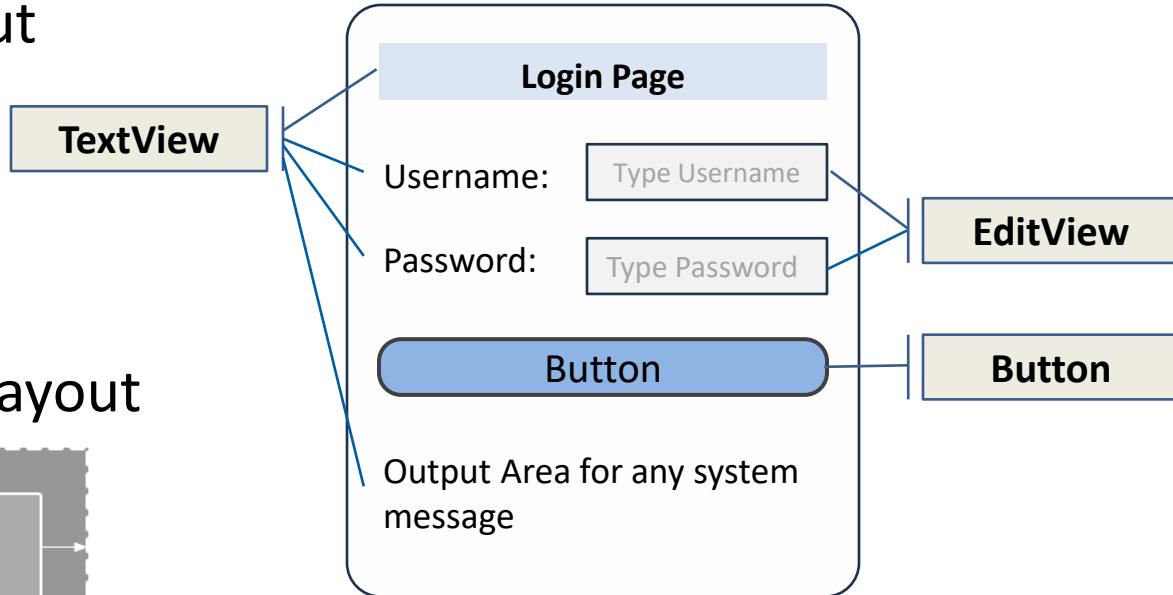
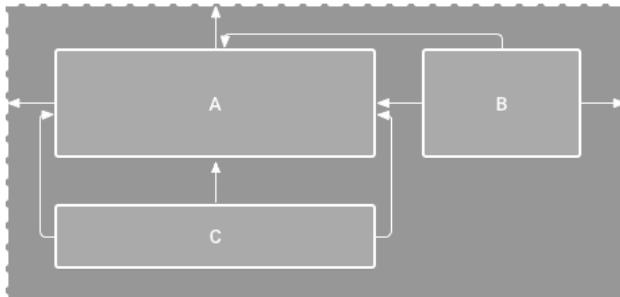
Handle configuration changes: <https://developer.android.com/guide/topics/resources/runtime-changes>

# Tasks 2: Create a Login Page

- A. Using Linear Layout



- B. Using Constraint Layout



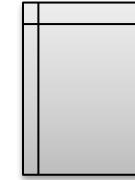
❖ On click ... when username and password are empty  
show an error message using **Toast**

# Style & Theme

- **Modify Theme**

- Remove App Bar
- Change Primary Color
- Increase Text Size

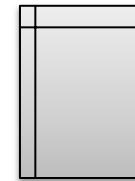
themes.xml



- **Specify Widget Style**

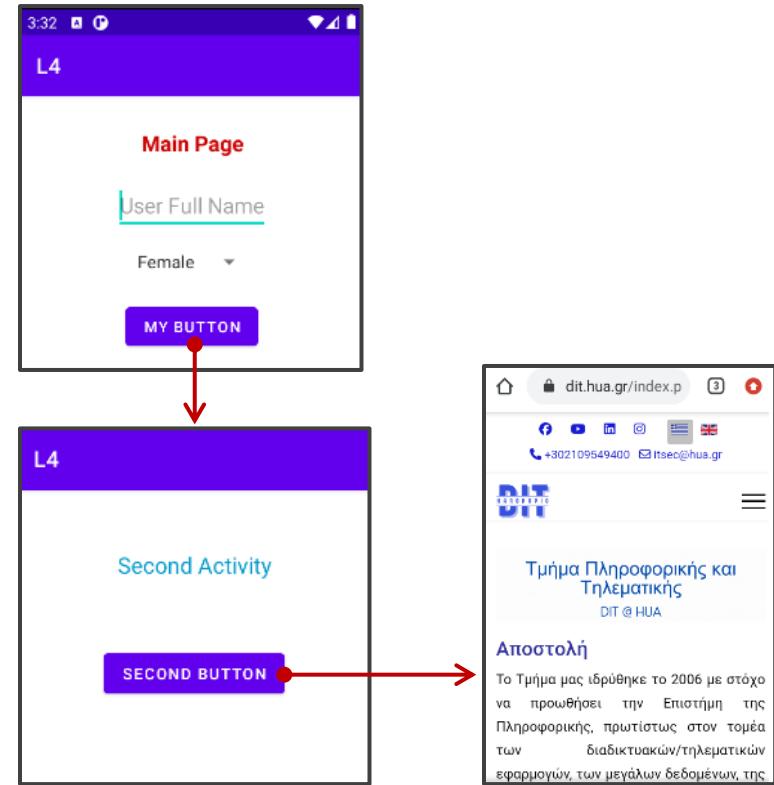
- Header Style → Text Size, Style and Color
- Edit Text Style
- Button Style

styles.xml



# Task 3: Navigation to (other) Activities

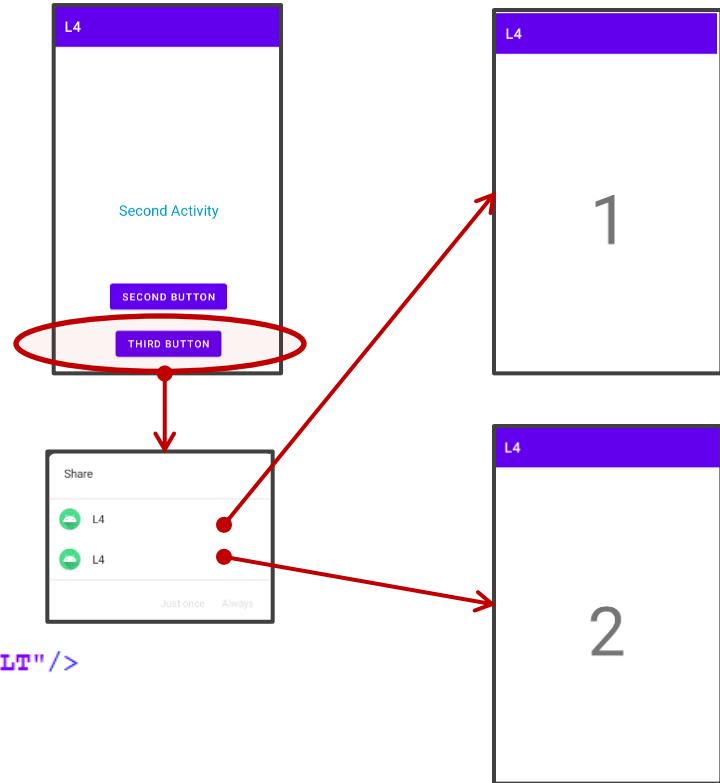
- Functionality
  - Go to another “page” using an **Explicit Intent**
    - Send **data** captured through the main App UI
  - Open a Web Page using an **Implicit Intent**
    - Enable users to open a **Web Browser** with the given **URL** by pressing a button



# Intents and Intent Filters

- Enable two Activities to handle the same Intent
  - Create an **Implicit** Intent when pressing a Button
  - Create two Activities with the same **Intent Filters**

```
<activity android:name=".Activity" android:exported="true">
    <intent-filter>
        <action android:name="android.intent.action.SEND"/>
        <category android:name="android.intent.category.DEFAULT"/>
    </intent-filter>
</activity>
```



# Conclusion

- Android App Development using Android Studio
- **Task 1:** Activity Lifecycle
- **Task 2:** Activity Layout XML
- **Task 3:** Intent and Intent Filters

# Thank You!

