Introduction to PIS: Integrated Software Project

Karim Lekadir, PhD

Universitat de Barcelona
Dpt. Matemàtiques & Informàtica

Objectives

1. Learn how to build an app using Android
2. Apply knowledge from past classes
3. Learn (a little bit) about the software development life-cycle
4. Learn how to work in a team
5. Learn how to learn (e.g. how to find information)
6. Gain experience on soft skills for software development

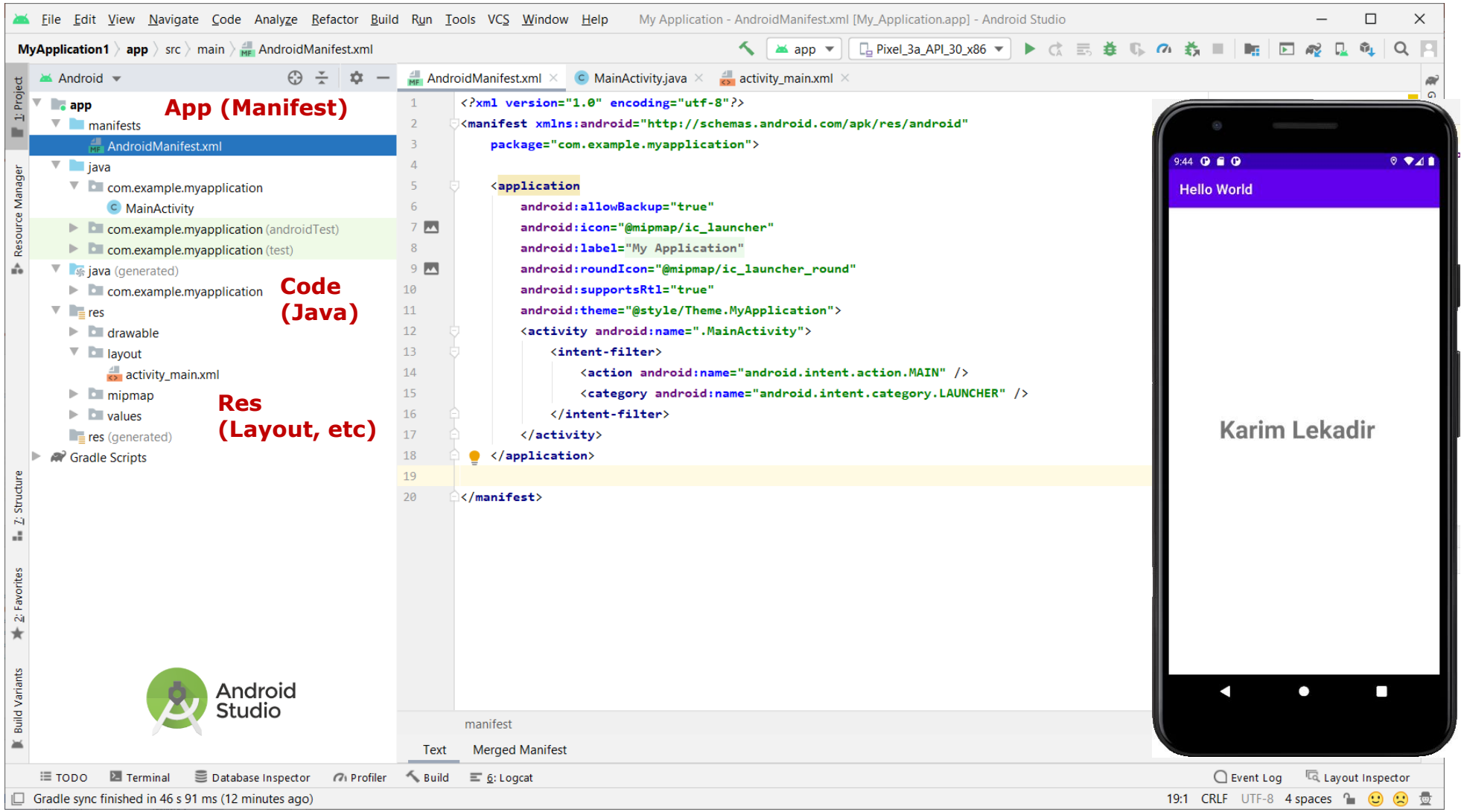
Objectives

1. Learn how to build an app using Android
2. Apply knowledge from past classes
3. Learn (a little bit) about the software development life-cycle
4. Learn how to work in a team
5. Learn how to learn (e.g. how to find information)
6. Gain experience on soft skills for software development

Android

- Linux-based (highly modular and secure)
- Open-source and free
- Easy to use (and to re-use code)
- Portable: Can be used in mobiles, tablets, watches, etc
- 85% of app share (2016)

Android Studio



1. Manifest

XML file containing all key information about the app:

- Name
- Main components
- Activities
- Permissions
- Software requirements
- Hardware requirements

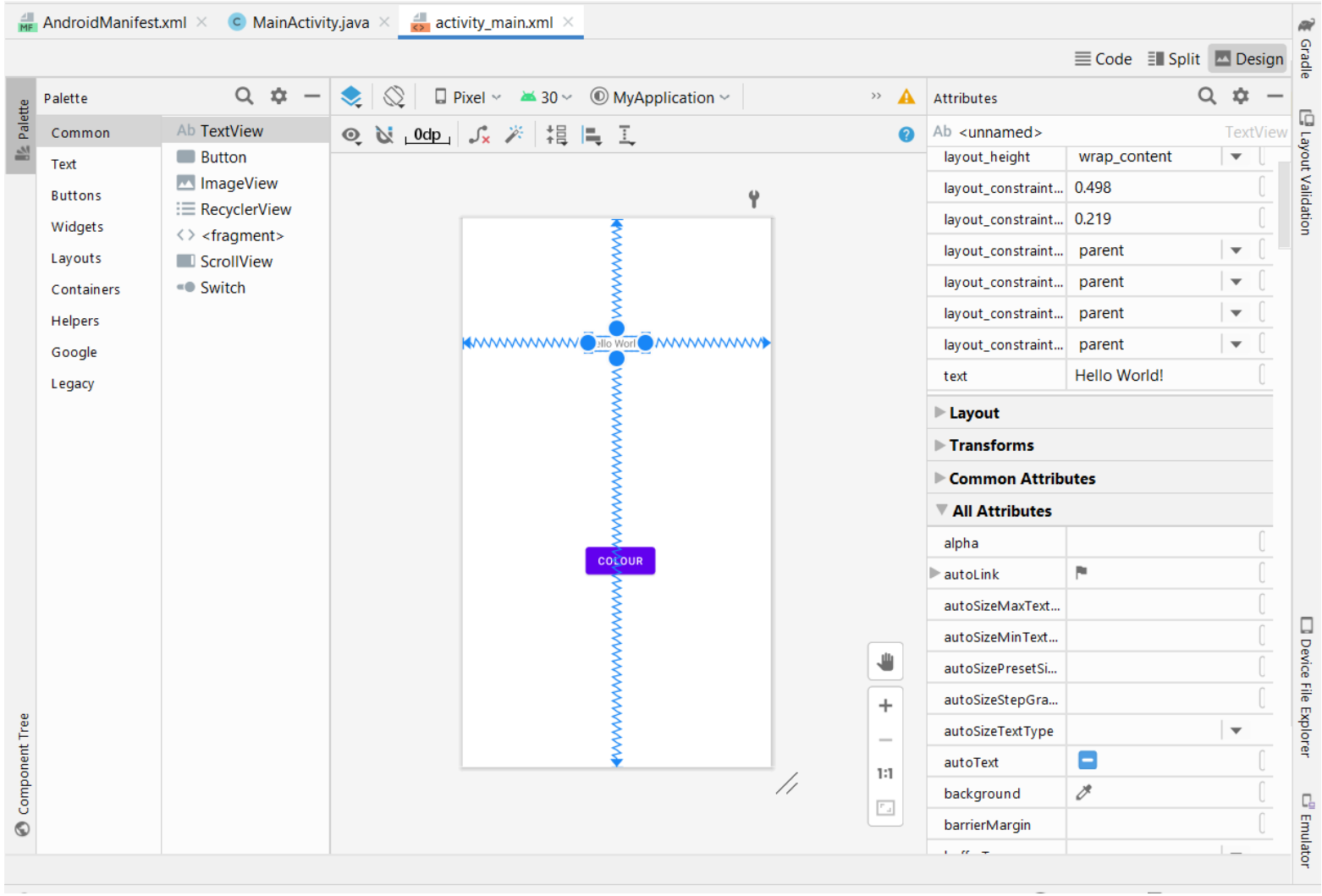
AndroidManifest.xml | MainActivity.java | activity_main.xml

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

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="My Application"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.MyApplication">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

2. Layout


AndroidManifest.xml
MainActivity.java
activity_main.xml
Code
Split
Design
Palette
Common
Text
Buttons
Widgets
Layouts
Containers
Helpers
Google
Legacy
TextView
Button
ImageView
RecyclerView
<fragment>
ScrollView
Switch
Pixel
30
MyApplication
0dp
COLOUR
Attributes
<unnamed>
TextView
layout_height
wrap_content
layout_constraint...
0.498
0.219
parent
text
Hello World!
Layout
Transforms
Common Attributes
All Attributes
alpha
autoLink
autoSizeMaxText...
autoSizeMinText...
autoSizePresetSi...
autoSizeStepGra...
autoSizeTextType
autoText
background
barrierMargin
Gradle
Layout Validation
Device File Explorer
Emulator
Component Tree

3. Code

AndroidManifest.xml | MainActivity.java | activity_main.xml

```java
package com.example.myapplication;

import ...

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        ConstraintLayout bgElement = (ConstraintLayout) findViewById(R.id.activity_main);
        bgElement.setBackgroundColor(Color.RED);
        myButtonListenerMethod();
    }

    public void myButtonListenerMethod() {
        Button button = (Button) findViewById(R.id.button);
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                ConstraintLayout bgElement = (ConstraintLayout) findViewById(R.id.activity_main);
                int color = ((ColorDrawable) bgElement.getBackground()).getColor();
                if (color == Color.RED) {
                    bgElement.setBackgroundColor(Color.BLUE);
                    button.setText("BLUE");
                }
                else {
                    bgElement.setBackgroundColor(Color.RED);
                    button.setText("RED");
                }
            }
        });
```

```
com.example.myapplication.R.id
public static final int activity_main = 1000134
```

R: Object containing all resource IDs

Functionalities (e.g. getColor())

Objectives

1. Learn how to build an app using Android
2. Apply knowledge from past classes
3. Learn (a little bit) about the software development life-cycle
4. Learn how to work in a team
5. Learn how to learn (e.g. how to find information)
6. Gain experience on soft skills for software development

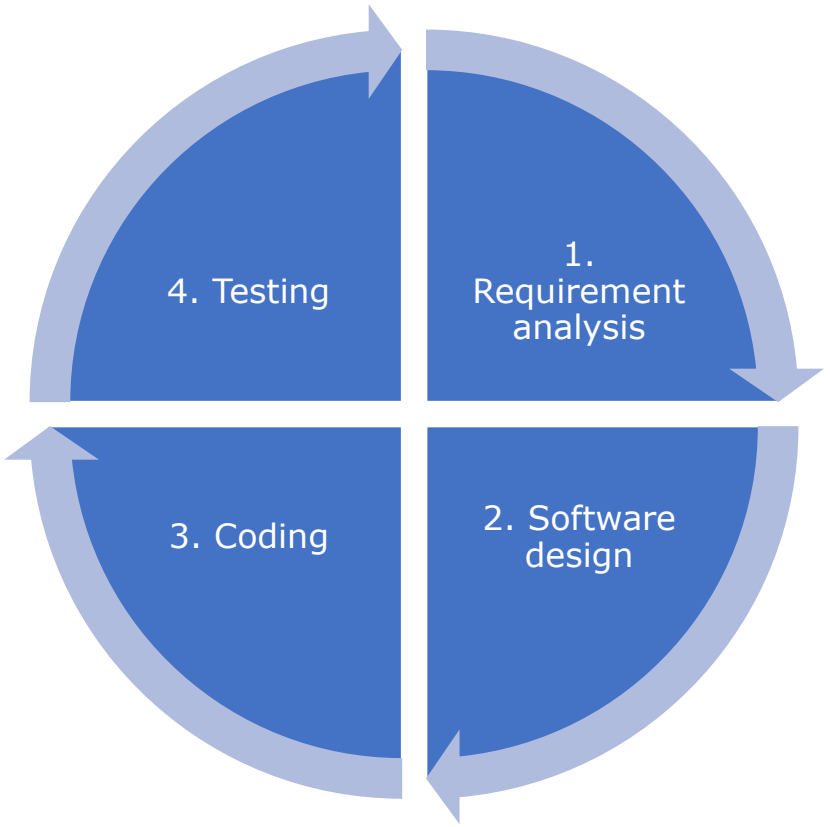
Knowledge Application

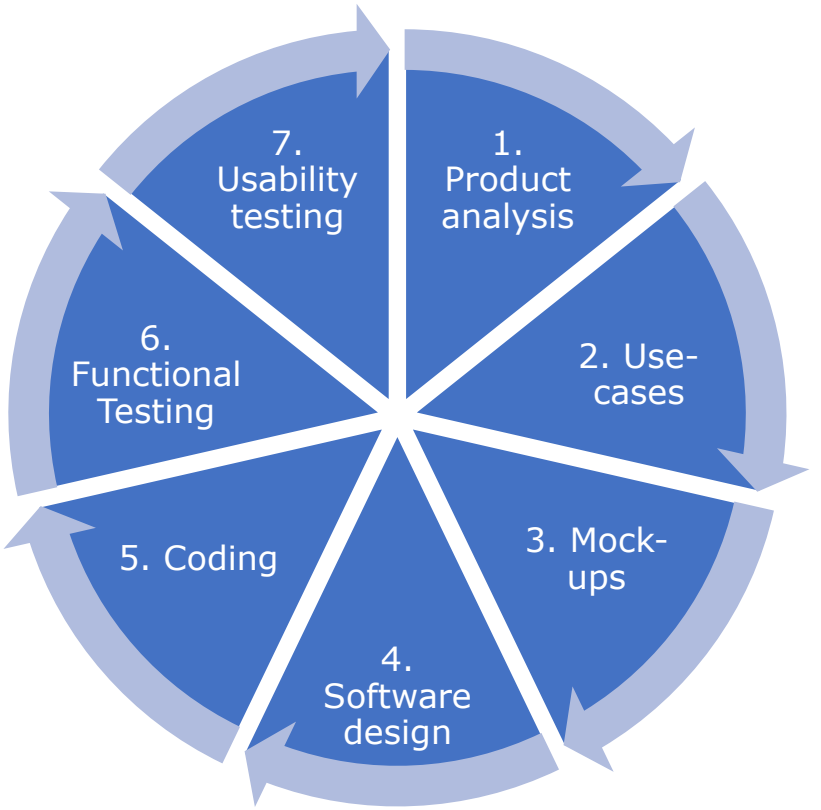
Past classes	Future classes
Programming I	Databases
Programming II	Distributed computing
Algorithms	Artificial intelligence
Advanced algorithms	Computer vision
Data structures	Graphics and visualization
Software design (patterns)	Software engineering

Objectives

1. Learn how to build an app using Android
2. Apply knowledge from past classes
3. Learn (a little bit) about the software development life-cycle
4. Learn how to work in a team
5. Learn how to learn (e.g. how to find information)
6. Gain experience on soft skills for software development

Development Life-Cycle





Objectives

1. Learn how to build an app using Android
2. Apply knowledge from past classes
3. Learn (a little bit) about the software development life-cycle
4. Learn how to work in a team
5. Learn how to learn (e.g. how to find information)
6. Gain experience on soft skills for software development

Team-Work

- Contribute to your team and help each others
- Define roles & responsibilities (incl. leader)
- Communicate and meet regularly (once a week?)

TE Scores & Team-Work

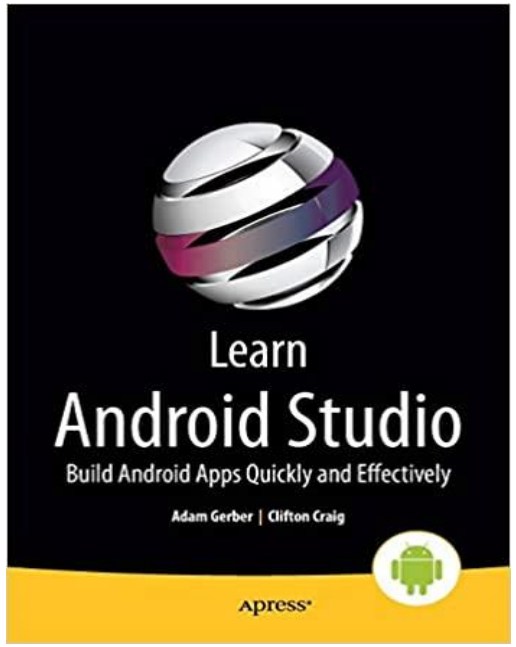
- Give a score (TE: Team Evaluation) to each term member on their contribution to the project (/10):
  - ✓ Member 1: 10/10
  - ✓ Member 2: 10/10
  - ✓ Member 3: 2/10
  - ✓ Member 4: 8/10
  - ✓ Lab teacher: 10/10
  - ✓ **TE (Median): 10/10**

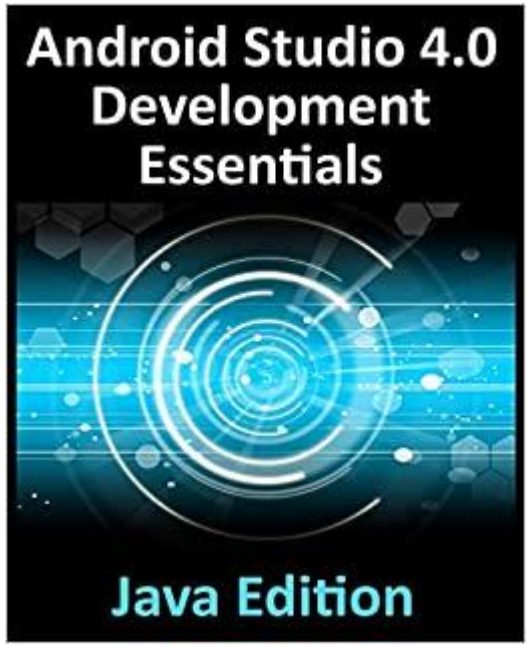
Objectives

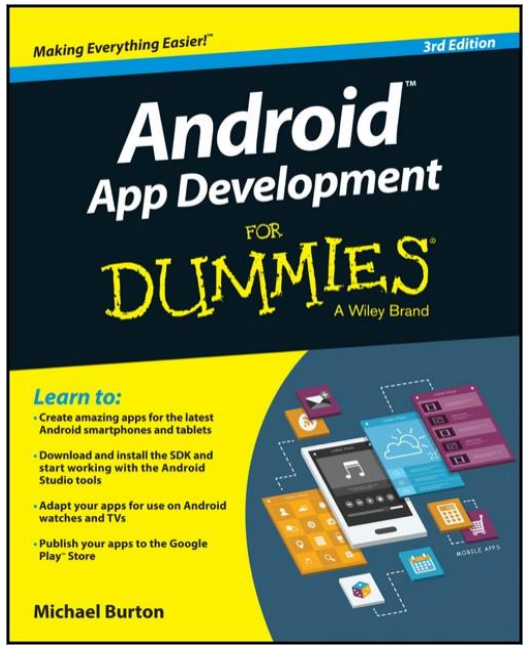
1. Apply knowledge from past classes
2. Learn how to build an app using Android
3. Learn (a little bit) about the software development life-cycle
4. Learn how to work in a team
5. Learn how to learn (e.g. how to find information)
6. Gain experience on soft skills for software development

Finding Information

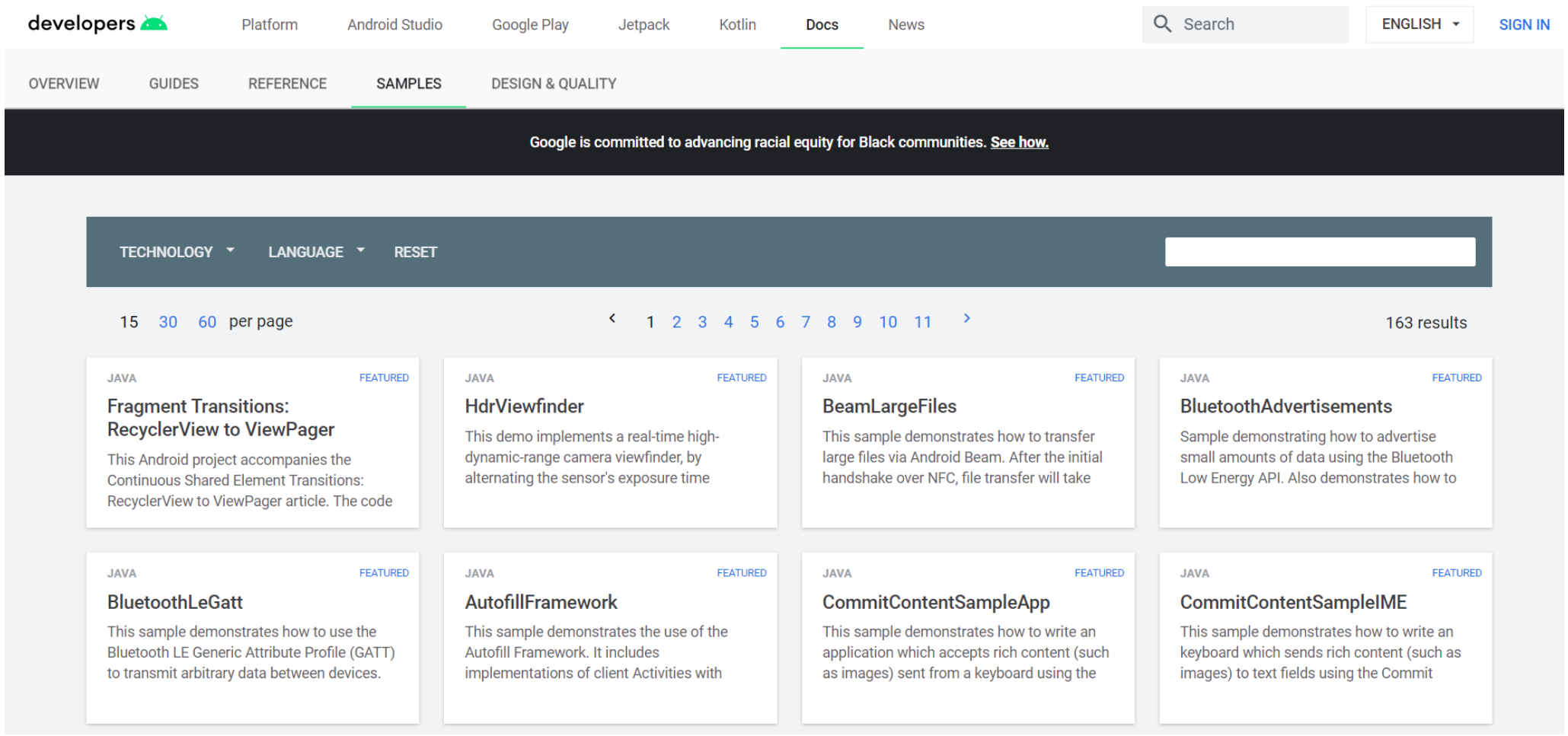
- Many topics in Android: UIs, text, sound, images, videos, calendars, GPS, maps, social media, external data, databases, wearables, sensors, etc, etc…







Finding Information



Objectives

1. Apply knowledge from past classes
2. Learn how to build an app using Android
3. Learn (a little bit) about the software development life-cycle
4. Learn how to work in a team
5. Learn how to learn (e.g. how to find information)
6. Gain experience on soft skills for software development

Soft Skills

Communication, presentations (e.g. demos), documentation, problem-solving, critical thinking, project management, etc

https://www.youtube.com/watch?v=MnrJzXM7a6o

Deliverables

Only three deliverables for the whole course:

1. Project team proposal/requirements + User Interface (UI)
2. Software design + demo
3. Final project + presentation

Continuous Evaluation

A. Project team proposal + User Interface (UI)
B. Software design + demo
C. Final project + presentation
D. 30-minute exam (multiple choice questions)

$$\text{Final mark} = (A*0.25 + B*0.25 + C*0.25)*TE + D*0.25$$

TE = Team Evaluation, will be normalised between 0.5 and 1

Unique Evaluation

A. Project team proposal + User Interface (UI)
B. Software design + demo
C. Final project + presentation
D. 30-minute exam (multiple choice questions) + 2.5 hours coding

Final mark = (A*0.20 + B*0.20 + C*0.20)*TE + D*0.40

TE = Team Evaluation, will be normalised between 0.5 and 1

Teachers

Theory (One hour per week)

Karim Lekadir

Labs (2.5 hours per week)

Carlos Martin

Victor Campello

Calendar

Teoricopràctica [Presencial]									
M1	dl.	dt.	dc.	dj.	dv.	2n sem.	11.00-12.00	Lekadir , Karim	Aula B5	Castellà

Pràctiques de laboratori d¿ordinadors [Presencial]									
a00	dl.	dt.	dc.	dj.	dv.	2n sem.	08.30-11.00	Martín Isla, Carlos	Aula ID	Català
b00	dl.	dt.	dc.	dj.	dv.	2n sem.	08.30-11.00	Martín Isla, Carlos	Aula IG	
c00	dl.	dt.	dc.	dj.	dv.	2n sem.	08.30-11.00	Campello Román, Víctor Manuel	Aula IG	
f00	dl.	dt.	dc.	dj.	dv.	2n sem.	08.30-11.00	Campello Román, Víctor Manuel	Aula ID	

Exàmens : 1r parcial [Presencial]					
G1	5 d'abril de 2022.	18.00-21.00	Lekadir , Karim Campello Román, Víctor Manuel Martín Isla, Carlos		-

Exàmens : Final [Presencial]					
G1	2 de juny de 2022.	15.00-18.00	Lekadir , Karim Campello Román, Víctor Manuel Martín Isla, Carlos		-

Exàmens : Reavaluació [Presencial]					
G1	22 de juny de 2022.	18.30-21.30	Lekadir , Karim Campello Román, Víctor Manuel Martín Isla, Carlos		-

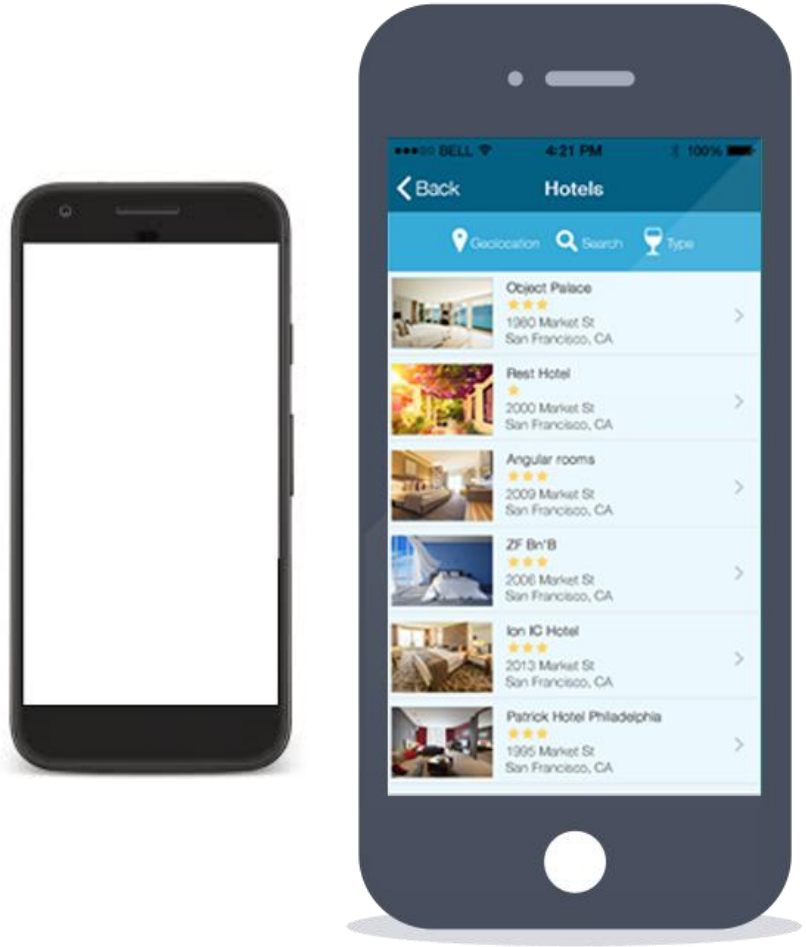
Calendar

Week		Theory	Labs	Deliverables
1	16 February	Introduction to PIS	Introduction to Android	
2	23 February	Design pattern 1	Session 1	
3	2 March	Requirements	Session 2	
4	9 March	Examples requirements	Session	
5	16 March	Feedback deliverable 1	Feedback & support	
6	23 March	Design pattern 2	Feedback & support	1. Requirements + UI
7	30 March	Software testing	Feedback & support	
8	6 April	Week partials (not for PIS)	Feedback & support	
9	13 April	*Semana Santa*	Feedback & support	
10	20 April	Feedback deliverable 2	Feedback & support	
11	27 April	*Matefest-Infofest (no lectiu)*	Feedback & support	2. Design + demo
12	4 May	*Fira d'Empreses*	Feedback & support	
13	11 May	Feedback deliverable 3	Feedback & support	
14	18 May	Trial exam	Feedback & support	
15	25 May	Trial exam	Presentations	3. Final project

Project 2022

How to choose a project:

- Something you like
- Something you can do
- Something that will allow you to learn
- BUT: keep it as simple as possible



Project 2022

Note-taking App integrating multiple options:

- Text
- Photo
- Sound
- Link (e.g. Facebook event)
- Drawing
- Calendar
- File
- Etc

Project 2022

Note-taking App integrating multiple functionalities:

- Add
- Edit
- Remove
- Share
- See statistics
- Add to calendar
- Mark as important
- Etc

Project 2022

Can be domain-specific:

- Personal
- Group (e.g. family, friends)
- Work
- Sports
- Cultural events
- Political
- News
- Etc…

Next…

1. Implement App examples in the labs
2. Set up groups: About 4 members
3. Start working on your project proposal
4. Next week we will start with Deliverable 1 of the project: Requirements & Mock-ups

Question 1

Manifest.XML in Android are used to specify:

A. The main objects and classes.

B. The main files in the project.

C. The main properties and components of the App.

D. The main layout of the App.

Question 2

Which of these lifecycles of software development is incorrect?

A. Product analysis, requirements, design, coding, testing.

B. Mock-ups, use-cases, design, coding, testing.

C. Requirements, design, coding, functional testing, usability testing.

D. Requirements, design, coding, functional testing.

Question 3

A software team member :

A. Must have only one role

B. Can have maximum two roles

C. Must be involved in all roles

D. Should be involved in all software development phases