PROFESSIONAL ANDROID APP DEVELOPMENT

LEARN FROM GOOGLE DEVELOPER EXPERTS

to be a Professional Android App Developer
using best practices and latest technology

Description

Learn mobile application development on the Android platform by acquiring strong knowledge of Android SDK and different versions of Android. This course is for anyone interested in creating applications for the Android platform. You will gain familiarity with RESTful APIs to connect Android applications to backend services and will practice with Genymotion emulator. It takes you through a series of sample projects, each introducing new features and techniques to get the most out of Android. It covers all the basic functionality to get started, as well as the information for experienced mobile developers to enhance existing products or create innovative new ones.

As a part of this course, you will create widgets, customize list view, create 4 applications using facebook, twitter, maps & location based services and test Android-based **mobile applications** using Android testing tools as Mockito.

Our Android course is taught by a group of **Google Developer Experts** and other industry professionals, who develop innovation mobile apps.

What you'll learn:

- Firebase and Android
- Model-View-Presenter (MVP)
- Clean Architecture Android
- Create and display a notification to the user
- Building at least 5 android applications.
    - #01 App: Building a Basic Chat Application
    - #02 App: Create a Twitter App
    - #03 App: Integrating Facebook with my App:Facebook Recipes
    - #04 App: Your social network of photographs!
    - #05 App: Building a note-taking app for android
- Assures the best possible performance, quality, and responsiveness of the application.

- Integrate code from an external support library
- Use the system log to output debug information

Target Audience

This course is designed to attract an audience as: Developers or anyone interested in building Android applications, It's expected that you'll have some experience in software development and be familiar with basic object-oriented development practices. An understanding of Java syntax (classes, control flow statements, instantiation, inheritance, initialization, assignment, method etc) is a requirement and detailed knowledge and experience is a distinct advantage, though not a strict necessity.

Course Length

8 weeks

Estimated Effort

8-10 hours per week

Syllabus

Lessons	Content covered	Competencies	App Development
Lesson 1 App #01: Building a Basic Chat Application	● Introduction ● Android Architecture ○ Introduction to Android Architecture ○ Model-View-Controller (MVC)& Model-View-Presenter (MVP) ○ Model View View Model (MVVM) ○ The Clean Architecture ● Android Architecture (Implementation) ○ How to implement MVP Architecture ○ EventBus ● About Firebase ○ Firebase ○ Firebase and Android ● User Interface (UI)	**The student will demonstrate:** ● An understanding of the Model-View-Presenter ● An understanding of	**APP (1)** Creation of the "Android Chat" application, login, contact list, add your own contacts and chat **APP (2)** Creation of the "Twitter client" application developed using, the same application development kit that provides Twitter Company called "FABRIC", it can interact with elements of twitter

	○ CoordinatorLayout ○ Constraint Layout ● Libraries ○ Butterknife, Glide & ImageCircleView ○ Butterknife (Tutorial) ● Working with my APP(1) ○ Android Chat App ● Setup Description ○ Firebase Signup ○ Setup Description ● Chat Login ● Contact List ● Add Contact ● Chat ● Recap Lesson 1	the Clean Architecture ● The use of Firebase ● How to apply and synthesize knowledge of advanced user interface design ● Knowledge of Android Libraries ● Knowledge to integrated with third-party as Twitter API ● Knowledge to integrated with third-party as Facebook API ● An understanding of how to include features that expand the application's audience through localization ● How to apply and synthesize knowledge of testing and debugging applications	in different ways. **APP (3)** Integrating an app with FACEBOOK, to develop this application, the expert used an external API called Food2Fork. Through this API, the necessary requests were made, together with "retrofit" to integrate with FACEBOOK. **APP(4)** Creation of a "social feed photos" in which everyone can share and see the result of the photographs, the application is developed using two backend "firebase" and "cloudinary", the last one is used to store the photographs
Lesson 2 App #02: Create a Twitter App	● Introduction ● Dart and Henson (Tutorial) ● Dependency Injection ● User Interface (UI) ○ ViewPager ○ Tabs ● Working with my APP(2) ○ Twitter App ● Twitter Kit (Tutorial) ● TwitterClient Setup, Domain, Libs ○ Setup (Gradle and Fabric) ○ Setup (API) ○ Setup (Libs) ○ Setup (Dependency Injection) ● TwitterClient Login ○ Login (login) ● TwitterClient Main ○ Main (Layout) ○ Main (Viewpager y tabs) ● TwitterClient Images ○ Images (Layout) ○ Images (MVP)		

	○ Images (Adapter) ○ Images (Fragment) ○ Images (Presenter & Interactor) ○ Images (Repository) ○ Images (Libs) ○ Images (Injected Fragment) ● TwitterClient Hashtags ○ Hashtags (Layout) ○ Hashtags (MVP Structure) ○ Hashtags (Adapter) ○ Hashtags (Repository) ○ Hashtags (Fragment) ● Navigation Drawer (Tutorial) ● Firebase Cloud Messaging (Tutorial) ● Recap Lesson 2		
Lesson 3 App #03 : Facebook Recipes	● Introduction ● Libraries ○ Retrofit ○ DBFlow ● API ○ Food2Fork ● User Interface (UI) ○ Swipe ● Working with my APP (3) ○ Facebook Recipes ● Facebook SDK (Tutorial) ● FacebookRecipes Setup, Domain, Libs ○ Setup (Config) ○ Setup (Data base) ○ Setup (Retrofit API) ○ Setup (Libs) ● FacebookRecipes Login ○ Login (Layout) ○ Login (Functionality) ● Volley, Retrofit 2 and OkHTTP		

	● FacebookRecipes RecipeMain ○ Main (Layout) ○ Main (MVP Structure) ○ Main (Activity) ○ Main (Menu) ○ Main (Presenter) ○ Main (Interactors) ○ Main (Repository) ○ Main (Injection) ○ Main (Swipe) ● GreenDAO, ORMLite and SugarORM ● FacebookRecipes RecipeList ○ List (Layout) ○ List (MVP Structure) ○ List (Adapter) ○ List (Activity) ○ List (Test Adapter) ○ List (Presenter & Event) ○ List (Interactors) ○ List (Repository) ○ List (Injection) ● Realm ● Recap Lesson 3		
Lesson 4 App #04: PhotoFeed	● Introduction ● Maps ○ Locations ○ Maps ● Camera ● Working with my APP (4) ○ PhotoFeed ● PhotoFeed Setup, Domain, Libs ○ Project Intro ○ Firebase Cloudinary ○ How to use the Cloudinary Android		

	SDK (Tutorial) ○ Google Maps ○ Project Setup ● Firebase DB Rules ● PhotoFeed Login ● PhotoFeed Main Activity ● Open Street Maps (Google maps alternative) ● Take pick photo ● Map-Fragment ● Resources Lesson 4		
Lesson 5 Building a note-taking app for android	• Introduction • Testing • Libraries o Robolectric & Mockito • Setup o Gladle o Base class • Testing API o Get recipe from backend o Getnorecipe and getrandomrecipe • MockWebServer (Tutorial) • RecipeMain o Presenter o Presenter save, dismiss and getnext o Prensenter remaining methods o Interactors o Repository - Saverecipe o Repository - Getnextrecipe o Gesture detector o View o Activity		

	• RecipeList o Presenter o Presenter - on event o Interactors o Repository - getsavedrecipes o Repository - update and delete o UnitShadowRecyclerView Adapter o Adapter setup o Adapter set, remove and itemclick o Adapter subitems click o Activity View o Activity UI o Activity recyclerview item click o UnitActivity clicks and test • Advanced Espresso • Firebase Test Lab • Recap Lesson 5 This lesson is about doing testing operations using software in an automated way, it explains the importance of this practice, which can be combined with a test conducted by another developer, however, it has several advantages to use software for testing, this lesson explains, how to do it and what software be use.		
Final project (Verified Learners Only) The Final Project is the culminating assignment of a series of activities and lessons that the learners have been doing along this course. It was designed to allow learners to take the knowledge they have gained through the course and put that knowledge into practice.			

Final Exam

The aim of the final exam is to put into practice everything you learned over this course´s six lessons

How to Succeed in this Course?

To be successful in this course the participant is expected to:

- Study all the lecture videos that conform each lesson, in them, the expert develops all the topics that were described on syllabus.
- Review the complementary readings or external references related to the topic.
- Perform all the activities in each lesson (Questionnaires, activities, final project).
- You will learn by doing “the more you practice, the more you can, the more you want to, the more you enjoy it” Practice means to perform over and over again. Each lesson comes with its sample code files, and concise instructions for how to build that code by yourself.
- Think! Reflect! Do! and Practice! about what you are learning.
- Participate in forums, your participation creates an active and dynamic learning community, in which you will find support and at the same time you will have the opportunity to contribute with your knowledge, experience and help your classmates.

Making your own planning and setting intermediate deadlines in advance of the deadline will allow you to achieve your goals.

This Course is Instructor-Paced

It follows a schedule that the instructor sets, with assignments and exams that have specific due dates. In contrast, self-paced courses contain assignments without due dates. You can progress through the course at your own speed.

Course materials become available at specific times as the course progresses. Assignments have due dates, and exams have start and end dates. On the Course page, indicators show when you have a graded assignment, as well as the due date for the assignment.

Grading Policy and Certification

Type of Assignment	Contribution to Final Grade	Total Points Possible	Number of Assignments
Course Content Review	0%	0	Videos, screencasts, and text based materials are for learning purposes only.
Questionnaires	50%	Varies per lesson	Sets of practice problems are interspersed in each lesson of the course.
Exercises	20%	Varies per lesson	5
Final exam	30%	100	1

Final Project (Verified Learners Only)

Verified learners have a special assignment "Final Project"

- The aim of this final project, in this second MicroMasters program course, is to teach learners skills that will be relevant and needed in coming courses about Android App developing.
- As part of the benefits of this final project, verified learners´s work will be guided and reviewed by staff course and peers. Each project will have a complete feedback from experts, reflecting his or her individual overall accomplishments, and a list of advices on how to improve the solution´s final project.

Earning a Certificate

If you are are interested in earning a certificate, you'll need to complete the formal quizzes that are housed at the end of each lesson, exercises, the final project and final proctored exam. Students who complete the formal quizzes, exercises, project and final exam with a cumulative 70% average will earn a **verified certificate** from edX, signifying successful completion of the course.

If you're seeking a verified certificate the **Verified Upgrade Deadline:** July 30, 2017

Once you got your verified certificate you have until the end of the MOOC (August 27, 2017, 23:30 UTC) to complete the formal quizzes, project and final exam—there is no weekly deadline for each one, as long as you earn a passing grade by the end date you will be eligible for the certificate.

A	100-90
B	89-80
C	79-70
Certification earned with 70 or higher.	
D	69-60
F	59-00

Communication policies

Assistant teachers are dedicated to answer the discussion forums. However, the same is not true with email. It is recommended to only use the forums as a single channel of communication, to write doubts or support your colleagues respecting the established categories. It is important to avoid creating new forums as much as possible.

- You are expected to communicate in a professional and friendly manner. Offensive or inappropriate language is subject to the policies of edX and may result in removal from the course.
- You are responsible for ethically contributing to the course. Your submissions in this course must be an accurate and true representation of your own work.

Using Forums (Guidelines)

The forums that are part of this course are a unique opportunity to interact with students from all over the world. The community is diverse in experience, knowledge, language and culture. The forums are an excellent means of communication to post your doubts, comments or concerns. Forums provides us with an incredible resource with different points of view, but remember, you are the one who enriches the forums with your participation and makes the community of the course stay active .

Respect the categories established in the course, for example, if you have doubts of lesson 1 write your doubt in the forum created for the purpose, this way your questions will be solved by a member of the course team or a classmate quickly.

What do we expect from you? Participate actively and support your peers, participate at least once a week and always keep in mind the rules of netiquette.

What can you expect from the course team? A member of our team will review the forums periodically to answer your questions and concerns.

Prerequisites

Students should be familiar with

- Java (it is suggested to review the course " Java Fundamentals for Android Development ")
- Skills in Programming language object-oriented.
- Android fundamentals (It is suggested to review the course “Android App Development for Beginners”)
- For more information about this courses, see MicroMasters Program

Multimedia*:* A variety of multimedia files are used in this course, such as audio, video and animation files. Check that you have the necessary software on your computer and browser compatible with multimedia content.