Android App Development for Beginners

DEVELOP ANDROID APPLICATIONS
*Learning basics skills and all you need to know
to make successful Android Apps.*

Description

This course is designed for students who want to learn how to develop Android apps. You'll learn how to create an Android project with Android Studio and run a debuggable version of the app. You'll also learn some android architecture and the key principles underlying its design, you will have a good understanding of the processes that are involved in an android developed application and you will become familiar with android development tools and user interface. By the end of the course, you'll build four simple apps that you can share with your friends.

Our Android course is taught by a group of google developer's experts who are working full time, developing innovation mobile apps.

What you'll learn:

- Android Application Structure

- Life-Cycle of Application

- Create a custom view class

- Implement a custom application theme

- Define a RecyclerView item list

- Implement menu-based or drawer navigation

- Integrate code from an external support library

- Schedule a time-sensitive task using alarms

- Schedule a background task using Job Scheduler

- Designing and building a functional Android application

- Debugging Android applications using different tools and plugins

- Setting up and understanding your Android Development Environment

- Create four applications in order to practice the concepts learned
- Design and develop your own Android app as part of your final project.

Audience

This course is for people who have some programming experience and have curiosity and passion about the world of Android development.

To this course you need to **be fluent in Java**, notice that we said fluent, no expert. This course is not for people who have never programmed on Java before. It is also not for expert on Android development, if you already have developing applications you can find the next courses of MicroMasters program on Android more challenging . Check this

Course Length

8 weeks

Estimated Effort

8-10 hours per week

Syllabus

TABLE OF CONTENTS	SKILLS AND ABILITIES CHECKLIST

MEET ANDROID STUDIO
 1. Introduction
     a. Introduction
 2. Android Studio Installation
     a. Install JDK (Windows)
     b. Install Android Studio (Windows)
 3. Android Studio Tour
 4. Android Emulator
     a. AVD in Android Studio
	The student will demonstrate understanding of *Integrated Development Environment (IDE)* **by**:
	
	 • Installing and utilizing the Android Studio IDE, Android emulator (or external emulator as Genymotion) and Android Plug-In tools necessary to create new project
	 • Creating application projects using Android Studio's Project Wizard. (Hello world)

b. Hardware Device **5.** Hello World Tutorial	

LESSON 1: BASIC CONCEPTS AND UI
  1. Introduction
    a. Introduction
    b. Learning Path Lesson 1
    c. About the App Lesson 1
  2. Android Overview
    a. Android Overview
    b. Android Architecture
    c. What is an App?
    d. Basics kinds of Apps
    e. App Components
    f. Additional elements of an App
  3. Android Basics Blocks
    a. Developing for mobile
    b. Activities
    c. Activity Life Cycle
    d. Intents
  4. Basic UI elements
    a. Layouts
    b. Input controls
  5. Strings.xml & message localization
  6. Resources and asset files
  7. Android Broadcast Intent and Broadcast Receiver
  8. Debugging
  9. **Working with my App**
    a. Tip Calculator App
    b. Introduction and Project
	The student will demonstrate understanding of the ***Android framework*** by:
	
	● Listing the services involved with the Android Application Architecture (ex. Activity Manager, Views, Content Providers, Resource Manager)
	
	The student will demonstrate how to ***create applications and activities*** **by:**
	
	● Thoroughly understand the life-cycle of an Android application and its main components.
	● Navigating between activities using intents
	● Creating activity hierarchies (i.e. back vs. up navigation)
	● Creating activities that displays a layout resource
	● Describing the Android application life cycle
	● Modifying the Android application manifest.
	
	The student will demonstrate knowledge of ***intents*** by:
	● Starting new activities and sub-activities using implicit and explicit intents.
	● Using intent filters and intent resolution to register activities

<table>
<tr>
<td>

      Creation

    c. Identifying the Views

    d. Choose colors, Strings and Dimensions

    e. Creating the Layout

    f. Making our App Functional

10. Persisting Application State

11. Debug Logcat Errors

12. Recap Lesson 1

</td>
<td>

The student will demonstrate knowledge of ***Testing and Debugging by:***

- Using the system log output (and logcat tools) to trace code execution
- Setting breakpoints and stepping through code in the Android Studio debugger
- Locating an error from the stack trace of an uncaught exception

</td>
</tr>
<tr>
<td>

LESSON 2: WORKING WITH DATA AND MATERIAL DESIGN

  1. Introduction

    a. Introduction

    b. Learning Path Lesson 2

    c. About the App Lesson 2

  2. Recyclerview, Adapter & ViewHolder

    a. Adding Support Library

  3. Fragments

  4. Material design elements

    a. Text Input Layout, Floating Action Button and Snackbar

    b. App bar y Toolbar

    c. Coordinator Layout

  5. Navigation

    a. ViewPager

    b. Drawer

    c. Action Bar Elements

  6. Testing with Espresso

  7. **Working with my App**

    a. About the App Lesson

</td>
<td>

The student will apply knowledge of ***user interface and material design by***:

- Creating a graphical user interface (GUI)
- Building layouts using XML and using Java code
- Using various views (ex. buttons, text boxes, check boxes) and implement menu-based, drawer navigation or interface layouts
- Implement menu-based or drawer navigation. Constructing options menus for action bar navigation
- Displaying layouts in an activity and accessing views from Java code
- Providing alternative resources for device configuration changes
- Grouping common UI design elements with styles
- Customizing the application theme
- Using RecyclerView to display a large data collection in a list or grid
- Handling item touch interactions in

</td>
</tr>
</table>

b. Create Project with a Navigation Drawer
  c. Understand and Customize the Navigation Drawer
  d. Create Fragments for now playing and upcoming Movies
  e. Creating movies lists using a RecyclerView
  f. Movie Detail Screen
  g. UI Testing with Espresso
8. Adding views dynamically
9. Building Layouts for screen configuration changes
10. Working with Custom Styles & Themes
11. Recap Lesson 2
	a RecyclerView

LESSON 3: ASYNCHRONOUS LOADING AND BACKGROUND TASKS
 0. Introduction
  a. Introduction
  b. Learning Path Lesson 3
  c. About the App Lesson 3
 1. Android Hierarchical Navigation
  a. Enabling Deep Links
  b. Navigation with Back and Up
 2. Webview
 3. Custom views
 4. Permission system
 5. Async task, threading and handlers
 6. Using AsyncTask vs. Java Threads (with Handlers)
	The student will demonstrate knowledge of *background tasks and threading by*:
	
	● Async task, threading and handlers
	● Background services (Alarm, JobScheduler)
	● Loader

a. AsyncTask
b. HandlerThread
7. Loader
8. AsyncTaskLoader & CursorLoader
9. Background Services
   a. Alarm Manager
   b. JobScheduler
   c. Sync Adapter
10. Android Scheduling task
11. Working with my App
   a. Android App Launched
   b. Create the basic app structure
   c. Create the Layouts, Strings and Dimensions
   d. Add contacts permissions
   e. Load the contacts using LoaderManager
   f. Send email to wish Happy BirthDay
   g. Adding an About me using webViews
12. Recap Lesson 3	

LESSON 4: PERSISTENT DATA STORAGE
1. Introduction
   a. Introduction
   b. Learning Path Lesson 4
   c. About the App Lesson 4
	The student will demonstrate knowledge of *save data persistently* by:
	
	● Determining appropriate use cases for local persisted data, and designing solutions to implement data storage using files,

2. Access Files in Assets
    a. Using Context
3. Access Resources
    a. Using Drawable
4. Save data and files
    a. Shared preferences
    b. Files
    c. Prevent Block Main Thread
    d. Project Setup (Working with my App)
5. SQLite databases
    a. SQLite Databases
    b. SQLite (Working with my App)
6. Content Providers
    a. Content Providers
    b. Content Provider (Working with my App)
7. Loaders
    a. Using Loaders (Working with my App)
8. Background services
    a. Intentservice
    b. Intentservice (Working with my App)
9. Recap Lesson 4

preferences, and databases.
● Creating an application private SQLite database
● Implementing database queries that return single and multiple results
● Preventing disk access on the main thread

The student will demonstrate knowledge about how to **connect your app to the web** by:

● Sharing Preferences and Files, SQLite databases, Content providers,

LESSON 5: PERSISTENT DATA STORAGE

1. Introduction
    a. Introduction
2. Widgets

The student will demonstrate knowledge to **get ready for deployment app** by:

● Registering and publishing on Play Store

      a. The Basics

      b. Widgets - (Extensions)

3. Notifications

      a. Notifications

      b. Notifications - (Extensions)

4. Getting ready for deployment

      a. Getting Ready for Deployment: Working with Flavors

      b. Playstore Deploy (APK)

5. Publish on Play Store

      a. Create an Application in the Google Store

6. Recap Lesson 5

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

Earning a Certificate

If you 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:** Oct 01, 2017

Once you got your verified certificate you have until the end of the MOOC (Oct 29, 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

Tutors 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 peers; respect 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

Previous experience with programming language Java is expected. You have to be aware that we expect you know the fundamentals, at least most of them. This program is designed for developers and technical professionals familiar with object-oriented programming language and interested in building Android applications.

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.