**Android Developer Fundamentals Course: Lesson Plan (V2)**

*Last updated Thu Sep 06 2018*

*By Jocelyn Becker*

*Copyright Google LLC 2018*

***SHARED PUBLICLY***

[Android Developer Fundamentals (V2)](https://developer.android.com/courses/fundamentals-training/overview-v2) is an instructor-led course created by the [Google Developers Training](https://developers.google.com/training) team. Developers taking the course learn basic Android programming concepts and build a variety of apps, starting with Hello World and working their way up to apps that use a Room database.

Each lesson contains a slide deck, a concepts chapter, and in most cases, one or more hands-on tutorial exercises (also known as codelabs). As developers work through the exercises, they create apps to practice and perfect the skills they're learning. Some lessons are purely conceptual and do not have practicals.

The course is offered as an in-person course at selected colleges, facilitated by college faculty. All instructors are invited to use our materials to run their own courses, in accordance with the Creative Commons license used in the course. The materials are also available online for self-study by anyone who is familiar with the Java programming language.

Android Developer Fundamentals prepares developers to take the exam for the [Associate Android Developer certification](https://developers.google.com/training/certification/associate-android-developer/).

**Prerequisites**

Developers taking the course must have Java programming experience.

Android Developer Fundamentals is intended for computer science and engineering undergraduates who have already learned the Java programming language.

## **Total course hours**

Concept hours: 31, Codelab hours: 31

## **Contents**

#### [Unit 1 – Get started](#_hr7seporzrb3)

[Lesson 1. Build your first app](#_36rgakv6dcmg)

[1.0: Introduction to Android](#_oss68kkhnvef)

[1.1: Your first Android app](#_301r34c3llvj)

[1.2: Layouts and resources for the UI](#_a7r9gn855ypa)

[1.3: Text and scrolling views](#_msyq8bponp25)

[1.4: Resources to help you learn](#_m2cdmloigb1w)

[Lesson 2. Activities and intents](#_sr1k81uwlcy4)

[2.1: Activities and intents](#_3hkl398iatqm)

[2.2: Activity lifecycle and state](#_wc2g5c4a7urh)

[2.3: Implicit intents](#_yi6sh3h20a5v)

[Lesson 3. Testing, debugging, and using support libraries](#_tf9q6dxqae67)

[3.1: The Android Studio debugger](#_n37mnk5lf83o)

[3.2: C App testing](#_c2qtnnydgl17)

[3.3: The Android Support Library](#_g2ib7kmubn65)

#### [Unit 2 – User experience](#_7gtrn4mmv919)

[Lesson 4. User interaction](#_iq1eqrd1cney)

[4.1: Buttons and clickable images](#_g1gsivaktcym)

[4.2: Input controls](#_sqrhan4gnwao)

[4.3: Menus and pickers](#_y3904e70od3u)

[4.4: User navigation](#_cj3wjlu5txsx)

[4.5: RecyclerView](#_cwogc42l7kc8)

[Lesson 5. Delightful user experience](#_eylzyacn1f3t)

[5.1: Drawables, styles, and themes](#_l3j74zb6urfq)

[5.2: Material Design](#_iwqqnjszzeqt)

[5.3: Resources for adaptive layouts](#_ffv3jw8xb81p)

[Lesson 6.Testing your UI](#_qef3wto7zpzq)

[6.1: UI testing](#_uf6fo67h9ef2)

#### [Unit 3 – Working in the background](#_rtuhp0vtzd7s)

[Lesson 7. Background tasks](#_7cgqu6mlg0qu)

[7.1: AsyncTask and AsyncTaskLoader](#_kivc6dft8fn7)

[7.2: Internet connection](#_8wvmwyptkyvk)

[7.3: Broadcast receivers](#_4euc4g52fhtv)

[7.4.0: Services](#_8cw8e2wf7ebd)

[Lesson 8. Alarms and schedulers](#_bxly0z3evth1)

[8.1: Notifications](#_jm9lo7f4fu19)

[8.2: Alarms](#_rp27uu5yipp8)

[8.3: Efficient data transfer](#_ed26oq4wp9ko)

#### [Unit 4 – Saving user data](#_hesao3bf8su0)

[Lesson 9. Preferences and settings](#_7nerd8ukic13)

[9.0: Data storage](#_kf83ky3n8giy)

[9.1: Shared preferences](#_yji8deqskm5m)

[9.2: App settings](#_6n75l8xpzvok)

[Lesson 10. Storing data with Room](#_9ihy7mhmlsqe)

[10.0: SQLite primer](#_ir2jt13je2sd)

[10.1: Room, LiveData, and ViewModel](#_hb036pyd058w)

# **Unit 1 – Get started**

This unit covers installing Android Studio, understanding project structure, building your first app, creating activities, testing your apps, and using the Android Support Library.  
  
First, you deploy a simple Hello World app. You go on to create an app with a single activity, and then you create a multi-screen app that passes data between activities. You also learn how to use the Android Support Library to provide backward-compatibility with earlier versions of the Android system for your app.

**Lecture hours: 10**

**Codelab hours: 11**

|  |  |
| --- | --- |
| **Lesson 1. Build your first app** **This lesson covers:**  - Installing Android Studio.  - Creating an Android app project.  - Deploying the app to an emulator and a device.  - Building a layout with UI elements including a scrolling list.  - Learning where and how to get help with building applications. | Lesson hours:  4 hours Concepts (C)  5 hours Codelabs (P) |
| **1.0: Introduction to Android** *Understand the motivation for developing Android apps* | 0.5 C |
| **1.1: Your first Android app** *Understand the development process for building Android apps.*  **Codelabs**  1.1 P Android Studio and Hello World  *Build your first Android project and run the Hello World app* | 0.5 C  1 P |
| **1.2: Layouts and resources for the UI** *Learn how to add interactive UI elements to your app and understand the range of UI elements available.*  **Codelabs**  1.2 Part A: Your first interactive UI  *Add views in the Constraint Layout editor. Update the UI dynamically depending on user input.*  1.2 Part B: The layout editor  *Update your app's layout to perform well in portrait and landscape mode. Learn how to use different layouts.* | 1 C  1 P  1 P |
| **1.3: Text and scrolling views** *Understand the performance implications of different ways to make content in an app scrollable.*  **Codelabs**  1.3: Text and scrolling views  *Learn more about TextViews, and use a scrolling view.* | 1 C  1 P |
| **1.4: Resources to help you learn** *Understand the ways that developers can learn for themselves.*  **Codelabs**  1.4: Available resources  *Practice finding answers and solving problems on your own.* | 1 C  1 P |

|  |  |
| --- | --- |
| **Lesson 2. Activities and intents**  **This lesson covers:**  - Activities, which are the major building blocks of your app's user interface.  - Intents: learn about both implicit and explicit intents which are used to communicate between activities.  - Callback events that perform tasks in each stage of the activity lifecycle. | Lesson hours:  3 hours Concepts (C)  3 hours Codelabs (P) |
| **2.1: Activities and intents** *Understand how to send an explicit intent to start a specific activity.*  **Codelabs**  2.1: Activities and intents  *Define new activities and start them by sending an explicit Intent.* | 1 C  1 P |
| **2.2: Activity lifecycle and state** *Learn about activity lifecycles.*  **Codelabs**  2.2: Activity lifecycle and state  *Explore activity lifecycles.* | 1 C  1 P |
| **2.3: Implicit intents** *Understand how to send an implicit intent to sends a general request for an activity that can handle the request.*  **Codelabs**  2.3: Implicit intents  *Start a new activity by sending an implicit intent that looks for an activity to handle the request.* | 1 C  1 P |

|  |  |
| --- | --- |
| **Lesson 3. Testing, debugging, and using support libraries** **This lesson covers:**  - Using the debugger.  - Setting and muting breakpoints.  - Unit testing your code.  - Using support libraries to ensure your app is backward compatible with previous versions of Android. | Lesson hours:  3 hours Concepts (C)  3 hours Codelabs (P) |
| **3.1: The Android Studio debugger** *Learn about the Android Studio debugger.*  **Codelabs**  3.1: The debugger  *Troubleshoot problems and debug your code.* | 1 C  1 P |
| **3.2: C App testing** *Overview of Android testing, and you learn about creating and running local unit tests in Android Studio with JUnit.*  **Codelabs**  3.2: Unit tests  *Organize and run unit tests in Android Studio* | 1 C  1 P |
| **3.3: The Android Support Library** *Learn how to use the Android Support Library to get backward-compatible versions of new Android features*  **Codelabs**  3.3: Support libraries  *Use a compatibility class from the support library to provide backward-compatibility for your app.* | 1 C  1 P |

# 

# **Unit 2 – User experience**

Create adaptive, responsive user interfaces that work across a wide range of devices. Create engaging, responsive interfaces that use material design principles. Test your app's user interface.

**Lecture hours: 9**

**Codelab hours: 10**

|  |  |
| --- | --- |
| **Lesson 4. User interaction**  **This lesson covers:**  - Understanding and implementing different navigation paths through your application.  - Customizing user input methods and controls.  - Building responsive navigation.  - Using buttons for navigation. | Lesson hours:  5 hours Concepts (C)  5 hours Codelabs (P) |
| **4.1: Buttons and clickable images** *Learn how to create buttons and clickable images for triggering actions initiated by the user.*  **Codelabs**  4.1: Clickable images  *Use an image as an interactive element to perform an action.* | 1 C  1 P |
| **4.2: Input controls** *Learn about input controls such as switches, spinners and more.*  **Codelabs**  4.2: Input controls  *Show the input keyboard, add radio buttons and a drop down menu (spinner).* | 1 C  1 P |
| **4.3: Menus and pickers** *Learn about the types of menus, dialogs and pickers.*  **Codelabs**  4.3: Menus and pickers  *Set up the app bar and options menu in your app. Use an alert dialog and a date picker.* | 1 C  1 P |
| **4.4: User navigation** *Learn about the different ways to enable users to navigate through your app.*  **Codelabs**  4.4: User navigation  *Add an Up button to the app bar. Add tabs to your app.* | 1 C  1 P |
| **4.5: RecyclerView** *Learn about RecyclerView, which displays items in a list in a way that uses memory efficiently.*  **Codelabs**  4.5: RecyclerView  *Display a list of interactive items in a RecyclerView. Use a floating action button (FAB) to add new items.* | 1 C  1 P |

|  |  |
| --- | --- |
| **Lesson 5. Delightful user experience**  **This lesson covers:**  - Using themes and styles.  - Creating responsive user interfaces that use material design principles.  - Creating layouts that work on different screen sizes and orientations.  - Creating accessible and easily localizable apps. | Lesson hours:  3 hours Concepts (C)  4 hours Codelabs (P) |
| **5.1: Drawables, styles, and themes** *Learn to use drawables, which are compiled images. Learn how styles and themes can give your app a consistent look with less XML code.*  **Codelabs**  5.1: Drawables, styles, and themes  *Add drawables, styles and themes to your app.* | 1 C  1 P |
| **5.2: Material Design** *Learn about material design, a visual design philosophy that allows apps to include material design attributes, such as depth and elevation.*  **Codelabs**  5.2: Cards and colors  *Apply material design guidelines to lists and cards. Use material design colors.* | 1 C  1 P |
| **5.3: Resources for adaptive layouts** *Learn how to create layouts that work well for different screen sizes and orientations, different devices, different locales and languages, and different versions of Android.*  **Codelabs**  5.3: Adaptive layouts  *Use resource layout folders to allow your app to work well in different orientations and screen sizes.* | 1 C  1 P |

|  |  |
| --- | --- |
| **Lesson 6.Testing your UI** **This lesson covers:**  - An overview to UI Testing.  - An introduction to the Espresso Framework.  - Manual Testing.  - Automated Testing.  - Using Espresso and UI Automator.  - Recording Tests. | Lesson hours:  1 hours Concepts (C)  1 hours Codelabs (P) |
| **6.1: UI testing** *Learn about testing the user interface of your app.*  **Codelabs**  6.1: Espresso for UI testing  *Use Espresso, a mechanism for recording user interactions, to test your app's user interface.* | 1 C  1 P |

# 

# **Unit 3 – Working in the background**

This unit covers how to do background work, how to schedule tasks, and how to trigger events. It covers the performance implications of executing work in the background, as well as best practices for reducing battery drain. You learn how Android determines which apps to keep running and which to stop when resources run low.  
  
You write an app that connects to the Internet in a background thread to find the author of any book. You also build apps that send notifications and schedule tasks, and you learn how to implement scheduling functionality for apps that run on earlier versions of Android.

**Lecture hours: 7**

**Codelab hours: 6**

|  |  |
| --- | --- |
| **Lesson 7. Background tasks** **This lesson covers:**  - Establishing an internet connection.  - Sending an HTTP request.  - Parsing a JSON response.  - Running work asynchronously in the background.  - Using AsyncTask and AsyncTaskLoader.  - Working with Broadcast Receivers.  - Understanding Services. | Lesson hours:  4 hours Concepts (C)  3 hours Codelabs (P) |
| **7.1: AsyncTask and AsyncTaskLoader** *Learn about using AsyncTask to run work in the background so that the user does not have to wait for the task to complete.*  **Codelabs**  7.1 P AsyncTask  *Add an AsyncTask to your app to run a task in the background.* | 1 C  1 P |
| **7.2: Internet connection** *Learn how to enable your app to connect to the internet.*  **Codelabs**  7.2 P AsyncTask and AsyncTaskLoader  *Enable your app to connect to the internet using an AsyncTask, then update it to use an AsyncTaskLoader so that the task keeps running if the user changes their device's orientation.* | 1 C  1 P |
| **7.3: Broadcast receivers** *Learn about how to send and process broadcasts.*  **Codelabs**  7.3: Broadcast receivers  *Responds to a system broadcast.Send and receive a custom broadcast.* | 1 C  1 P |
| **7.4.0: Services** *Learn about the different types of services, how to use them, and how to manage their lifecycles within your app.* | 1 C |

|  |  |
| --- | --- |
| **Lesson 8. Alarms and schedulers**  **This lesson covers:**  - Scheduling and triggering background tasks.  - Using alarms.  - Understanding the impact of data transfer on battery power and performance.  - Working with Job Scheduler. | Lesson hours:  3 hours Concepts (C)  3 hours Codelabs (P) |
| **8.1: Notifications** *Learn how to create, deliver, and reuse notifications.*  **Codelabs**  8.1: Notifications  *Send and update a notification.* | 1 C  1 P |
| **8.2: Alarms** *Learn how to schedule alarms.*  **Codelabs**  8.2: The alarm manager  *Schedule and cancel an alarm.* | 1 C  1 P |
| **8.3: Efficient data transfer** *Learn how data transfer can drain your users' batteries, and learn strategies for minimizing battery drain.*  **Codelabs**  8.3: JobScheduler  *Use JobScheduler to schedule tasks in a way that reduces battery drain.* | 1 C  1 P |

# **Unit 4 – Saving user data**

This unit discusses how to store user data. You learn how to use shared preferences to save simple key value pairs, then you learn how to use the Room database to save, retrieve, and update user data. This unit also introduces you to the Android Architecture Components, which represent best practices for structuring your app.

**Lecture hours: 5**

**Codelab hours: 4**

|  |  |
| --- | --- |
| **Lesson 9. Preferences and settings** **This lesson covers:**  - ways to save data  - using Shared Preferences as a way to save data  - using the Settings activity to provide an interface that allowins users to set and save app settings | Lesson hours:  3 hours Concepts (C)  2 hours Codelabs (P) |
| **9.0: Data storage** *Learn the different ways to store data in your app.* | 1 C |
| **9.1: Shared preferences** *Learn when and how to use SharedPreferences to save data as key-value pairs.*  **Codelabs**  9.1: Shared preferences  *Use SharedPreferences to save and retrieve user preferences.* | 1 C  1 P |
| **9.2: App settings** *Learn how to allow users to update and save app settings.*  **Codelabs**  9.2 P App settings  *Add a settings activity to an app to save the user's preferred app settings.* | 1 C  1 P |

|  |  |
| --- | --- |
| **Lesson 10. Storing data with Room** **This lesson covers:**  - intro to SQLite  - Architecture Components  - using Room Database, ViewModel and Repository to save and manage data in your app | Lesson hours:  2 hours Concepts (C)  2 hours Codelabs (P) |
| **10.0: SQLite primer** *Refresher and quick reference to interacting with an SQLlite relational database management system. Android provides a builtin SQLite database.* | 1 C |
| **10.1: Room, LiveData, and ViewModel** *Introduction to Architecture Components, which provides guidelines and libraries to help you implement best practices in app architecture.*  **Codelabs**  10.1 Part A: Room, LiveData, and ViewModel  *Use Android's Room database to save and retrieve data in the database. Use ViewModel and Repository to learn best practices.*  10.1 Part B: Room, LiveData, and ViewModel  *Delete and update items in a Room database.* | 1 C  1 P  1 P |

**That's all folks! THE END**