

Android Bootcamp

Course Summary

Description

The Android Bootcamp Training Course is a hands-on introduction to designing and building mobile applications using the Android open source platform. Android Bootcamp explains the philosophy of developing for Android through its main application development building blocks and their interaction with one another.

This completely hands-on course encourages students to learn by building increasingly more sophisticated and meaningful mobile applications for Android.

By the end of the course, each participant will build their own complete Android application incorporating most of the key aspects of the platform. Typically, we build a Twitter-like microblogging app for Android, but there are other choices depending on the interests of participants.

Objectives

By the end of this course, students will be able to:

- Build your own Android apps
- Understand how Android applications work, their life cycle, manifest, Intents, and using external resources
- Design and develop useful Android applications with compelling user interfaces by using, extending, and creating your own layouts and Views and using Menus
- Take advantage of Android's Application Framework API to build complex applications
- Utilize the power of background services, threads, and notifications
- Use Android's communication APIs for SMS, telephony, network management, and internet resources (HTTP)
- Secure, tune, package, and deploy Android applications

Topics

- | | |
|---|--|
| <ul style="list-style-type: none">• History of Android• Android Stack• Hello Android• Architecting Android Apps• Debugging and Testing Android Apps• Android Security Overview• Activities and User Interface• Action Bar and Navigation | <ul style="list-style-type: none">• Preferences• Services• Broadcast Receivers• Content Providers• Lists and Adapters• Fragments• Testing Overview |
|---|--|

Audience

This course is designed for software developers interested in designing, creating, deploying, and testing applications for the Android mobile phone platform. It is valuable to both novices as well as gurus (who already have experience in developing mobile applications for other platforms).

Prerequisites

To take this course, you must know Java. You should be able to answer most of the following questions:

- What is a difference between a class and an object?
- What is the difference between static and non-static field?
- What is the difference between extends and implements keywords?
- What is an anonymous inner class?
- What is the purpose of @Override?

Duration

Five days

Android Bootcamp

Course Outline

I. History of Android

History of Android module focuses on how we got here. The goal of the module is to give us an idea of design philosophy behind Android, and what it may say about the future of the platform. At this high level, we explore both technical and business choices that effect the platform for a long time. We'll look at how and Android operating system gets put together by various parties involved, from Google to carriers. Topics in this module include:

- A. Vision for Android
- B. History overview
- C. Android Open Source Project (AOSP) versions
- D. OEM add-ons
- E. Third Party Android add-ons
- F. Carrier Bloatware
- G. Android fragmentation

II. Android Stack

Understanding how the entire Android operating system is put together is important in order to understand how to leverage everything it has to offer. This module provides an overview of each layer of the stack and its role. We'll look at both the business and the technical design choices.

Topics in this module include:

- A. Design philosophy
- B. Open Source licenses
- C. Linux kernel space
- D. Native layer
- E. Android Runtime (Dalvik + ART)
- F. Application layer
- G. Apps

III. Hello Android

The goal of this module is to have you write a simple Hello, World application. By the end of this module, you should be able to identify main parts of an Android app. You should also be able to verify that your tools are setup properly and that you can run the application on an Android device, either physical or emulated.

Topics covered include:

- A. Android SDK: How to set it up and what it consists of
- B. Android Studio + Gradle: The power of the tools
- C. Creating a new Android project

- D. Anatomy of a project: What are all the moving parts
- E. Running the app: Emulator, real device, tools to monitor them both

IV. Architecting Android Apps

Android framework provides some new and different concepts for developing an app. The goal of this module is to introduce you to main components used to create Android apps. By the end of this module, you should have a good idea what Android app building blocks are, and their key properties. For each building block, we'll explore its typical usage, the life-cycle, and gotchas related to the implementation. Most of the focus of this module is on the conceptual understanding at the lines-and-circles level of the design. Topics covered include:

- A. What makes up an Android app
- B. Activities and Android UI
- C. Fragments and best practices for reusable UI
- D. Intents, Action Bar, and More
- E. Services, IntentServices, Remote Service
- F. Content Providers and Loaders
- G. Lists and Adapters
- H. Broadcast Receivers
- I. App Widgets
- J. Federation of apps design model

V. Debugging and Testing Android Apps

This module provides an overview of some of the tools available in Android SDK that can help you with debugging and testing your apps. By the end of this module you should have a general knowledge of what tools are available and how to use them. The tools we'll explore include:

- A. Logcat
- B. Step-through Debugger
- C. Method Tracing
- D. Debugging View Layout
- E. Heap Dumps + Memory Leak Detection

Android Bootcamp

Course Outline (cont'd)

VI. Android Security Overview

The goal of this module is to provide high level overview of how Android security is architected. By the end of this module, you should know how the apps are sandboxed how the security is enforced. Topics covered include:

- A. Security design overview
- B. Application sandboxing
- C. Using permissions
- D. Granting and enforcing of permissions
- E. Social vectors of attack
- F. Top 10 Bad Permissions

VII. Activities and User Interface

In this module, you will learn how to create basic user interface in Android. You will learn that there is a declarative (XML-based) as well as programmatic (Java-based) approach to creating Android UI. Additionally, you will learn about activities, one of the most important building blocks for Android apps. You will explore at length the lifecycle of an activity as it is highly managed by the system and is important to understand from performance point of view. Topics in this module include:

- A. Two ways to create Android UI: XML versus Java approach
- B. Main view properties: width, height, ids, strings, and more
- C. Structure of UI: Views and Layouts
- D. Linear Layout vs. Relative Layout vs. Frame Layout vs. Absolute Layout
- E. Localization of UI
- F. Best practices for targeting various form factors: phone, tablet, TV
- G. Handling UI events: a bit about listeners
- H. Best practices when working designing Android UI

VIII. Action Bar and Navigation

Your app is a collection of various "screens", implemented as activities. You also may have a few services, and other features that require a button to click to activate. Android offers a standardized method for managing menus and navigation across various components of your app in its Action Bar framework. In this module, you will learn how to use Action Bar to

implement navigation in your app. Topics include:

- A. What Action Bar is
- B. Enabling the Action Bar
- C. Removing the Action Bar
- D. Adding Action Items
- E. Split Action Bars
- F. Using the App Icon for Navigation
- G. Implementing the "Go Home" App Icon Feature
- H. Implementing "Up" Navigation with the App Icon
- I. Adding Navigation Tabs
- J. Adding Drop-down Navigation

IX. Preferences

Your app often needs to store some user data for future use. For example, it needs to know user's login information for an online service, or other settings. Android framework offers a system for just doing that in form of Preferences. In this module, you will learn how to use preferences within your app. Topics include:

- A. Overview of preferences
- B. Preference resource: defining what you'll store locally
- C. Preference activity: displaying the preference screen
- D. Reading in the preference values
- E. Programmatically editing the shared preferences
- F. Registering for preference changes via a listener
- G. Filesystem overview: where are the files stored and is it secure

X. Services

An Android application is just a collection of various building blocks. Services are one of the major such blocks. They represent something working in the "background" in your app. By background, we mean it is not visible, but is crucial to the functionality of your app. In this module you will learn what services are, when to use them, how they work, and how to implement them. Topics covered include:

- A. Overview of Android services
- B. Service lifecycle
- C. Declaring a service

Registering a service

Android Bootcamp

Course Outline (cont'd)

- D. Starting and stopping a service
- E. Threads and other concurrency considerations with services
- F. Bound versus unbound services
- G. Remote versus local services

XI. Broadcast Receivers

Broadcast receivers are Android's publish-subscribe mechanism that. It is an essential part of most Android apps. In this module, you will learn when to use broadcast receivers, how to implement them, as well as register them for intent actions. Topics covered include:

- A. Broadcast receiver usage patterns: when and why to use them
- B. Implementing a broadcast receiver
- C. Registering a broadcast receiver via the manifest file
- D. Registering a broadcast receiver programmatically

XII. Content Providers

An Android app by default cannot access another app's data. This is the cornerstone of Android security, the principle of sandboxing. But often, you do want an app to share some data with others. Content providers are such an interface to data so that the data can be mashed together across apps. In this module, you will learn how to design and develop content providers. Topics covered include:

- A. Why content providers: an overview
- B. Using existing content providers: creating client code that can read and modify the data managed by a content provider
- C. Creating a content provider: implementing a basic content provider to expose structured data to other applications
- D. Intelligent data loading: using loaders to retrieve a Cursor from a content provider without blocking your application's main thread

XIII. Lists and Adapters

Android apps often have to deal with data, and frequently large amounts of it as well. Think an email client, opening up a mailbox of tens of thousands of emails. Given that the screen itself is rather small, and network connection often limited, the intelligence of how to efficiently handle large data sets on the screen is crucial to many apps. To help with that, Android offers a number of selection widgets, such as a list view. It also provides for a management of data that such widget display in a form of adapters. In this module, you will learn how to work with Lists and Adapters. Topics covered include:

- A. Overview of selection widgets
- B. Working with lists
- C. Working with adapters
- D. Implementing a list view and connecting it with an adapter
- E. Creating custom item views
- F. Creating custom view bindings

XIV. Fragments

Android runs on variety of devices, from phones with small screen sizes, to tablets and large-screen TV sets. Fragments in a nutshell represent a smaller part of a user interface that could take a whole screen, or be part of a larger UI. By creating and using fragments, your app can adapt to variety of devices and screen sizes. In this module, you will get a solid overview of

fragments and will learn how to design them, develop them, and use them both statically and dynamically. Topics covered include:

- A. What is a fragment: an overview and motivation
- B. The compatibility package: targeting pre-Honeycomb devices
- C. Fragment lifecycle
- D. Creating a fragment class
- E. Creating a fragment layout
- F. Statically including fragments in an activity
- G. Dynamically attaching fragments
- H. Handling run-time configuration changes
- I. Using Fragments with no Layouts
- J. Finding Fragments

Android Bootcamp

Course Outline (cont'd)

- K. Fragment Operations
- L. Performing Fragment Transactions
- M. Managing the Fragment Back Stack
- N. Integrating Fragment Action Bar/Options Menu Items
- O. Integrating Fragment Action Bar/Options Menu Items (example)
- P. Communication Between the Fragment and the Activity
- Q. Best Practices: Loose Coupling of Activities and Fragments
- R. Best Practices: Define Fragment Interfaces to Invoke Activity Behavior
- S. Best Practices: The Activity as a Switchboard
- T. Advanced Fragment Initialization
- U. Implementing Dialogs Using Fragments
- V. Using a Fragment-Based Dialog
- W. Fragment-Based Preference Management
- X. Additional Fragment Subclasses

XV. Testing Overview

XVI. Testing is an important part of any app development. There are many facets of testing, such as unit testing, performance, or functional testing. In this module, you will get an overview of various tools that Android supports for testing apps. Topics covered include:

- A. Testing guidelines: why test and what to test?
- B. Overview of Android Testing: unit, functional, integration, black/white box
- C. Android Testing Tools: JUnit, Monkeyrunner, UIAutomator/UIAutomatorViewer, Robotium, Espresso