Big Nerd Ranch

Advanced Android

Training Syllabus

Course Overview

This 4-day class is intended to help intro-level Android developers get the skills needed to work on production-ready applications. They will learn industry best practices and libraries that will enable them to create well designed code that is easy to maintain. They will also learn how to test the different parts of their applications, as well as using performance testing tools to identify potential issues before their users.

Who Should Take This Course

- Current Android developers with 6+ months of development experience on the platform. If you are comfortable with creating new projects and building a few screens but want to learn the practices and tools the industry uses for applications then this class is a good fit for you.
- Students should be familiar with standard Android components and how they work together. Activities and Fragments, their lifecycles, layout inflation, and background threading should be familiar topics to potential students.

Syllabus

Coroutines, Channels, and Flow

- Explore Kotlin's first-party approach to writing asynchronous code.
- Leverage coroutines to efficiently and safely move work between threads.
- Use Channels to communicate between coroutines.
- Represent application state using the Flow API to observe and react to changes in values over time.

Model Architecture

- Configure your application for dependency injection using Dagger 2 so your application components can get the dependencies they need without unnecessary coupling.
- Leverage Coroutines to safely access network resources in your UI.
- Add View Binding to your application to help implement the Model-View-ViewModel architecture pattern.
- Reap the benefits of dependency injection by easily testing your model layer objects.

Jetpack Compose

- Become familiar with Google's new declarative UI toolkit for Android: Jetpack Compose.
- Learn how Compose differs from the Framework UI Toolkit and how to set up Compose in an existing project.
- Add components to your UI using composables like `Text`, `Button`, `Image`, and `CheckBox`.
- Use `Row` and `Column` to arrange content in a composable.
- Use `LazyColumn` to quickly create a scrollable list of items.
- Learn how to create your own composable functions.
- Use the `Modifier` class to set attributes like background color, padding, and click listeners on a composable.
- Learn how to use state and recomposition in Compose to automatically update your UI.
- Explore how to show dialogs in a declarative fashion.
- Explore how Compose-only applications can be built without libraries like AppCompat and Fragment.
- Set styles on your Composable UIs with MaterialTheme.
- Explore CompositionLocal variables to access information that affects your UI.

Android Views

- Create a fully custom list item view to gain as much performance as possible from your UI.
- Apply a notification style to change how your data is displayed to the user depending on the type of data.
- Test the UI of your application by using the Espresso framework to write your integration tests.

Performance Testing

- Utilize the built-in testing tools to identify performance issues in views that can lead to dropped frames and suboptimal user experience.
- Identify memory leaks in your application as well as conditions that can lead to high battery drain and unhappy users.

Creating Builds

- Configure your application to produce different build flavors and identify how these can be used.
- Prepare your application for release in the Play store by configuring your application, signing it, and learning about the release process.