

Android Developer Nanodegree Syllabus



Before You Start

In order to succeed in this program, you should have 1-2 years of experience programming in Java, or another object-oriented programming language like Python or C++. If you don't have this experience, we recommend you enroll in our [Android Basics Nanodegree](#) program.

You also must be proficient using git and GitHub. If you don't have this experience, take our [free course on using git and GitHub](#).

Project 1: Popular Movies App, Stage 1

Build a simple movies app that communicates with the internet and provides a responsive user experience. In this project, you will:

- Fetch data from the Internet with the MovieDB API.
- Use adapters and custom list layouts to populate list views.
- Incorporate libraries to simplify the amount of code you need to write

Supporting Learning Content

Lesson Title	Learning Outcomes
CREATE PROJECT SUNSHINE	→ Create and configure a new project in Android Studio using mock data
CONNECT TO THE INTERNET	→ Connect your project to the cloud and replace mock data with data from a weather service, using Android permissions and network I/O
RECYCLERVIEW	→ Display and interact with scrolling information using RecyclerView, and update your app to display information in individual views
INTENTS	→ Create structure of your app and navigate between screen, using activities from other apps within your own app

Project 2: Popular Movies App, Stage 2

Build off of your existing movies app to create a fully featured application that looks and feels natural on the latest stable Android operating system (Nougat, as of November 2016). In this project, you will:

- Allow users to view and play trailers (either in the YouTube app or a web browser).
- Allow users to read reviews of a selected movie.
- Allow users to mark a movie as a favorite in the details view by tapping a button (star).
- Create a database and content provider to store the names and IDs of the user's favorite movies (and optionally, the rest of the information needed to display their favorites collection while offline).
- Modify the existing sorting criteria for the main view to include an additional pivot to show their favorites collection.

Supporting Lesson Content

Lesson Title	Learning Outcomes
LIFECYCLE	→ Leverage the Android framework to handle the Android lifecycle
PREFERENCES	→ Enable users to set their preferences for customized views, and save those preferences in app settings and configurations
STORING DATA IN SQLITE	→ Build an SQLite database for your app
CONTENT PROVIDERS	→ Use a pre-existing content provider in your app to grab user data from your user's phone
BUILDING A CONTENT PROVIDER	→ Build your own content provider for a todo list app, and update the way data is managed in your app
BACKGROUND TASKS	→ Connect your content provider to your app, using the content provider and database to store information in the background
COMPLETING THE UI	→ Try different views, viewgroups and alternative layouts, perform data binding, make your app accessible
POLISHING THE UI	→ Add visual polish and styling to your app, including custom colors, fonts and styles, accounting for multiple devices

Project 3: Baking App

In this project, you will create an app to view video recipes. You will handle media loading, verify your user interfaces with UI tests, integrate third party libraries and provide a complete UX with home screen widget.

In this project, you will:

- Use MediaPlayer/Exoplayer to display videos.
- Handle error cases in Android.
- Add a widget to your app experience.
- Leverage a third-party library in your app.
- Use Fragments to create a responsive design that works on phones and tablets.

Supporting Lesson Content

Lesson Title	Learning Outcomes
WIDGETS	→ Give users easy access to your app with a homescreen widget
FRAGMENTS	→ Make reusable components in your app to support different form factors
ESPRESSO	→ Use User Interface testing to test and verify workflows in your app
LIBRARIES	→ Use external libraries to personalize your app
FIREBASE CLOUD MESSAGING	→ Add messaging functionality and push notifications to your app using Firebase Cloud Messaging
MEDIA PLAYBACK	→ Integrate video and audio media in your app
PLACES	→ Use the Places API to add location to your app
PUBLISHING YOUR APP	→ Publish your app on the Google Play Store

Project 4: Build It Bigger

In this project, you will use Gradle to build a joke-telling app, factoring functionality into libraries and flavors to keep the build simple. You'll configure a library to connect to a web service for jokes. The finished app will consist of four modules:

1. A Java library that provides jokes
2. A Google Cloud Endpoints (GCE) project that serves those jokes
3. An Android Library containing an activity for displaying jokes
4. An Android app that fetches jokes from a web service and passes them to the Android Library for display

Supporting Lesson Content

Lesson Title	Learning Outcomes
GRADLE FUNDAMENTALS	→ Learn the basics of Gradle and Gradle build scripts
GRADLE FOR JAVA	→ Use Gradle to build Java projects
GRADLE FOR ANDROID	→ Use Gradle with Android Studio, and create free and paid versions of an app
ADVANCED ANDROID BUILDS	→ Use Gradle to automate tasks, including dependency management and unit testing.
SPECIAL TOPICS AND MULTI-MODULE PROJECTS	→ Update old projects and libraries, learn more about custom tasks

Project 5: Make Your App Material

In this project, you will update the look and feel of an app to meet Material Design specifications.

Supporting Lesson Content

Lesson Title	Learning Outcomes
ANDROID DESIGN FUNDAMENTALS	→ Build a layout using Material Design principles
SURFACES	→ Implement paper surfaces in your design
BOLD GRAPHIC DESIGN	→ Use design elements of space, color, type and imagery to make your app more beautiful and impactful
MEANINGFUL MOTION	→ Apply real world principles of motion to make your interface more understandable and continuous
ADAPTIVE DESIGN	→ Implement Material Design for new form factors

Project 6: Capstone, Stage 1

In this project, you will demonstrate the ability to communicate an app idea formally, using:

- An app description
- UI flow mocks, similar to what you've seen in other Nanodegree projects, like the Popular Movies overview
- A list of required tasks that you will complete to build the app, in order

The Capstone project will give you the experience you need to own the full development of an app. This first stage replicates the design and planning experience that proficient Android Developers are expected to demonstrate.

Supporting Lesson Content

Students will rely on skills acquired from all previous lessons to complete this project

Project 7: Capstone, Stage 2

In this project, you will demonstrate the skills you've learned in your Nanodegree journey, and apply them to creating a unique app experience of your own. By the end of this project, you will have an app that you can submit to the Google Play Store for distribution.

The Capstone project will give you the experience you need to own the full development cycle of an app.

Supporting Lesson Content

Students will rely on skills acquired from all previous lessons to complete this project