



NANODEGREE PROGRAM SYLLABUS

# Android Basics



# Overview

We built this curriculum with Google for aspiring Android developers who are new to programming to ensure that you get the real-world skills you need to know how to build and accelerate your journey towards becoming a professional Android Developer.

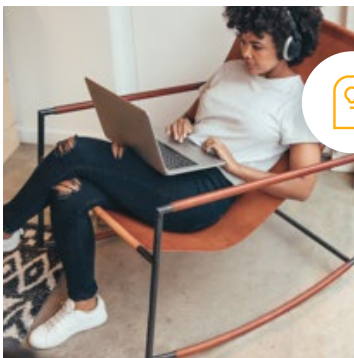
IN COLLABORATION WITH



**Estimated Time:**  
3 Months at  
10hrs/week



**Prerequisites:**  
No Experience  
Required



**Flexible Learning:**  
Self-paced, so  
you can learn on  
the schedule that  
works best for you.



**Technical Mentor Support:**  
Our knowledgeable  
mentors guide your  
learning and are  
focused on answering  
your questions,  
motivating you and  
keeping you on track

# Course 1: Android Basics: User Interface

Learn how to transform hand-drawn app designs into layouts using the XML markup language, and use Views and ViewGroups to display images and text.

## Course Project

### Build a Single Screen App

Design and implement a single screen app that displays information about a fictional small business. Think along the lines of your favorite coffee shop, local restaurant, or that gem of a store that sells those rare comic books or records.

Your design must include:

- Business name
- At least one photo representing the business
- Two or more other pieces of information, such as:
  - Contact information for the business (eg phone number, email address, website)
  - Address of the Business
  - Description of business
  - Hours of operation

## LEARNING OUTCOMES

### LESSON ONE

#### Building Layouts: Part 1

- Create the basic structure of a single-screen app by placing layouts on a page and integrating images, buttons, and text.

### LESSON TWO

#### Building Layouts: Part 2

- Group layouts to allow for more creativity and variation in your design.

## Course 2: Android Basics: User Input

Harness the power of Java to build a coffee-ordering app, use variables to add interactivity to your app, and learn the basics of object-oriented programming.

### Course Project Score Keeper App

Build a Score Keeper app, which gives a user the ability to keep track of the score of two different teams playing a game of your choice.

This project is about combining various ideas and skills we've been practicing throughout the course:

- Adding button code to your app
- Updating views
- Properly scoping variables
- Finding views by their ID

### Course Project Quiz App

The Quiz App project is a chance for you to combine and practice everything you learned in this section of the Nanodegree program. You will be making your own Android app - taking it from the idea stage to building out the full app. You can share your app with family and friends, as well as with other students in this course. The goal is to create an educational app that quizzes a user about a certain topic of your choice. We want you to be creative about how you accomplish this. It's up to you to decide what the quiz questions will be about and how you want to present them to your user. (We recommend about 5-10 quiz questions for a reasonable scope of your app).

This project is about combining various ideas and skills we've been practicing throughout the course. They include:

- Planning your app design before coding.
- Taking an app layout from drawing to XML code.
- Creating, positioning, and styling views.
- Creating interactivity through button clicks and Java code.
- Commenting and documenting your code.

## LEARNING OUTCOMES

### LESSON ONE

#### Making an APP interactive: Part 1

- Create interactive layouts utilizing buttons.

### LESSON TWO

#### Making an APP interactive: Part 2

- Continue practicing creating an interactive user experience.

### LESSON THREE

#### Object Oriented Programming: Part 1

- Create an object in Java and call methods on that object

### LESSON FOUR

#### Object Oriented Programming: Part 2

- Use booleans, conditions and relational operators, and practice adding intents and styles to your app



## Course 3: Android Basics: Multi-Screen Apps

Learn how to show multiple screens, and add audio and images to an app. Learn how Android handles touch events, and make visual changes to add polish to your app.

### Course Project Musical Structure App

The goal is to design and create the structure of a Music app. Note for this project, the app does not actually need to play music. The focus of this project is to implement Intents, new Activities, ArrayLists, Loops and Custom Classes to design the user flow of a music playing app. There are many music player apps, and it will be your job to design the Java structure to store and present the user with song information as well as the user flow through the app. Will you build an app to play music from the user's library of music? Will you build an app to stream random songs from a database? Will you build a musical suggestion engine? Those choices are up to you! This project is about combining various ideas and skills we've been practicing throughout the course. They include:

- Designing an app experience to achieve a certain goal
- Creating new activities
- Using explicit Intents to link between activities in your app
- Using OnClickListener to add behavior to buttons using Java code.
- Creating your own Custom Class
- Looping through an ArrayList
- Populating a ListView or GridView with an ArrayAdapter

### Course Project Tour Guide App

You will be making your own multi-screen Android app to share your knowledge about a city you are very familiar with. The goal is to create a Tour Guide app which presents relevant information to a user who's visiting your city. The app can list top attractions, restaurants, public places, or events for the city. It can contain all the best known secrets that only locals know. It's up to you which categories you want to provide, as well as what information to provide in each category.

This project is about combining various ideas and skills we've been practicing throughout the course. They include:

- Planning your app design and navigation before coding
- Selecting proper data structures to store lists of information
- Building layouts to display those lists of data
- Navigating between lists in Fragments using a ViewPager or Navigation Drawer
- Creating your own Custom Class
- Properly handling images or audio (if applicable)

## LEARNING OUTCOMES

### LESSON ONE

#### Intents and Activities

- Use intents and activities to build out the structure of an app

### LESSON TWO

#### Java Break: Interfaces

- Learn how to take advantage of Interfaces - a core OOP concept in Java that makes your code more flexible.

### LESSON THREE

#### Arrays, List, Loops & Custom Classes

- Learn how to use arrays, lists, and loops to populate your app

### LESSON FOUR

#### Java Break: Loops

- Use for and while loops to iterate over a block of code. Learn how to write nested loops. Use 2D arrays to store and retrieve information.

### LESSON FIVE

#### Quick Guide to Gradle

- Learn how to work with gradle

### LESSON SIX

#### Images and Visual Polish

- Polish image and visuals to improve the quality of your app.

### LESSON SEVEN

#### Activity Lifecycle and Audio Playback

- Use an external library to add audio to your app.

### LESSON EIGHT

#### Practice set: Activity Lifecycle and Audio Playback

- Solidify your understanding of the Activity Lifecycle and Audio Playback.

### LESSON NINE

#### Fragments

- Modify the structure of an app to make a smoother user experience.

# Course 4: Android Basics: Networking

Discover the power of Web APIs and how to use them, and learn the basics of networking in Android, including HTTP networking, JSON parsing, and threads.

## Course Project News App

Create a News feed app which gives a user regularly-updated news from the internet related to a particular topic, person, or location. The presentation of the information as well as the topic is up to you.

This project is about combining various ideas and skills we've been practicing throughout the course. They include:

- Connecting to an AP
- Parsing the JSON response
- Handling error cases gracefully.
- Updating information regularly
- Using an AsyncTask
- Doing network operations independent of the Activity lifecycle
- Use Uri.Builder class to add query parameters to the URL

## LEARNING OUTCOMES

### LESSON ONE

#### JSON Parsing

- Learn how to parse JSON, one of the most useful data formats.

### LESSON TWO

#### HTTP Networking

- Learn how to connect to the internet in your Android code.

### LESSON THREE

#### Threads & Parallelism

- Learn how to use threads to make your device do multiple things at once.

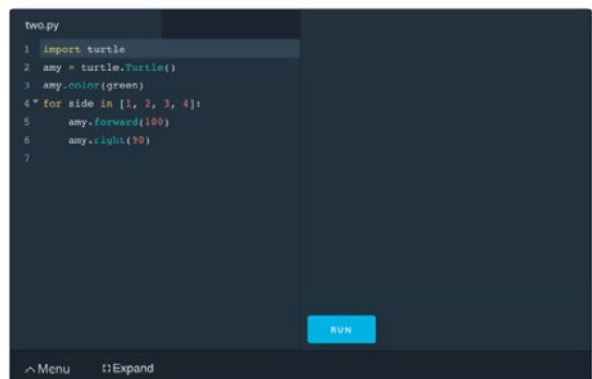
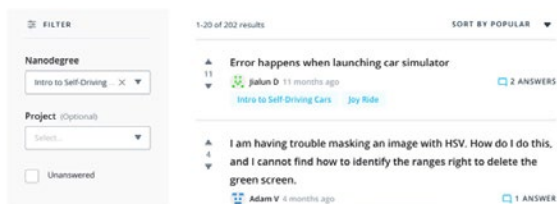
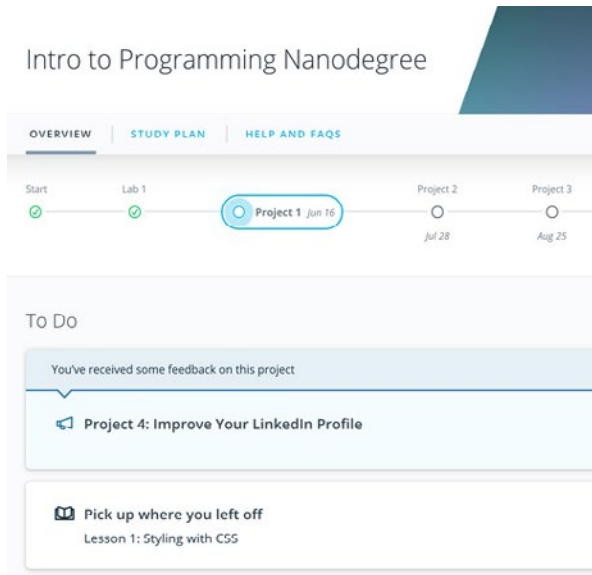
### LESSON FOUR

#### Preferences

- Learn how to add a new Settings Activity and use the user's stored preferences to change the URL



# Our Classroom Experience



## REAL-WORLD PROJECTS

Build your skills through industry-relevant projects. Get personalized feedback from our network of 900+ project reviewers. Our simple interface makes it easy to submit your projects as often as you need and receive unlimited feedback on your work.

## KNOWLEDGE

Find answers to your questions with Knowledge, our proprietary wiki. Search questions asked by other students, connect with technical mentors, and discover in real-time how to solve the challenges that you encounter.

## WORKSPACES

See your code in action. Check the output and quality of your code by running them on workspaces that are a part of our classroom.

## QUIZZES

Check your understanding of concepts learned in the program by answering simple and auto-graded quizzes. Easily go back to the lessons to brush up on concepts anytime you get an answer wrong.

## CUSTOM STUDY PLANS

Create a custom study plan to suit your personal needs and use this plan to keep track of your progress toward your goal.

## PROGRESS TRACKER

Stay on track to complete your Nanodegree program with useful milestone reminders.

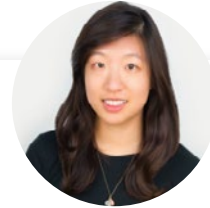
## Learn with the Best



**Katherine Kuan**

INSTRUCTOR

Katherine Kuan, formerly a Developer Advocate at Google, was a software engineer on the Android Apps team for Google Keep, Google Play, and the People app.



**Jessica Lin**

INSTRUCTOR

Jessica Lin is the Android Basics Curriculum Lead at Udacity, teaching various aspects of the Android development ecosystem. When untethered from her devices, she can be found training for her next Muay Thai competition.



**Jennie Kim Eldon**

PRODUCT LEAD

Jennie Kim Eldon is the Product Lead for Android and iOS Nanodegree programs at Udacity, where she previously worked as a software engineer. Before Udacity, she served at the US State Department, leading programs for women and girls in Afghanistan.



**Asser Samak**

INSTRUCTOR

Asser Samak is a Content Developer at Udacity, with over 9 years experience in software engineering, and a great passion for teaching. He also teaches Udacity's Java course series.

## Learn with the Best



**Lyla Fujiwara**

INSTRUCTOR

Lyla Fujiwara is an Android Developer Advocate at Google. Prior to joining Google, Lyla worked at Udacity on the Android Developer and Android Basics Nanodegree teams. She's taught on three continents and is a former Peace Corps volunteer.



**Joe Lewis**

INSTRUCTOR

Joe Lewis has been working with Google Analytics since 2014, and is an author of several books on web development. He is also a professionally-trained classical bassist with a masters degree from New England Conservatory and an avid mountain biker.



**Vicky Tu**

INSTRUCTOR

Vicky Tu is a Course Manager for Android Basics at Udacity, and an Android app developer. In her free time, she enjoys playing with dogs and dancing.



**Chris Lei**

INSTRUCTOR

Chris is inspired and humbled by all who embrace computing as a medium to realize their dreams. He holds an M.S in Embedded Systems and a B.S in Computer Engineering.

# All Our Nanodegree Programs Include:



## EXPERIENCED PROJECT REVIEWERS

### REVIEWER SERVICES

- Personalized feedback & line by line code reviews
- 1600+ Reviewers with a 4.85/5 average rating
- 3 hour average project review turnaround time
- Unlimited submissions and feedback loops
- Practical tips and industry best practices
- Additional suggested resources to improve



## TECHNICAL MENTOR SUPPORT

### MENTORSHIP SERVICES

- Questions answered quickly by our team of technical mentors
- 1000+ Mentors with a 4.7/5 average rating
- Support for all your technical questions



## PERSONAL CAREER SERVICES

### CAREER SUPPORT

- Github portfolio review
- LinkedIn profile optimization



# Frequently Asked Questions

## PROGRAM OVERVIEW

### WHY SHOULD I ENROLL?

Want to develop Android apps, but don't know where to start? In this Nanodegree program, we'll show you the way. We developed this curriculum with Google for true beginners interested in developing apps for the next billion Android users.

Android dominates the market of mobile operating systems, with over 80 percent of the global market share. If you want to develop Android apps, but have no prior programming experience, this is the program for you.

### WHAT JOBS WILL THIS PROGRAM PREPARE ME FOR?

We built this program with Google for true beginners who want to get started building Android apps right away. This program will teach you how to build apps through six (6) hands-on projects, and will also provide you with Java programming foundations.

### HOW DO I KNOW IF THIS PROGRAM IS RIGHT FOR ME?

If you want to build Android apps and learn Java programming basics, and have no prior programming experience, this is the right program for you!

## ENROLLMENT AND ADMISSION

### DO I NEED TO APPLY? WHAT ARE THE ADMISSION CRITERIA?

There is no application. This Nanodegree program accepts everyone, regardless of experience and specific background.

### WHAT ARE THE PREREQUISITES FOR ENROLLMENT?

No prior programming experience is required.

You will need to be comfortable with basic computer skills, such as managing files, running programs, and using a web browser to navigate the Internet.

You will need to be self-driven and genuinely interested in the subject. No matter how well structured the program is, any attempt to learn programming will involve many hours of studying, practice, and experimentation. Success in this program requires meeting the deadlines set for your cohort and devoting at least 10 hours per week to your work. This requires some tenacity, and it is especially difficult to do if you don't find the subject interesting or aren't willing to play around and tinker with your code—so drive, curiosity, and an adventurous attitude are highly recommended!

You will need to be able to communicate fluently and professionally in written and spoken English.



## FAQs Continued

All students will need access to a personal computer that is capable of running **Android Studio**. Please see the System Requirements listed on the **[Android Studio download](#)** page and ensure that your computer meets these minimum requirements.

### IF I DO NOT MEET THE REQUIREMENTS TO ENROLL, WHAT SHOULD I DO?

No programming experience is required. If you are unsure if you are ready for this program, we recommend that you check out our Android Basics: User Interface free course for a preview of the content in this Nanodegree program.

### TUITION AND TERM OF PROGRAM

#### HOW IS THIS NANODEGREE PROGRAM STRUCTURED?

The Android Basics Nanodegree program is comprised of content and curriculum to support six (6) projects. We estimate that students can complete the program in three (3) months working 10 hours per week.

Each project will be reviewed by the Udacity reviewer network. Feedback will be provided and if you do not pass the project, you will be asked to resubmit the project until it passes.

#### HOW LONG IS THIS NANODEGREE PROGRAM?

Access to this Nanodegree program runs for the length of time specified in the payment card above. If you do not graduate within that time period, you will continue learning with month to month payments. See the **[Terms of Use](#)** for other policies around the terms of access to our Nanodegree programs.

### SOFTWARE AND HARDWARE

#### WHAT SOFTWARE AND VERSIONS WILL I NEED IN THIS PROGRAM?

All students will need a personal computer that is capable of running **Android Studio**. Please see the System Requirements listed on the **[Android Studio download](#)** page and ensure that your computer meets these minimum requirements.

Access to an Android device is helpful, but not necessary. You may use the emulator in Android Studio to run your apps if you do not have a physical Android device.

