



Programming Mobile Applications for Android Handheld Systems

Part of the ["Mobile Cloud Computing with Android" Specialization »](#)

Introduction to the design and implementation of applications for handheld systems, such as smartphones and tablets, running the Android Platform.

[Preview Lectures](#)

About the Course

Handheld systems, such as smartphones and tablets are now the most common way for people to access and interact with computing services. The demand for application development skills is therefore growing at a breathtaking pace. These skills, however, are multi-- faceted, requiring students to master computer science and engineering principles, to learn the details of specific mobile application platforms, and to design artistic and engaging user interfaces that respond to how, where and why handheld applications are used.

This course will cover the fundamental programming principles, software architecture and user experience considerations underlying handheld software applications and their development environments. To bring these concepts alive, the course will involve in--depth, hands--on examples, implemented in the Android Platform, the fastest growing segment of the handheld system user base. Students will apply these teachings, also using the Android Platform, in laboratory projects and in a large--scale semester project.

Note: This course is part of a trans-institution course sequence.

This course and two others, led respectively by Drs. Douglas Schmidt and Jules White of Vanderbilt University, have been designed to complement each other. Therefore, some of the programming assignments and the course project for these courses will be coordinated.

This course focuses on handheld systems and the design of user-facing applications, and will be taught first. The first Vanderbilt University course, [Pattern-Oriented Software Architectures: Programming Mobile Services for Android Handheld Systems](#) will focus on systems programming topics, such as middleware services and background processing. That course is currently scheduled to start in March 2014. The second Vanderbilt University course, [Programming Cloud Services for Android Handheld Systems](#) will focus on connecting Android mobile devices to cloud computing and data storage resources, essentially turning a device into an extension of powerful cloud-based services on popular cloud computing platforms, such as Google App Engine and Amazon EC2. That course is currently scheduled to begin in early June 2014.

Nevertheless, each of these courses stands alone. Students are not required to take all of them. Those who do, however, will gain a much more detailed, end-to-end understanding of handheld systems and their applications.

Course Syllabus

Tentative Schedule

Week #1:

- Lecture #1 – The Android Platform
- Lecture #2 – The Android Development Environment
- Lab #1: Setup: Students identify required software & install it on their personal computers. Students perform several tasks to familiarize themselves with the Android Platform and Development Environment.

Week #2:

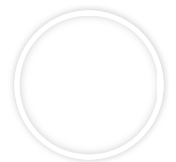
- Lecture #3 – Application Fundamentals
- Lecture #4 – The Activity Class
- Lab #2 – The Activity Lifecycle & Reconfiguration: Students build applications that trace the lifecycle callback methods issued by the Android platform and that demonstrate Android's behavior when the device configuration changes (e.g., when the device moves from portrait to landscape mode and back).

Week #3:



■ [Verify](#)
■ [Speci](#)

Instruct



Catego

[Information, 1](#)
[Computer Sc](#)

Share



- Lecture #5 – The Intent Class
- Lecture #6 – Permissions
- Lecture #7 – The Fragment Class
- Lab #3a - Intents & Permissions: Students build applications that require starting multiple Activities via both standard and custom Intents.
- Lab #3b - Permissions: Students build applications that require standard and custom permissions.
- Lab #3c – Multi-pane and single-pane User Interfaces: Students build an application that uses a single code base, but creates different user interfaces depending on a device's screen size.

Week #4:

- Lectures #8 – User Interface Classes - Part I
- Lectures #9 – User Interface Classes - Part II
- Lab #4 – ToDoManager: Students build a ToDo list manager using the user interface elements discussed in lecture. The application allows users to create new ToDo Items and to display them in a ListView.

Week #5:

- Lecture #10 – User Notifications
- Lecture #11 – The BroadcastReceiver Class
- Lecture #12 – Threads, AsyncTask & Handlers
- Lecture #13 - Alarms
- Lecture #14 - Networking
- Lab #5a – Threads: We'll write concurrent, multi-threaded code to load ToDo items from a networked server via background threads (i.e., without blocking the main UI thread).
- Lab #5b - Broadcast Receiver: We'll build an application that uses a BroadcastReceiver to react when events such as connecting and disconnecting the charger occur.

Week #6:

- Lecture #15 – Graphics & Animation I
- Lecture #16 – Graphics & Animation II

- Lecture #17 – Multi-touch & Gestures
- Lecture #18 – MultiMedia
- Lab #6a - Gesture Sampler: Students build an application that accepts gesture input, such as using an "X" gesture to delete, using a "?" gesture to show help, etc.
- Lab #6b - Bubble Popper: We'll write an application to display and animate bubbles (graphics that look like bubbles) on the device's screen. When users touch the screen where a bubble appears, the bubble pops.

Week #7:

- Lecture #19 – Sensors
- Lecture #20 – Location & Maps
- Lab #7a - Obstacle Course: Students build an application that uses the orientation of the device (tilting, rotating, etc.) to guide an object around obstacles.

Week #8:

- Lecture #21 – DataManagement
- Lecture #22 – The ContentProvider Class
- Lecture #23 – The Service Class
- Lab #8a - Data Management (SQL): Students develop a database for storing and retrieving multimedia notes with textual tags.
- Lab #8b - Multimedia Notes Content Provider: Students extend the multimedia notes database so multimedia notes can be shared via a ContentProvider across multiple applications.

Final Project: iRemember

We'll implement the front end of a complex handheld application involving many of the concepts presented in the class.

Recommended Background

This course is directed to Sophomore- or Junior-level undergraduate students. Students should already know how to program in Java, but are not expected to have studied mobile application development.

If you don't already know Java, but have some familiarity with programming languages, you can improve your Java knowledge, by taking one of the many Java tutorials and online courses available on the web.

Suggested Readings

There is no textbook for this course, but I strongly encourage students to explore the wide range of freely-available Android-related resources.

- Keep up with changes and improvements in the Android ecosystem by reading the [Android Developer's Blog](#).
- Watch tutorials and other presentations from the [Google I/O](#) conference.

Course Format

Each lesson will consist of video presentations, logically divided roughly into 5-10 minute sections. Throughout the lecture video there are ungraded "in-video" questions to help ensure that students are understanding the material. Each week there is also a short quiz containing questions/tasks to help ensure that the students have understood that week's lectures.

The in-video questions are related to the current video section, and provide supplementary information, "Food For Thought" discussion, and "FLASHBACK" questions, which challenge the students to think about something we have discussed in a previous lesson.

Each week's videos will be accompanied by a lab exercise consisting of an assignment write-up, partially completed source code, and executable test cases. The students will run the test cases on their own computers to determine whether they've successfully completed the lab. For some labs, we will also require students to examine and grade the submissions of other students.

Each lesson covers multiple Android source code examples. I strongly encourage students to download all the examples and to consult them both while watching the lecture and afterwards. The source code is publicly available on github:

<https://github.com/aporter/coursera-android>

The course will also have a complex final project.

FAQ

Some Frequently Asked Questions.

- **When I try to watch the videos before the class start, I get an error message.**
You can preview some of the videos by pressing the "Preview Lectures" Button at the top of the [course webpage](#)
- **Is the course broadcast live? I live on the other side of the world from you!**

No. Course lectures are videotaped. Students watch the lectures and do programming assignments and quizzes when it's convenient for them.

- **What are the course objectives?**

Upon completing this course, students should be able to:

- Understand the Android platform's organization, patterns and programming mechanisms and be able to use them effectively to develop their own Android applications.
- Use development tools, such as those found in the Android Developer's Toolkit to efficiently create, understand, debug and optimize Android applications.
- Understand the key forces and constraints acting on handheld devices and know how to accommodate these when designing and building their own Android applications.
- Know where to find additional sources of information to understand and solve Android-related problems.

- **Will I get a Statement of Accomplishment after completing this class?**

Yes. Students who successfully complete the class will receive a Statement of Accomplishment signed by the instructor.

- **What resources will I need for this class?**

For this course, you'll need is an Internet connection, a computer on which to run free Android developer tools, and the time to read, write, and discuss.

- **What is the coolest thing I'll learn if I take this class?**

How to write the software applications that you and half the world are running on your handheld devices.

- **What computer languages do I need to know?**

Some experience with Java should be enough. As mentioned in the Recommended Background Section, those who don't know Java, but have some familiarity with other languages may want to take a Java tutorial prior to starting this course.

Can students use programming language, other than Java, for the course?

Not really. The main programming language for Android is Java.

- **What should I be reading to prepare for class?**

There is no course textbook. If you want to get started early, dive into the [Android Developer's Website](#)

- **Do I need to buy an Android device?**

No. All the graded exercises will be done using the Android Emulator.

- **What is a "trans-institution sequence of MOOCs?"**

This MOOC course and another course tentatively called [Pattern-Oriented Software Architectures for Concurrent and Networked Mobile Devices and Clouds](#), taught by Professors Doug Schmidt and Jules White of Vanderbilt University have been designed to complement each other. In particular, some of the programming assignments and the course project for both courses will be coordinated. The focus of Vanderbilt MOOCs will be systems programming topics, such as server-side Android concurrency, background processing, networking, and computing cloud integration, and will be taught starting in late March 2014. Additional information on our trans-institutional MOOC sequence is available [here](#) and [here](#).

- **Is it necessary to take all the courses in the sequence?**

No. If you just want to take some of the courses in this sequence--or take them all in different order--you're certainly welcome to do so, and you'll still learn a lot. However, if you take all the courses in this sequence in the order presented you'll gain a deeper, end-to-end understanding of handheld systems, their applications and services, as well as their integration into the cloud.

- **When will the course material be made available each week?**

All the course material (e.g., video lectures, quizzes, short essays, programming assignments, etc.), for each week will be made available at 12:00am eastern time (5am UTC/GMT). each Tuesday of the course.

- **How does this MOOC compare/contrast with courses at the University of Maryland?**

This MOOC is heavily based on courses I teach at UMD, called CMSC436, Programming Handheld Systems. The course lecture material is similar, but the quizzes, programming assignments, and level of feedback for the UMD courses are more challenging, given that we have about twice the amount of time to work on it. Also, as the UMD course has many fewer students, there's significantly more personalized guidance from the professor and TAs that can't (yet) be replicated via a MOOC. This is one reason why it's important for students to take on some of the role of the Instructor. When all of us work together, we all benefit.

- **Will there be a Statement of Accomplishment for students who complete this class?**

Yes. Students who successfully complete the quizzes and assignments in this class will receive a Statement of Accomplishment signed by the instructor. There will be two levels of Statements of Accomplishment: Normal Track and Distinction Track. The difference between these two tracks is that the Distinction Track

requires successful completion of the class project.

- **What is the most effective way to learn material covered in the course?**

I recommend watching the videos multiple times, looking for different levels of meaning in the diagrams and the examples. Likewise, I recommend reading outside sources of information. Naturally, participating in the online discussion forums (and ideally, a meetup group if one is available in your area) will help make the course material more engaging.

- **Can students take this course if they have no prior experience with Android programming or programming with Java?**

This course assumes that students are comfortable programming in Java and have some experience programming Android apps. If you don't have any significant Java programming background, please look on the Internet for one of the many Java-related tutorials that are freely-available.

- **Will you use Eclipse or Android Studio in this course?**

All of my videos use Eclipse as Android Studio is still in a pre-release state. If you're using a non-Eclipse development environment, such as AndroidStudio, you may need to enlist the help of other students on the online discussion forum.

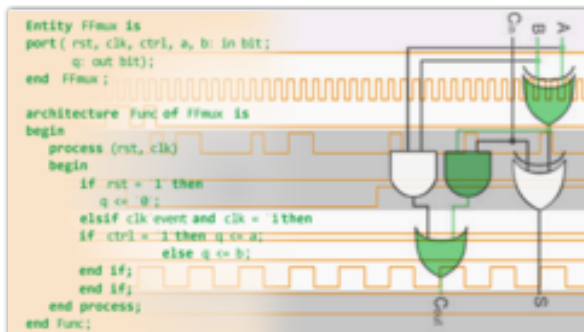
- **Which web browsers are recommended?**

Coursera recommends using the Chrome and Firefox browsers. There's also a [mobile app](#) for Coursera MOOC, as well.

- **Where can students download the slides that are presented in the videos?**

PDF versions of the slides will be available online as the videos are released.

Related Courses



Digital Systems - Sistemas Digitales: De las puertas lógicas al procesador



La visione del mondo della Relatività e della Meccanica Quantistica



وية

[Browse more courses](#)

Coursera empowers people to improve their lives, the lives of their families, and the communities they live in with education.

© 2014 Coursera Inc. All rights reserved.

COMPANY[About](#)[People](#)[Leadership](#)[Careers](#)**FRIENDS**[Partners](#)[Community](#)[Programs](#)