

# CS 395 - Mobile Development for Social Change

Sorelle Friedler  
sorelle@cs.haverford.edu

Spring 2015

Mobile Development for Social Change will focus on standard software engineering principles, object oriented programming, event-driven and multi-threaded programming, Android-specific mobile development concepts, and designing a positive user experience in the context of a semester-long placement with a local non-profit.

## Prerequisites

CMSC 105 and 106 or the equivalent (with permission of the instructor).

## Textbooks and Devices

*Required (free subscription codes to be provided to the class):*

- Mark L. Murphy. *The Busy Coder's Guide to Android Development*. CommonsWare, 2014.

*References (available in the science library):*

- Reto Meier. *Professional Android 4 Application Development*. John Wiley & Sons, Inc., 2012.
- G Erich, H Richard, J Ralph, V John. *Design patterns: elements of reusable object-oriented software*. Reading: Addison Wesley Publishing Company, 1995.
- McConnell, Steve. *Code complete*. Microsoft press, 2009.
- Sommerville, Ian. *Software Engineering*. Addison-Wesley, 2007.

Students are not required to own an Android device; the class will be using Eclipse, which includes an Android emulator. However, students will find that using an Android device is different than using an application on an emulator, and so Android devices are available for students to check out from the science library for debugging and design purposes and for use when visiting the local non-profit with which they have a placement.

## Class Time and Office Hours

Class: F 1:30 - 4pm in KINSC H110.

Lab: T 2:30 - 4pm in KINSC H110.

Office Hours: TBA or by appointment.

Add the class Google calendar for all times and deadlines. Office hours and any changes to them will be indicated there.

## Schedule of Topics

This schedule is *tentative*. **Hard copies are required** for all non-programming assignments - any assignments missing a hard copy by the 10am deadline will not be accepted. Students should expect **at least 10 hours of work each week**. Work for each project checkpoint should be started *well in advance* of that deadline. Project checkpoint deadlines are **10am Friday** on the week in which the checkpoint is listed (unless otherwise specified). More specific project checkpoint expectations will be discussed and distributed during class.

Week 1. Prototypes and how to fail quickly, Java and Android basics, including object oriented programming, testing and documentation, and engineering for social change.

- Project checkpoint (Friday): Groups chosen and a ranked list of three non-profits or on-campus groups handed in at the start of class.
- Watch (due Friday): “The Prototyping Manifesto - Stanford Graduate School of Business,” presented by Alberto Savoia in January, 2012. <http://youtu.be/t4AqxNekecY>.
- Book sections: Key Android Concepts, Android and Projects, Contents of Android Projects, Inside the Manifest, Some Words about Resources, Debugging Your App.
- Lab: Eclipse setup and Hello, World.
- Tutorials: Tutorial #1 - Installing the Tools, Tutorial # 2 - Creating a Stub Project, Tutorial #3 - Changing our Manifest, Tutorial #4 - Adjusting our Resources.

Week 2. Android user interface basics: widgets, layouts, and adapters. More object-oriented programming and basic event-driven programming.

- Project checkpoint (Tuesday): Non-profits or on-campus groups assigned. **Once you have been assigned a non-profit or on-campus group, you have made an honor code commitment to complete the project. This implies a commitment to not dropping this course after this date.**
- Reading (due Tuesday):  
Chapters 1 and 8 from *Persuasive Technology: using computers to change what we think and do* by B. J. Fogg. <http://tripod.brynmawr.edu/find/Record/.b3829565>.  
“What I Learned From Building An App For Low-Income Americans” by Ciara Byrne <http://www.fastcolabs.com/3038792/what-i-learned-from-building-an-app-for-low-income-americans>
- Watch (due Tuesday): A technology TED talk or Solve for X talk.
- Book sections: The Theory of Widgets, The Android User Interface, Basic Widgets, LinearLayout and the Box Model, Other Common Widgets and Containers, GUI Building Continued, AdapterViews and Adapters.

- Lab: Object-oriented programming basics, applying persuasive technology to your project.
- Tutorials: Tutorial #5 - Making Progress.

Week 3. More Android user interface techniques.

- Book Sections: Defining and Using Styles, JARs and Library Projects, The Action Bar.
- Project checkpoint: pretotype.
- Tutorials: Tutorial #6 - Adding a Library, Tutorial #7 - Setting Up the Action Bar.
- Lab: with the DIY class.

Week 4. Android processes, activities, and lifecycles. Fragments. Threads.

- Book sections: Android's Process Model, Activities and Their Lifecycles, The Tactics of Fragments, Dealing with Threads.
- Tutorials: Tutorial #8 - Setting up an Activity, Tutorial #9 - Starting Our Fragments.

Week 5. Saving state, preferences.

- Book sections: Requesting Permissions, Assets Files and Data Parsing, Using Preferences.
- Tutorials: Tutorial #11 - Adding Simple Content, Tutorial #12 - Displaying the Book, Tutorial #13 - Using Some Preferences.

Week 6. Using databases and content providers, including relational databases (SQLite), internet access, and searching.

- Book sections: SQLite Databases, Internet Access.
- Tutorials: Tutorial #14 - Saving Notes.
- Distinguished Visitor Mac Smith at Haverford. Talk on Friday, February 27th at 4:30pm. Attendance is **required**.

Week 7. Advanced event-driven programming - intents, broadcast receivers, services, notifications, etc..

- Book sections: Intents, Intent Filters, Broadcasts and Broadcast Receivers, Services and the Command Pattern, AlarmManager and the Scheduled Service Pattern, Notifications.
- Tutorials: Tutorial #15 - Sharing Your Notes, Tutorial #16 - Updating the Book, Tutorial #17 - Periodic Book Updates, Tutorial #18 - Notifying the User.
- Project checkpoint: prototype. Be prepared to discuss your idea in lab and class.

Week 8. Mid-semester break.

- Week 9. Parallel programming theory, including deadlock, race conditions, and starvation. Multi-threaded / parallel programming, including understanding threads and when to use them, and the separation between the UI thread and worker threads.
- Week 10. Midterm. Large screen strategies and backward compatibility. 3 seminar presentations.
- **Midterm** on Tuesday in lab. Will include all topics covered until now, including last week.
  - Book sections: Supporting Large Screens, Backwards Compatibility Strategies and Tactics.
  - Tutorials: Tutorial #19 - Supporting Large Screens.
- Week 11. Using and designing APIs via Maps and location based services, including understanding how APIs work, API design principles, triangulation vs. fingerprint-based methods of location calculation, and privacy implications of location use. 3 seminar presentations.
- Book sections: Mapping with Maps V2, Accessing Location-Based Services, The Fused Location Provider.
  - Project checkpoint: project completed.
- Week 12. User interface design and user experience research. 3 seminar presentations.
- UI Reading: [http://www.wired.com/business/2012/04/ff\\_abtesting/](http://www.wired.com/business/2012/04/ff_abtesting/)
  - UX Reading: <http://www.uxbooth.com/articles/better-experimental-design-for-better-user-testing/> and <http://www.uxbooth.com/articles/complete-beginners-guide-to-design-research/>
- Week 13. 4 seminar presentations.
- Week 14. 4 seminar presentations.
- Project checkpoint: User experience research completed. Final meeting with the non-profit scheduled (held by the end of exams).
- Week 15. Publishing your app. 3 seminar presentations.
- Book sections: Signing your App, Distribution.
  - Lab: poster presentations.
- Week 16. Exam week. Project checkpoint: available in the Google Play Store. Due at the end of exam week.

## Large Project and Placement

Students will work in pairs on a semester-long project to create an Android application for a local non-profit. All students in the group will receive the same grade for the project. The main expectations for this project are as follows. Completing these items ensures a grade of C on the project. In order to get a B or A, students must go beyond these expectations.

1. The satisfaction of the non-profit with the final product.
2. The inclusion of the following elements in the application:
  - (a) basic event-driven Android elements, e.g., buttons, list pickers, etc..
  - (b) appropriate responses to Android lifecycle events
  - (c) a UI thread and a separate worker thread
  - (d) an external API, e.g., a Google map.
3. A consistently working application that doesn't throw exceptions or errors.
4. A user experience research report whose conclusions have been incorporated into the design of the application.
5. An in-class presentation about the design, creation, and use of the application.

There will be four major deadlines. These deadlines must be met - **no extensions will be given.**

1. By **the end of the third week** of classes, students should have chosen a partner, chosen a non-profit, and met with the non-profit to discuss their needs. A complete prototype of the project should be finished and approved by both partners and the non-profit. Students will also need to have their prototype approved by the instructor to ensure that it is ambitious yet attainable.
2. **Before the mid-semester break**, students should have an outline prototype of their project finished. No polish or graphics are expected at this point, and placeholders are acceptable for large functional sections of the application, but all screens should be present and the flow from one screen to another should be working. This form of the application should be shared with the non-profit and any differences in expectations should be resolved at this point.
3. By **the end of the 10th week of classes** the application should be done. The application should be shared with the non-profit, their feedback should be taken into consideration for revisions, and their help should be acquired (if necessary) in finding 10 potential users for a user experience study.

4. By **the last day of class** the application should be done and in the Google Play Store. A final meeting with the non-profit should have been held, including an evaluation of the non-profit's satisfaction with the application. Results and conclusions from a user experience study involving at least 10 users, and the changes that were made based on this study, should be included in a written report. In the last week of class, pairs will be presenting their work and doing a demo of their application for the class.

### **Seminar presentations**

Each student will be required to individually lead the class for a half hour session once during the semester. The goal of this session will be to teach the class an advanced topic, including a short lab exercise. More details will be discussed in class before the first presentations.

### **Total grade breakdown**

Project and placement	55%
Midterm	25%
Seminar presentations	15%
Attendance / participation	5%

Grades will be awarded based on the number of points earned and according to the percentage breakdowns shown. Students will not be graded on a curve.

### **Late work policy**

All extensions must be requested **at least 24 hours in advance** of the deadline. Extensions will be granted based on individual circumstances. Work handed in late without a previously granted extension may not be accepted.

### **Rules and Pet Peeves**

- **Be on time.** This includes class, lab, office hours, and appointments.
- **Expect 24 hours before an email response.**
- **Read email within 24 hours.**

### **Attendance and Participation**

Attendance at, and active participation in, all class sessions and lab sessions is expected of all students. This includes attendance at the class on the Friday before spring break. Attendance is taken into account strictly in the 5% attendance grade.

### **Collaboration**

You are encouraged to discuss the lecture material and the labs and problems with other students, subject to the following restriction: the only "product" of your discussion should be your memory/understanding of it - you may not write up solutions together, or exchange written work or computer files. The group project is the only exception to this - in this case, these collaboration rules apply to students outside of your group and you

may freely work closely with students within your group. Collaboration is not allowed on examinations or quizzes.

As usual, anything taken from outside sources should be cited. Code should not be copied without permission from the instructor. If permission is given, code should be cited at the location it is used with a comment.

### **Learning Accommodations**

Students who believe they may need accommodations in this course because of the impact of a disability are encouraged to meet with the instructor in private (e.g., during office hours) early in the semester (i.e., ASAP). Students should also contact Rick Webb, Coordinator, Office of Disabilities Services ([rwebb@haverford.edu](mailto:rwebb@haverford.edu), 610-896-1290) to verify their eligibility for reasonable accommodations as soon as possible. Early contact will help to avoid unnecessary inconvenience and delays, and facilitate learning.