CSC263 Programming Mobile Devices II
Spring Semester 2017

INSTRUCTOR: Randi R McCray
rmccray@gwcc.commnet.edu

Syllabus for Mobile Applications

Required Materials:

If you want to use your own computer for Android development, you will need to download and install

- the Java SDK (from Oracle)

- the Android SDK (from Google)

- XCode (from the Apple Store)

Description:

This level two course builds on many of the foundational concepts you learned in the Programming Mobile Devices I course. This level II course will provide a deeper learning and understanding on building more complex, production worthy applications. This essentially means that you will be learning skills, concepts, and tools that will raise your level of expertise and learning as a marketable mobile application developer. The specific activities, assignments, and expectations in this course go beyond level one expectations. Assignments will be more detailed and rigorous and should be submitted on time as they [all] connect to the overall final project in expectations and quality of the end product.

Prerequisites:

Programming Mobile Devices Level I or sufficient learning and understanding of the following concepts; HTML, Javascript, CSS3, Java, Objective C, and/or Swift.

Java is the native programming language of Android, and therefore prior Java knowledge will save a lot of time, and smoothen the learning curve of programming Android Apps. The course will not teach Java – but will focus on those frameworks and classes which are new and unique to Android.

iOS's native programming language is Objective-C. The course will not teach you Objective-C – but will focus on the app programming model. The leap from C to objective-C isn't that large, so if you're comfortable with C, consider this course.

Students should familiarize themselves with the development environments of XCode and Android Studio.

Links to tutorials will be presented, but the IDEs themselves will not be discussed extensively during the course.

Course Expectations:

Be prepared for a *lot* of hard work: Be prepared to code, often, and much. The aim of this course is to get you to a point where you can boast your own "App portfolio". But Apps aren't going to write themselves.

There is a lot of self-study required: Resources on mobile programming are plentiful, and – as we aim to cover two mobile OSes – time is limited. Each presentation will have required pre-reading and post-reading. Make sure to, at the very least, skim the references. Our lecture time is, alas, limited to only 1.5 hours at a time, and the goal is to maximize that time. - Homework assignments and projects will be given in advance.

Don't leave things to the last minute! Often surprisingly simple tasks can take more than you thought they would (q.v. Parkinson's law). - "assignments" are theoretical, to test your comprehension of the architecture. "projects" involve coding.

You should write code that works. You will also began to focus on style and User Interface (UI) design.

Readable code helps, however, in those cases where your code doesn't work as well as you hoped, and we need to pinpoint the problem so as to grade It as fairly as possible. In general, my interest in your code is inversely squared proportional to the performance of your App (in other words, write solid apps which work, and we'll take it for granted the code quality is good!)

Both your mid-term and final project is to be done individually. However, collaboration is welcomed and peer support is key. Avoid plagiarism and reusing code where you have not made significant modifications to build something new.

You should write ORIGINAL code. Reusing open source/Internet-borne code is acceptable, if and only if you point out exactly which component it is that you have used in whatever way. In those reuse cases, you should limit it for specific subtasks for which you do not want to reinvent the wheel (e.g. compression, SSL, etc). Don't try to reuse an entire app.

Evaluation

Your grade for the course will be based on the following percentages:

Labs / Assignments 50%
Final Project 50%

The final project will be in the form of a program. You should do your own work on exams/projects and for computer assignments. Copying another student's work is not acceptable. Any indication of cheating and/or plagiarism on an exam/assignment/project will be an automatic 0 (zero) for the exam/assignment/project for all students involved.

Regarding codes in assignments / projects, you may be required to explain the code you submitted. In case of discursive explanation, the instructor holds the right to lower your grade.

Letter grades will be assigned according to the following scale:

A - at least 90% of the total points
B - at least 80% of the total points
C - at least 70% of the total points
D - at least 60% of the total points
F - less than 60% of the total points

STUDENTS WITH DISABILITIES

The Americans with Disabilities Act (ADA) is a federal anti-discrimination statute that provides comprehensive civil rights protection for persons with disabilities. Among other things, this legislation requires that all students with disabilities be guaranteed a learning environment that provides for reasonable accommodation of their disabilities. If you have a disability requiring an accommodation, please reference the link below to contact the department.

http://www.gatewayct.edu/Offices-Departments/Student-Accessibility-Services/Contact