CS 646 Android Mobile Application Development
Spring Semester, 2017
Syllabus

San Diego State University -- This page last updated 4/19/17

CS 646 Syllabus

Instructor	Roger Whitney
Office	GMCS 561
Phone	594-3535
Email	whitney@sdsu.edu
Office Hours	5:00-6:15 pm Tuesday, Thursday, 10:30 am - Noon Friday

Course Objectives: Learn the basics of developing Android apps.

Course WWW Site: The course website is in the SDSU Blackboard system. SDSU students can access it through their SDSU portal. It can also be access at: https://blackboard.sdsu.edu/

Class Location and Time: The class meets in room GC 1504 (Gateway Center) from 7:00 pm to 8:15 pm Tuesdays and Thursdays.

Textbook:
Android Programming: The Big Nerd Ranch Guide (2nd Edition), Phillips, Stewart, Hardy & Marsicano, Big Nerd Ranch Guides, August 3, 2015, ISBN 978-0134171456

Prerequisites: The course assumes that you know Java. All assignments in the course will use the Java programming language to develop Android apps. This involves designing and building GUIs and using SQL databases. While prior experience with GUIs and SQL databases are not required prior experience will be very useful in the course.

Required Hardware: You can use a Windows, Mac or Linux based computer to develop Android apps. While it is not strictly required to have an Android device it is useful that you have access to one. There is an Android emulator but there are a number of things the emulator can not do like record sound.

On-Line Students: All the course materials are placed on-line via blackboard for access by all students. The class lectures are recorded and can be accessed in realtime during class and after the class is over. There is an on-line discussion board for questions outside of class. On-line students in the San Diego area are welcome to attend class.

Grading: Your grade will be based on one exam (200 points) assignments (500 points), a project (200 points). There is no extra credit work in this course. There will be 5 assignments. All assignments are worth 100 points. Tentative due dates for the assignments are Jan 29, Feb 12, Feb 26, March 19, and April 2. The exam will be March 9.

Remote Students and Exams: Off campus students need to find someone to proctor them while they take the March 9 exam. I need to know who your proctor will be by Feb 21 so I can get the exam to them in time.

Course Project: There is a course project required for this course. You can work on the the course project by your self or in teams of two people. Teams larger than 2 are not allowed. You are to come up with an idea for an Android application and implement the application. The project is due May 12. Your project will be evaluated using the following criteria:

Originality (5 points)
How original is the idea and/or implementation of the project.

Size of project (20 points)
Is the project an appropriate size for a semester project. A project that is too large is as bad as a project that is too small. The project should be larger in scope than an assignments in the course? The project should take longer than 2-3 week to design and implement?

Quality of UI (80 points)
How well does the app follow users mental model of how the app should work. Are the UI elements used effectively or not? Is the app UI structured in a way to make it easy to use and understand. Are the text used in menus, labels, buttons, etc. concise and have clear meaning? Are all the words spelled correctly? Do all titles start with capital letters? Do all the elements fit on the screen? Is the app complete, that is does it correctly handle the lifecycle of its activities, does it support a range of screen sizes and device rotation, does it support user preferences, etc.

Working code (70 points)
Do the features implemented work. All UI elements should actually do what they are supposed to do? Are features fully or only partially implemented. Does the app have enough features to actually do something. Does the app run? Are there bugs and memory leaks? Does the app crash?

Quality of code (20 points)
The code should be formatted in a reasonable and consistent manner. Names of classes, methods and variables should understandable and follow standard naming conventions. The code should be clear and well organized. The code should be appropriately documented.

Readme (5 points)
Short file describing your project. What does the project do? Any special instructions on how to run the project. Information about third party libraries used.

Crash Policy: The last day to add this course is January 31.

Dropping the course. If you drop this course CES will refund part of your fees depending on when you drop the course. If you drop the course at least one day before the first time the class meets you will receive a full refund minus $17 administration fee. If you drop the course after the first meeting of the class but before 25% of the course time as elapsed you will receive 65% of your fees back minus $17 administration fee. After this date you will not receive refund if you drop the course. If you wish to drop this course to take another of the Certificate course this semester you can do so without any penalty. If you have questions about this and other issues related to CES please contact the CES registration office at 619-594-5152.

Late Policy: Late homework will be accepted, but with a penalty. An assignment turned in 1-7 days late, will lose 3% of the total value of the assignment per day late. The eight day late the penalty will be 40% of the assignment, the ninth day late the penalty will be 60%, after the ninth day late the penalty will be 90%. Once a solution to an assignment has been posted or discussed in class, the assignment will no longer be accepted. Late penalties are always rounded up to the next integer value.

Cheating: Any one caught cheating will fail the course and they will be reported to the SDSU Judicial Procedures Office.

Course Goals and Outcomes:

- Design Android apps using Java.
- Use Android IDEs and API effectively in Android applications.
- Implement and debug Android apps.
- Separate resources (text, sound, images) from Android code to facilitate working with graphic designers and localization of apps.
- Measure resource consumption in Android applications to help optimize performance.
- Support multiple versions of Android OS and multiple Android devices with one app.
- Develop Android GUI interfaces separate from Android app code.

This will improve your ability to:

- Analyze a problem, and identify and define the computing requirements appropriate to its solution
- Design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs
- Use current techniques, skills, and tools necessary for computing practice.

Topics covered in the course:

- Using Android Studio
- Android Activities and Activity life cycle
- Fragments
- Layouts, Intents & Intent Filters
- Permissions
- GUI widgets
  - Labels, buttons, Check boxes, text fields, Lists, pickers, etc.
- Containers and Keyboards
- Layout styles
- Threads and Concurrency
- Fonts
- WebViews
- Services
- AppWidgets
- Files, Preferences, SQLite database
- Notifications, Resources
- Network Access
  - JSON,
- 2D Graphics
- Maps
- Dealing with multiple screen sizes
- Testing
- Tablets
- Amazon's Voice & Text recognition for mobile devices
- Sensors
  - Touch, gyroscope, accelerometers

Disabled Students: If you are a student with a disability and believe you will need accommodations for this class, it is your responsibility to contact Student Disability Services at (619) 594-6473. To avoid any delay in the receipt of your accommodations, you should contact Student Disability Services as soon as possible. Please note that accommodations are not retroactive, and that accommodations based upon disability cannot be provided until you have presented your instructor with an accommodation letter from Student Disability Services. Your cooperation is appreciated.