CS175 MOBILE DEVICE DEVELOPMENT

Fall Semester 2017

CS 175

By Dr. Angus Yeung

ESSENTIAL INFORMATION

Instructor:	Dr. Angus Yeung
Office Location:	DH 282
Telephone:	(408) 768-5817
Email:	fai.yeung@sjsu.edu
Office Hours:	Saturday 12:45 – 02:00 or by appointment
Class Days/Time:	Saturday 10:00 – 12:45
Classroom:	MH 422
Prerequisites:	CS 047, and knowledge of Java equivalent to that of CS 046A or CS 049J

COURSE DESCRIPTION

Introduction to building mobile applications for devices based on Android and iOS operating systems, including use of standard IDE as well as testing and debugging on devices and emulators/simulators.

COURSE DESCRIPTION

Topics cover programming languages, Java for Android programming / Swift for iOS programming, and mobile platform APIs for user interface, graphics, networking, data and web services.

PREREQUISITES

Prerequisites are any college-level courses for modern and object-oriented programming languages such as Java.

DR. ANGUS YEUNG

- Head of Applications and SDK, Intel Corporation
  - Managing all Android and iOS core platform development
- Technical Leader (Director Level), Intel Corporation
  - Virtual Reality, Broadcasting for MLB, PGA and Olympics
- Taught at San Jose State University
  - CS 164 Data Structures and Algorithms
  - CS 151 Object Oriented Design and Patterns
  - CS 175 Mobile Device Development
- Taught at Sabanci University, Istanbul
  - CS 480/580 Wearable Computing

DR. ANGUS YEUNG

- Research interests
  - computer vision, machine learning, wearable computing
  - 19 pending patents in the areas of wearable computing
  - conference and journal papers in medical imaging areas
- Qualification
  - Ph.D. from University of Rochester , Electrical & Computer Engineering
  - MBA from UC Berkeley, Haas School of Business
  - M.S. from University of Rochester , Electrical & Computer Engineering
  - B. S. from University of Rochester, Electrical & Computer Engineering

CLASS TA

- Ms. Anukriti Sinha
  - Graduate Student at CS Department
  - Responsible for grading homework and programming assignments

ADD CODE

- Current registration at 29/30
- Will give out 4 more add code in priority order:
  - Graduating Senior – CS students
  - Junior – CS students
  - Graduating Senior – non-CS students

ATTENDANCE

- Attendance won’t impact your class grade directly
- But it may impact your class grade indirectly….

CLASSROOM LOGISTICS

- 15 minute break from 11:15 am to 11:30 am.
- Make yourself comfortable, just don’t fall in asleep
- Sit close to the door if you have to leave early
- Okay to use your laptop
- DON’Ts
  - Do not check your cell phone
  - Do not sleep (in front of me)
  - Do not play games

OBJECTIVES

Course Goals

Learning Outcomes

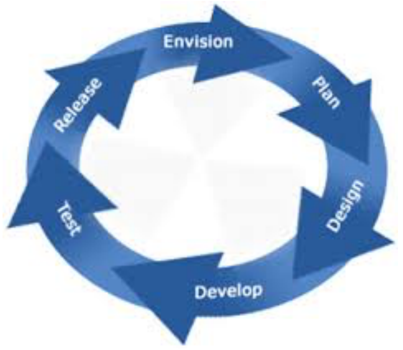
COURSE GOALS

- Understand the mobile software architecture and building blocks for Android and iOS;
- Get familiar with the workflow and lifecycle of components for developing mobile applications;
- Develop Model-View-Controller based app with simple user interface;
- Work with mobile platform framework APIs for device sensors, graphics and location services;

COURSE GOALS

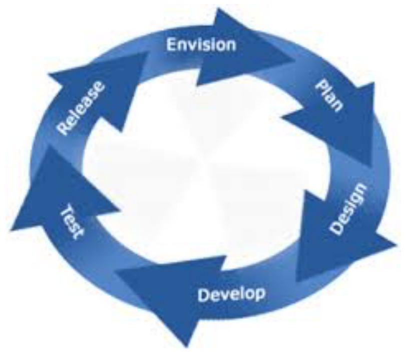
- Understand the messaging and threading model for user interface events;
- Develop multi-threading, concurrent and background processing solutions for mobile applications;
- Work with platform API for persistence storage, database and cloud storage;

MOBILE APPLICATION DEVELOPMENT



- Understand the workflow for mobile app development
- Design, implement, test and debug mobile applications with object-oriented languages
- Use model-view-controller model for user interface programming
- Implement user interface layout design and handle event messaging

MOBILE APPLICATION DEVELOPMENT



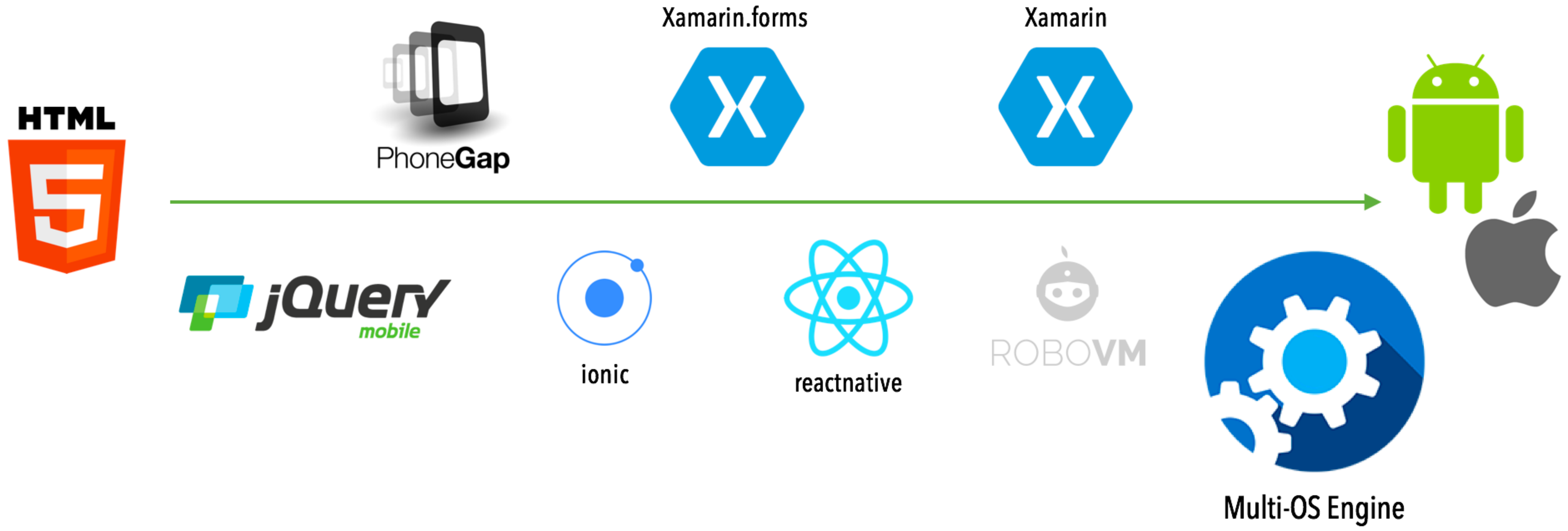
- Develop features using mobile device hardware features: touch, gesture, orientation, graphics, and location services
- Implement client-side code to work with web services

DEVELOPMENT ENVIRONMENT

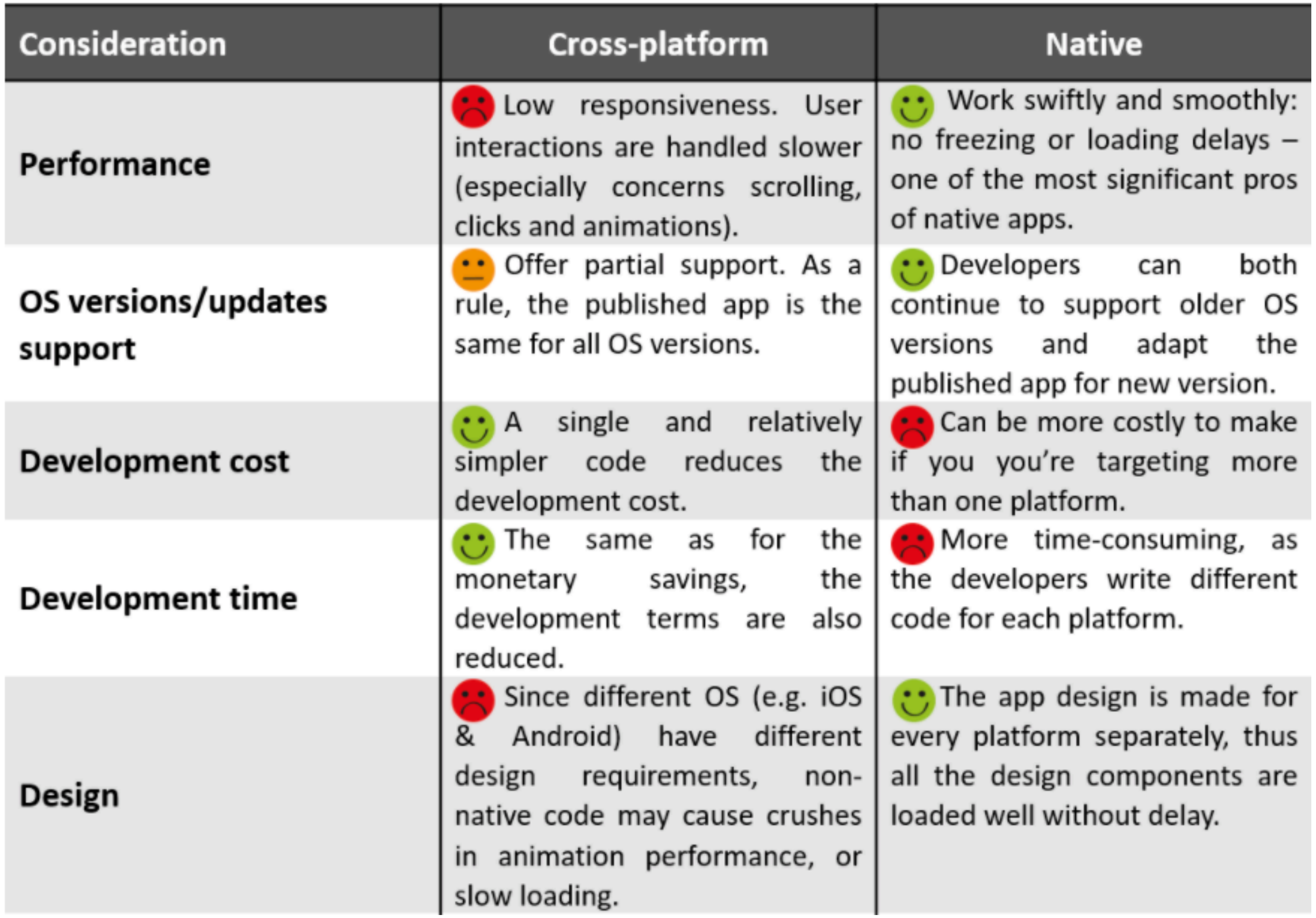
Android Mobile Development
iOS Mobile Development

Native
vs
Hybrid
vs
Web
Mobile App Development
©FBOMBMEDIA

Cross-platform vs native

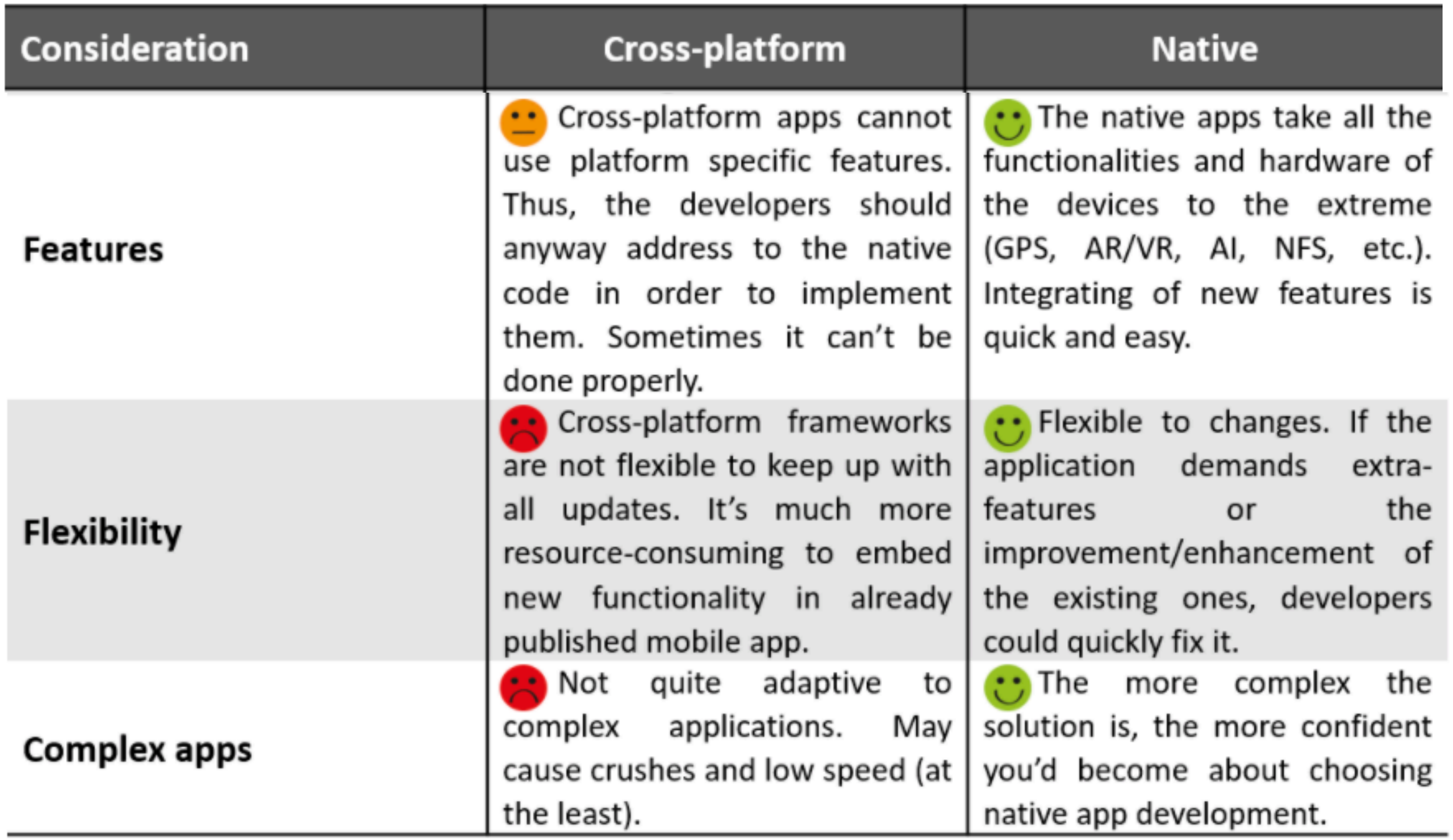


NATIVE VS. CROSS-PLATFORM



Consideration	Cross-platform	Native
Performance	☹ Low responsiveness. User interactions are handled slower (especially concerns scrolling, clicks and animations).	🙂 Work swiftly and smoothly: no freezing or loading delays – one of the most significant pros of native apps.
OS versions/updates support	😐 Offer partial support. As a rule, the published app is the same for all OS versions.	🙂 Developers can both continue to support older OS versions and adapt the published app for new version.
Development cost	🙂 A single and relatively simpler code reduces the development cost.	☹ Can be more costly to make if you you're targeting more than one platform.
Development time	🙂 The same as for the monetary savings, the development terms are also reduced.	☹ More time-consuming, as the developers write different code for each platform.
Design	☹ Since different OS (e.g. iOS & Android) have different design requirements, non-native code may cause crushes in animation performance, or slow loading.	🙂 The app design is made for every platform separately, thus all the design components are loaded well without delay.

NATIVE VS. CROSS-PLATFORM



Consideration	Cross-platform	Native
Features	😐 Cross-platform apps cannot use platform specific features. Thus, the developers should anyway address to the native code in order to implement them. Sometimes it can't be done properly.	🙂 The native apps take all the functionalities and hardware of the devices to the extreme (GPS, AR/VR, AI, NFS, etc.). Integrating of new features is quick and easy.
Flexibility	☹ Cross-platform frameworks are not flexible to keep up with all updates. It's much more resource-consuming to embed new functionality in already published mobile app.	🙂 Flexible to changes. If the application demands extra-features or the improvement/enhancement of the existing ones, developers could quickly fix it.
Complex apps	☹ Not quite adaptive to complex applications. May cause crushes and low speed (at the least).	🙂 The more complex the solution is, the more confident you'd become about choosing native app development.

ANDROID DEVELOPMENT

- Understand Android system architecture, runtime, manifest file
- Understand essential app components: Activity, Service, Broadcast Receiver and Content Provider
- Use Android Studio, Android SDK manager, debugging monitor, and emulator

IOS DEVELOPMENT

- Develop iOS application using Swift programming language
- Use Xcode, Interface Editor, View Controller, emulator and other tools

ANDROID MOBILE DEVELOPMENT

This course uses Android Studio as the main integrated development environment (IDE).

Unless specified otherwise, all Android mobile development in this course shall target for **Android API level 21**, also known as Lollipop.

ANDROID MOBILE DEVELOPMENT

While the instructor mainly uses Mac OS X operating system in this course, students can use Windows or Linux operating systems for Android code development.

Students are expected to configure the environment of their own computers, including the installation of Java Development Kits and the configuration for USB based debugging.

ANDROID MOBILE DEVELOPMENT

Newer Android hardware device (phone or tablet) is recommended but not required for this course.

However, students using an Android device emulator alone will most likely expect sluggish performance and/or decreased productivity.

Not all Android mobile platform APIs are supported in outdated Android hardware devices.

IOS MOBILE DEVELOPMENT

In order to properly learn iOS mobile development, all students taking this course/section is recommended to use a computer with Mac OS X operating system.

Xcode and tools for iOS development are available for Mac OS X operating system only.

IOS MOBILE DEVELOPMENT

Xcode 8.0 or later will be used in this course. All iOS applications will be developed using Swift 4 and iOS 10+.

You are not required to apply for an Apple Developer account ($99 per year) in order to develop and test your iOS mobile app.

However, you'll need a developer account if you want to publish your app to Apple's App Store (not required by this course).

IOS MOBILE DEVELOPMENT

RECOMMENDED!

An iOS 10+ or above device (iPad, iPhone, etc.) is recommended but not required for this course.

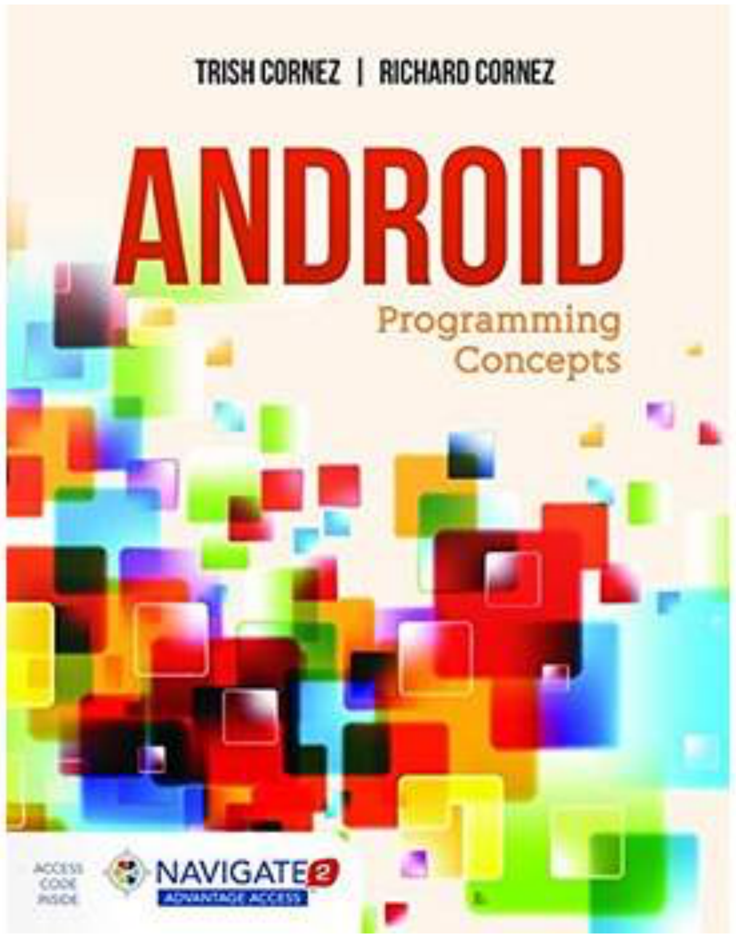
Without an iOS device, you'll still be able to develop your iOS app using iOS device emulator in Xcode.

READINGS

Textbooks

Required Readings

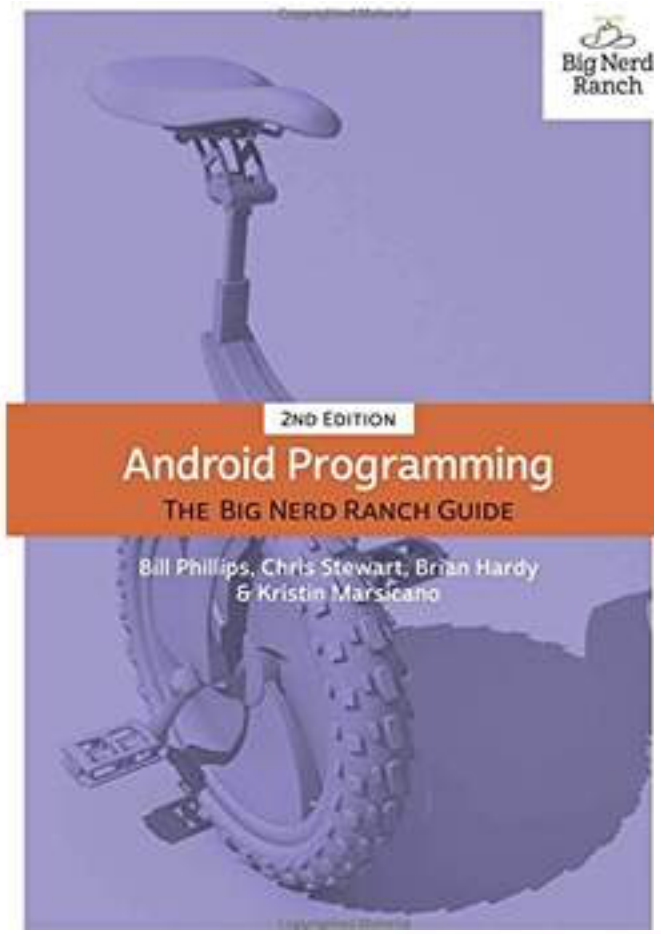
TEXTBOOK



T. Cornez and R. Cornez
Android Programming Concepts
Jones & Bartlett Learning

REQUIRED

TEXTBOOK



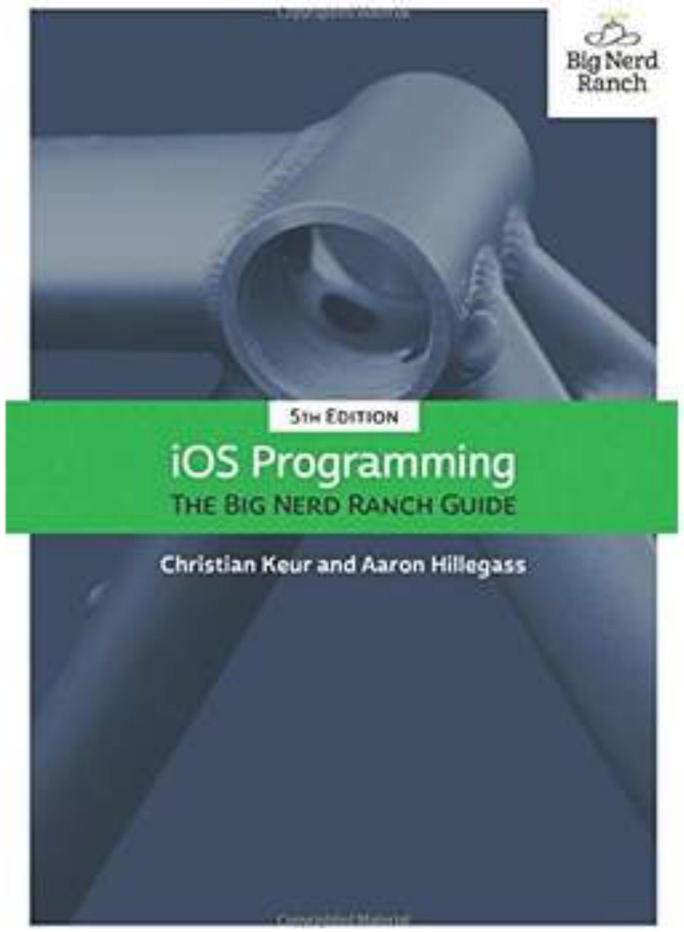
OPTIONAL

B. Phillips et al.
Android Programming:
The Big Nerd Ranch Guide,
2nd Ed.

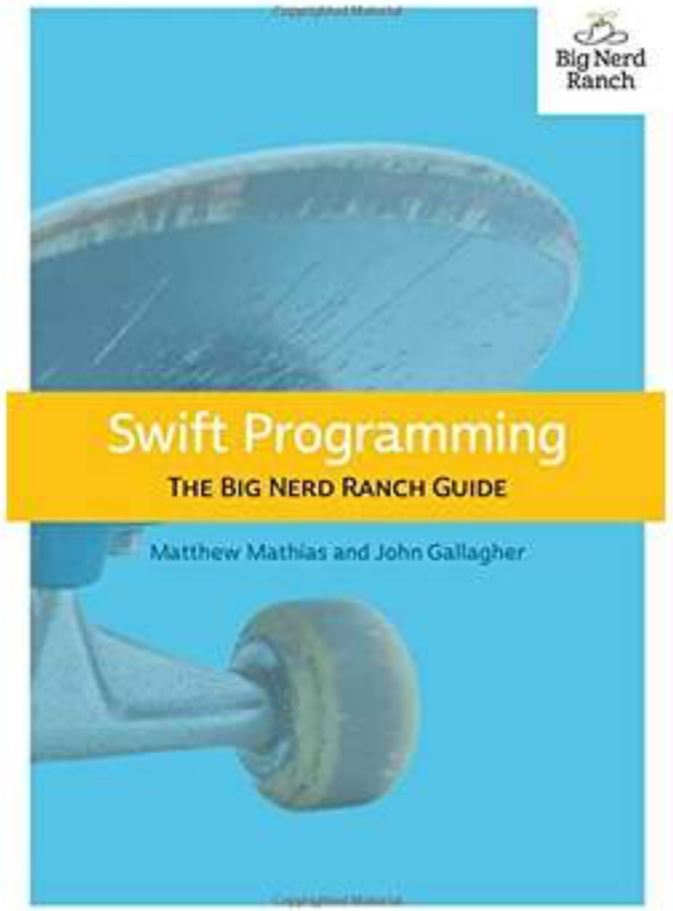
OPTIONAL

J. Annuzzi Jr. et al.
Advanced Android Application Development
Addison-Wesley Professional, 4th ed.

TEXTBOOK



C. Keur and A. Hillegass
iOS Programming: The Big Nerd Ranch Guide OPTIONAL



M. Mathias and J. Gallagher
Swift Programming: The Big Nerd Ranch Guide OPTIONAL

REQUIRED READINGS

Students are expected to go over required readings after each lecture.

The required readings cover the required textbook chapters as well as supplementary learning material provided by the instructor.

The end-of-chapter exercises (if available) are *optional* but some of the exercises may be included in homework assignments.

HOMEWORK

Assignments: 2 + 3

Total: 40 Points

Weighting: 40%

ASSIGNMENT

There are a total of three written assignments and and two programming assignments.

The written part consists of essay questions, multiple choices, true or false, and fill-in-the-blank.

The programming part requires each student's individual effort to come up with coding solution to challenging programming questions.

ASSIGNMENT

All homework assignments including both written and programming parts are submitted electronically. The guideline for submission will be on the first homework assignment.

VIDEO CAPTURE

To get best grade for programming assignment, it's useful to video capture a demo for your program.

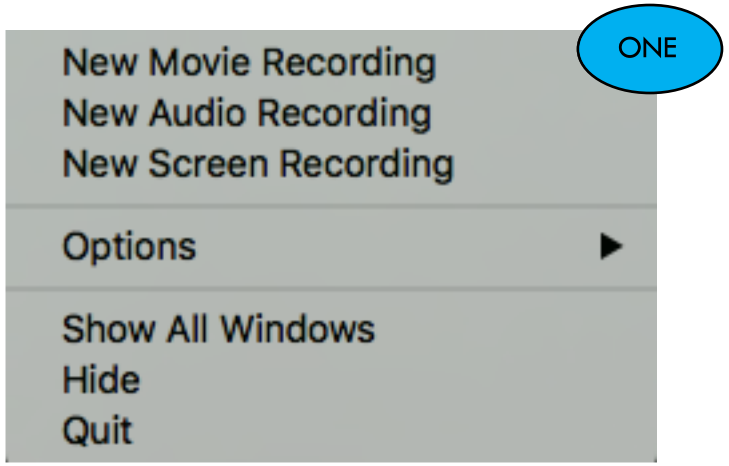
Software You Need:

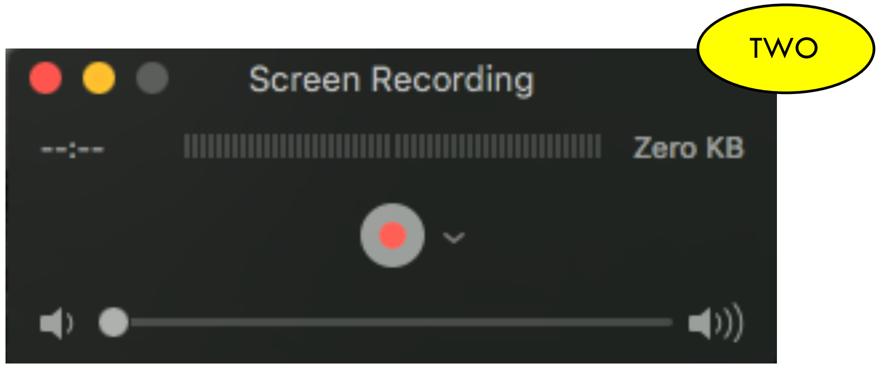
- Quick Time Player
- Vysor Chrome Plug-in

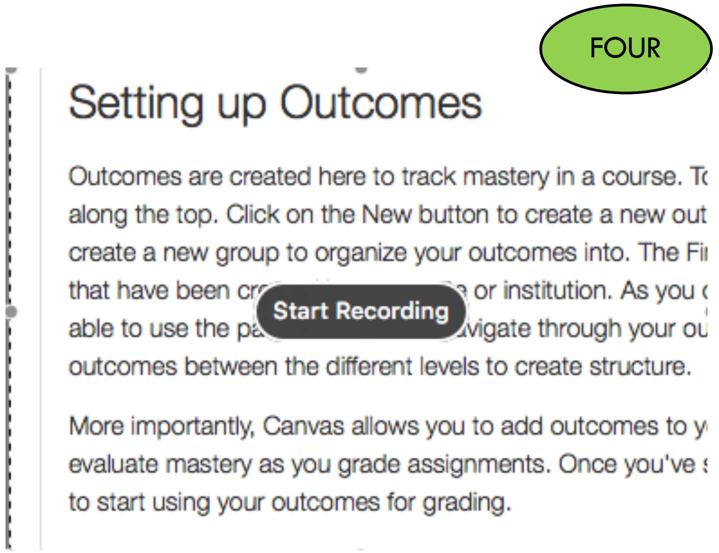
Homework

QUICK TIME PLAYER

Perform Screen Recording REQUIRED



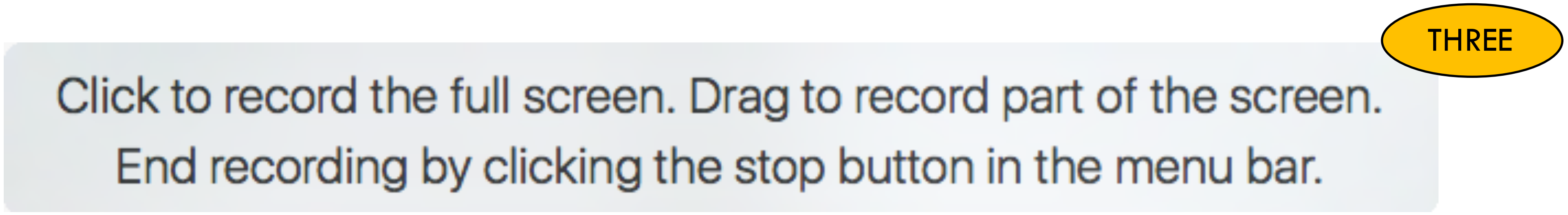




Setting up Outcomes

Outcomes are created here to track mastery in a course. To along the top. Click on the New button to create a new out create a new group to organize your outcomes into. The Fi that have been cr or institution. As you able to use the pa vigate through your ou outcomes between the different levels to create structure.

More importantly, Canvas allows you to add outcomes to y evaluate mastery as you grade assignments. Once you've to start using your outcomes for grading.

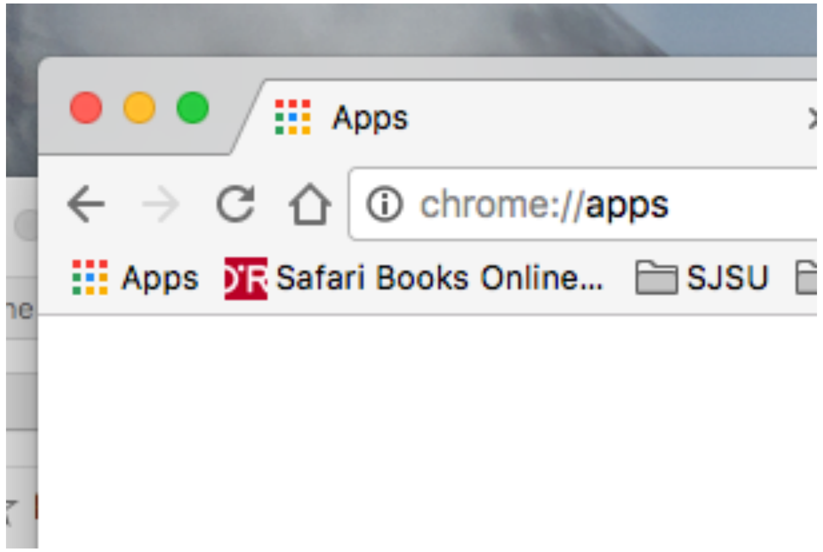


Click to record the full screen. Drag to record part of the screen.
End recording by clicking the stop button in the menu bar.

Homework

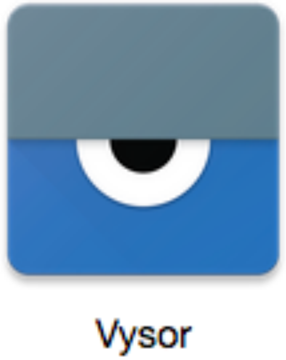
VYSOR

Mirror Device Screen on Desktop



ONE

TWO



SHARE YOUR CODE USING GITHUB

http://www.github.com REQUIRED

GitHub

GitHub
GIT CHEAT SHEET

https://services.github.com/kit/downloads/github-git-cheat-sheet.pdf

FINAL PROJECT

Total: 20 Points

Weighting: 20%

FINAL PROJECT

Final Project can be either Android or iOS application.

Final project is individual based and NOT a group project.

The requirements for final project will be available on Canvas.

FINAL PROJECT

The assessment for final project is based on:

1) business, technical, and quality requirements, and

2) in-class presentation of student’s project.

FINAL PROJECT

Students are allowed to include third-party libraries and sample code in the mobile application for final project but the links for original source code and license information must be fully disclosed and included in source code file(s) for final project submission.

FINAL PROJECT REQUIREMENTS – 15 POINTS

The project requirements have a combined 50% of the final project score (total = 30 points).

Think of each requirement set as a check list. You will be given 5 points for each requirement set. If you miss one item, you will be deducted 0.5 point from that requirement set. You will be awarded no point if the deducted points for a requirement set exceed 5 points.

TECHNICAL REQUIREMENTS (5 POINTS)

- Demonstrate your proficiency in mobile user interface
  - Use model-view-controller model for user interface in your app
  - Implement user interface layout design and handle event messaging
  - Show the changing UI in response to orientation sensor (portrait mode and landscape mode)
- Show your skills in working with mobile device features
  - Demonstrate the use of touch and gesture
  - Work with at least one kind of sensor technologies, such as accelerometer, gyroscope, compass, etc.
  - Use one of the location services: geo-location, geo-coding, etc.

TECHNICAL REQUIREMENTS

- Show your code for data storage
  - Add application state persistence to your app
  - Demonstrate the read/write operations for file storage
  - Handle database operations: create/read/update/delete
- Work with web services
  - Enable single sign-on authentication service using either Google or Facebook sign on service
  - Use at least one web service such as weather report, real-time stock quote, current traffic info, etc.

BUSINESS REQUIREMENTS (5 POINTS)

- Explain the problem(s) that your app is trying to solve
  - What are the underserved and / or overlooked needs?
  - What are the pain points for the end-users?
  - What do you sell?
- Position your app
  - Think of a tagline that describes your product well., e.g., Allstate, you're in a good hand.
  - Describe at least two competitive products that resemble to your product
  - Why is your product better? And in what aspects?
  - How are you going to differentiate your product from the existing solutions?
  - Describe the early adopters for your app., e.g., IT professional living in San Francisco

BUSINESS REQUIREMENTS

- Innovation in your app
  - Does the product show your nature of creativity?
  - Any particular features that are innovative?
  - What's your strategy preventing others from copying your innovative idea?
- Monetize your app
  - How do you make money out of your app?
  - How would you charge your user? For example, ad sponsored, in-app purchase, one-time charge, etc.
  - How do you set your price?
  - How are you going to promote your app?

QUALITY REQUIREMENTS (5 POINTS)

- Deliver the best user experience
  - Does the design of your product look appealing to users?
  - Does your product operate smoothly and seamlessly without manual intervention from users?
  - Are there any show stoppers or fatal bugs?
- Become excellent in engineering design
  - How robust is your implementation as compared to a commercial app?
  - Are you able to provide good technical documentation for your app?
  - Are there any error checking and unit testing methods?

QUALITY REQUIREMENTS

- Develop strategy for product features
  - What are the features you will include in your first release? Why are those features essential?
  - What are the features for your second release and when is it?
  - How are you going to decide which feature to include in your app?
- Streamline your development
  - What are the development tools you will employ to speed up your development cycle?
  - How will you support the release and distribution of your app to different markets?
  - What process / methodology you will adopt to integrate thorough testing in your code development?

PITCH BOOK FOR FINAL PROJECT

Prepare a Pitch Book slide deck that describes your project in terms of the above requirements.

The slide deck shall be at least 10 pages and highlight all the major points with clarity.

EVALUATION – 15 POINTS

The following criteria are used by your instructor to evaluate your final project.

The weighting is 50% of total score for final project.

Even though the criteria for evaluation are subjective, I am providing my assessment guideline here as clearly as possible.

For each assessment area, I expect 80% of students to receive 3 or 4 points, 10% to receive 5 points, and 10% to receive 1 or 2 points.

CLASS PRESENTATION – 5 POINTS

I'll listen to your pitch presentation.

This is the opportunity that you tell me about your mobile app, before I go over your submitted write-up for final project.

You shall present your work with clarity and deliver the message across.

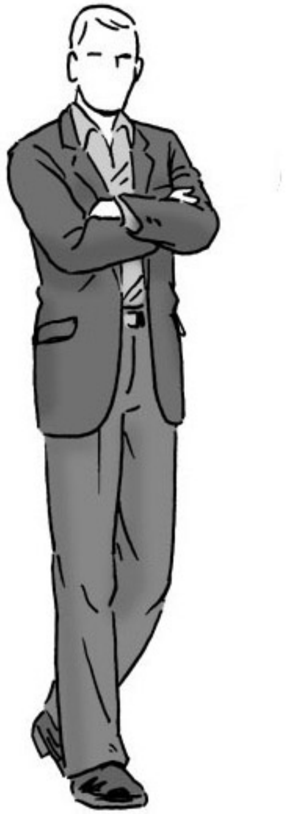
CLASS PRESENTATION

It will be useful if you can develop a tag line so people understand instantly what you are trying to convey.

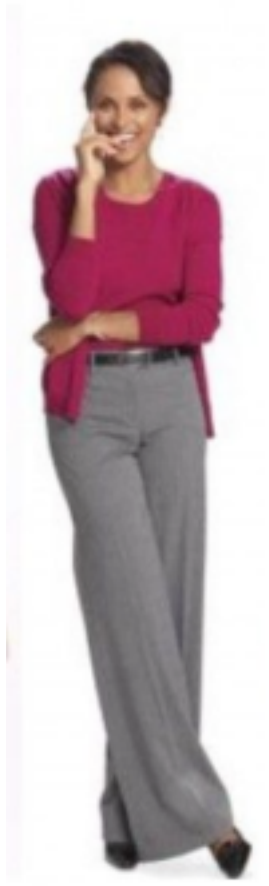
Any technique you developed for public speech will be useful here: use a commanding voice to show you own the stage, keep good pacing of your speech, show good eye contact, speak loudly enough so people can all hear you, and demonstrate great enthusiasm.

DRESS CODE: BUSINESS CASUAL

For Gentlemen:



For Ladies:



TECHNICAL / CODING QUALITY – 5 POINTS

I'll evaluate the quality of your programming skill. Your app shall be compilable and runnable after the source code is cloned.

If additional compile & build instructions are needed, you must provide the step-by-step instruction clearly.

I will look for the comments in your source code in order to understand how well/bad the code is written.

I pay attention to good programming techniques you used in your code as well as how you organize your code.

TECHNICAL / CODING QUALITY

Use of software design pattern in your code will be helpful but not mandatory.

I should not encounter any issues when I execute your app.

I may try to break your app so you shall provide for major corner cases in your app.

I may launch your app, close it and re-launch it again.

In any time, I should not encounter any Application Not Responding (ANR) or similar issues.

USAGE / BUSINESS APPLICATION – 5 POINTS

I'll examine the business application of your app.

In other words, I will be looking for the usefulness of your app as a product.

For example, if you submit your app to Google Play and App Store, how likely will the users download your app?

USAGE / BUSINESS APPLICATION

If your app charges for money, I'll put myself into an end-user's shoe and decide if I will pay for it.

Your app doesn't need to be disruptive, but it should solve the problem(s) that you claimed.

I will be making sense out of your app with regard to its usefulness and applicability.

EXAMS

Mid-term Exam: 15%

Final Exam: 25%

EXAMS

There will be one mid-term exam and one final exam during the class time for this course.

The mid-term exam covers Android mobile development.

The final exam covers both Android and Swift/iOS mobile development.

It is mandatory for the students to take both mid-term and final exams.

EXAMS

Exams cover all class material and concepts taught in this course.

All questions in the exam are written questions, similar to the written part of assignments and consisting of essay questions, multiple choices, true or false, and fill-in-the-blank.

There won’t be any programming questions in the exams.

EXAMS

Both mid-term exams will be available on the day of the exam and on Canvas.

Students are expected to submit their answers electronically on Canvas.

COURSE GRADE

Grading Information

Letter Grades

GRADING INFORMATION

Android Written Assignment #1	5%
Android Written Assignment #2	5%
iOS Written Assignment #3	5%
Android Programming #1	10%
Android Programming #2	10%
Mid-term Exam	15%
Final Exam	25%
Final Project	20%
	100%

Late or missed work will not be accepted.

THE LETTER GRADES

A, A+, A-: ~33%

B, B+, B-: ~50%

COURSE SCHEDULE

Topics / Readings

Deadlines

COURSE SCHEDULE

WK	Date	Topics	Deadlines
1	08/26	Class introduction. Introduction to Android Development	
2	09/02	Android primer. App Components	
3	09/09	Android Activities. Basic User Interface Programming	
4	09/16	Broadcast Receiver. Advanced User Interface	HW #1
5	09/23	Concurrency and Services.	
6	09/30	Sensors and Location.	Programming #1
7	10/07	Networking and Multimedia	
8	10/14	Android Storage. Device Programming.	HW2
9	10/21	Mid-term Exam	
10	10/28	Introduction to Swift Programming	
11	11/04	Advanced Topics in Swift Programming	Programming #2
12	11/11	Basic iOS Programming	
13	11/18	iOS Programming: Building a complete app	HW#3
	11/25	Thanksgiving Week. No Class.	
14	12/02	Android and iOS Mobile Programming Review	
15	12/09	Final Project: Demos / Evaluation	Submission of Final Project
16	12/16	Final Exam 10:00 am – 12:15 pm, MH422	

COURSE SCHEDULE

1
- Class introduction: lecturer, class logistics, expectation, grading.
- Android development: the big picture, building a simple Android application.

2
- Android primer: Android system, NDK, Android Tools, Workflow, Agile Process.
- App components: Activity, Broadcast Receiver, Services, Content Provider.

3
- Activities: Activity, FIFO Stack, Lifecycle, Intents, Fragments, Shared Reference.
- Basic UI: Layout, controls, event handling, progress bar, styles and themes

4
- Broadcast Receiver: send & receive, custom BR, email/SMS/phone calls.
- Advanced UI: MVC, drag & drop, gestures, multi-touches, web view, animation

COURSE SCHEDULE

5
- Concurrency: Java concurrency, AsyncTask, Looper, Handler & HandlerThread
- Services: Local, remote, and global remote services

6
- Sensors: device sensor, reading from motion sensors, sensor simulator
- Location: geo-location, geocoder, geofencing, Google Maps API

7
- Networking: web services, RESTful, RPC, JSON, Protobuf
- Multimedia: image processing, camera, media player, TTS, OpenCV

8
- Storage: content provider, internal storage, SQLite, NoSQL database
- Device: BLE, device programming