CS230 Developing Mobile Apps + CS230 Lab

**Summary:**

This course will cover the basics of programming mobile applications for the Android operating system using Android Studio. The majority of the course (assignments, lab) will involve hands-on coding of mobile apps in Java/Kotlin programming languages. Setting up and using Android Studio IDE and basic Git will be covered in the introduction along with Android OS. Students will learn about mobile application environments and platforms and how to design and develop applications to account for the limited screen size, memory, and access to the internet. Other important concepts to be covered are networking, streaming, sensors (camera, gps, gyroscope, etc), Google Maps, and User Interface (UI) and mobile app specific software engineering patterns, graphics, networking, security, media to create new, real world, practical applications. Development, design, implementation, testing, debugging, and maintaining these applications will also be coveredThe course concludes with monetization and app store optimization strategies. The goal of the course is for each student to develop and release an app by the end of the course as a final project worth 25%. For the final project, students are free to develop using their preferred platform/language (iOS, or multi-platform such as Unity and ), and are not restricted to Android. User experience and user interface patterns for each platform are examined so you can create engaging mobile experiences that take into account content, platform and use cases. Teach basic user interface programming with hardware features provided by a mobile device.

**Outline:**

Android studio, Git

Android OS, Cross-platform app development (React Native, Flutter, Xamarin, Unity, etc)

Languages – Java and Kotlin

Activities, Intents, Fragments.

Layouts, Input and Output, Views, Adaptors, screen flow, skeumorphic vs flat

UI design, Touch Gestures, XML, HCI – major motivations, philosophies, goals. User data driven designs

Sensors (camera, gps, gyroscope, accelerometer, gps), Sensors, Sensor Manager, Accessibility

Photo Editing, 2D graphics, games, OpenGL ES

Google maps, Location, Geocoder

Networking, Web Services, REST, JSON

Android Storage, Database, Content Provider, Synchronization and Replication of Mobile Data

Multithreading, Concurrency, UI Thread, data streaming

Local, Remote, and Global Remote Services, Communications Via Network and the Web, State Machine, Correct Communications Model, online transactions, database schemata

Broadcast Receiver, Pending Intent, SMS

Security and hacking, Active Transactions, Hacking Android

Production – Monetization strategies (in-app purchase, advertising), App store optimization, Updates, Reviews, deployment, maintenance, management.

Enterprise requirements – performance, scalability, modifiability, availability, security.

**Course goals:**

- Analyze current best practices for application development

- Understand and practice methods in topics such as optimization and object-oriented programming

- Gain a deeper understanding of programming principles

- Be exposed to technology and business trends impacting mobile applications

- Be competent with the characterization and architecture of mobile applications.

- Be competent with understanding enterprise scale requirements of mobile applications.

- Be competent with designing and developing mobile applications using one application development framework.

- Understand the unique aspects of mobile application design.

- Work in resource sensitive and resolution variant environments.

- Develop applications with location awareness and hardware sensors.

- Understand the use of a mobile device API.

- Develop applications in a client-server environment.

- 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.

- 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.

- 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

- 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

- 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

- Develop features using mobile device hardware features: touch, gesture, orientation, graphics, and location services

- Implement client-side code to work with web services

- 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

The final project (alone or in pairs) is expected to be released onto the Google Play Store or Apple App Store by the end of the course.

Resources:

**developer.android.com**

**Udacity**

**Textbook**

T. Cornez and R. Cornez, *Android Programming Concepts*, Jones & Bartlett Learning, ISBN 1284070700

*The Swift Programming Language (Swift 4)*, Apple Inc., Available on iBook Store

**Other Readings**

D. Jemerov and S. Isakova, Kotlin in Action, 1st Ed. Manning Publications, ISBN 1617293296

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

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

C. Keur and A. Hillegass, iOS Programming: The Big Nerd Ranch Guide, 6th Ed., ISBN 0134390733

Extracts from the following books will be used in class:

1. Bill Phillips, Chris Stewart, Brian Hardy, and Kristin Marsicano, *Android Programming:*

*The Big Nerd Ranch Guide*, Big Nerd Ranch LLC, 3rd edition, 2017;

2. Rajiv Ramnath, Roger Craw\_s, and Paolo Sivilotti, *Android SDK 3 for Dummies*, Wiley.

The following books (available on Safari Books Online) may be used as references:

1. B. Phillips et al., *Android Programming: Big Nerd Ranch Guide* (as mentioned above);

2. Christian Keur and Aaron Hillegass, *iOS Programming: The Big Nerd Ranch Guide*, 6th

edition, 2016;

3. Valentino Lee, Heather Schneider, and Robbie Schell, *Mobile Applications: Architecture,*

*Design and Development*, Prentice Hall, 2004;

4. Tomasz Nurkiewicz and Ben Christensen, *Reactive Programming with RxJava*, O’Reilly

Media, 2016;

5. Raoul-Gabriel Urma, Mario Fusco, and Alan Mycroft, *Java 8 in Action: Lambdas, Streams,*

*and Functional-Style Programming*, Manning Publications, 2015;

6. Benjamin J. Evans and Martijn Verburg, *The Well-Grounded Java Developer: Vital Techniques*

*of Java 7 and Polyglot Programming*, Manning Publications, 2013;

7. Brian Fling, *Mobile Design and Development*, O’Reilly Media, Inc., 2009;

8. Maximiliano Firtman, *Programming the MobileWeb*, O’Reilly Media, Inc., 2nd ed., 2013;

9. Cristian Crumlish and Erin Malone, *Designing Social Interfaces*, 2nd ed., O’Reilly Media,

Inc., 2014;

10. Suzanne Ginsburg, *Designing the iPhone User Experience: A User-Centered Approach to*

*Sketching and Prototyping iPhone Apps*, Addison-Wesley Professional, 2010;

11. Benjamin Muschko, *Gradle in Action*, Manning Publications, 2014;

**Textbooks:** Introduction to Android Application Development: Android Essentials, 5th Edition (Covers Android 6)

By Joseph Annuzzi, Lauren Darcey, Shane Conder

Published Dec 18, 2015 by Addison-Wesley Professional.

ISBN-10: 0-13-438945-X

ISBN-13: 978-0-13-438945-5

**List of Required Texts / Materials:**

1. Head First Android Development by Dawn Griffiths and David Griffiths publisher:

O'Reilly

2. iOS Programming The Big Nerd Ranch Guide by Christian Keur and Aaron Hillegass 5th

Edition, publisher: Pearson

Additional Readings:

Android Developer’s website: developer.android.com

Apple Developer’s website: developer.apple.com

Reference material:

Textbook:

Simon Ng, Beginning iOS 10 Programming with Swift. AppCoda

http://www.appcoda.com/swift/

Additional, useful references:

The Swift Programming Language. Apple Inc.

Simon Ng, Intermediate Swift and iOS 10 Programming. AppCoda

http://www.appcoda.com/intermediate-swift-programming-book/

**2) Acquired**

**Course Readings**: The required readings will be drawn from a textbook published by Course Technology/Cengage Learning. The author is Thomas J. Duffy. The title of the textbook is Programming with Mobile Applications – Android™, iOS, and Windows® Phone 7 (ISBN: 9781133628132). Students should complete the required readings and solve all problems in the exercise sections before continuing to the next topic.

**Iversen & Eierman MOBILE APP DEVELOPMENT FOR iOS and ANDROID, Edition 2.0**

**Prospect Press**

**Copyright 2018**

**eTextbook:**

**- ISBN: 978-1-943153-27-5**

**Textbook**:

*Android Programming: The Big Nerd Ranch Guide (2nd Edition), Phillips, Stewart, Hardy & Marsicano,*

*Big Nerd Ranch Guides, August 3, 2015, ISBN 978-0134171456*

(Module: app) file here is in the app folder. Its build settings apply only to the app module. If there were another module, then that module would have its own build.gradle file, too. As an example I made a library project with three modules: a library module, a demo app module, and another app module that I plan to use for testing. Each of them have their own build.gradle files that I can tweak.