C S 64A: WRITING APPS FOR THE ANDROID

Foothill College Course Outline of Record

Heading	Value
Effective Term:	Summer 2024
Units:	4.5
Hours:	4 lecture, 2 laboratory per week (72 total per quarter)
Advisory:	C S 1B or 2B.
Degree & Credit Status:	Degree-Applicable Credit Course
Foothill GE:	Non-GE
Transferable:	CSU
Grade Type:	Letter Grade (Request for Pass/No Pass)
Repeatability:	Not Repeatable

Student Learning Outcomes

- A successful student will be able to write many different types of Android apps, making use of diverse aspects such as user interface layout, XML, the Android support library, location-awareness, 2-D and 3-D graphics, and app signing and publishing.
- A successful student will be able to configure an Android SDK emulator and use it to write and debug basic apps that can be uploaded and tested on an actual Android device.

Description

Introduction to programming mobile apps for the Android. Coding topics include the Android SDK for Eclipse, the ADT plugin, XML fundamentals, and a survey of API methods and objects used to control the Android user interface. Concept topics include layouts, activity lifecycles, runtime binding, intents, location awareness, audio, video, OpenGL ES, and monetizing apps.

Course Objectives

The student will be able to:

1. Use the basic tools used by an Android programmer.
2. Configure an Android emulator and a hardware connection to an Android device.
3. Describe the Android development lifecycle.
4. Define XML and give examples of how it is used to express data.
5. Write interactive programs on the Android.
6. Demonstrate the use of activity lifecycles to control an app.
7. Analyze a design's ability to support multiple screen resolutions and natural languages.
8. Design dynamic UIs using fragments and the Android support library.
9. Analyze a design's ability to interact with other apps on the device.
10. Produce location-aware apps.
11. Incorporate network and/or cloud operations into an app.
12. Use layout hierarchies to produce reusable layouts.
13. Explain how audio is used in Android apps.
14. Use photo and video in an Android app.
15. Design 2-D and 3-D graphics into apps using the mathematics of linear and affine transformations through the OpenGL application programmer interface (API).
16. Sign and publish apps.
17. Describe the basics of app monetization and the incorporation of TV, accessibility, and web searching into apps.

Course Content

1. Android development tools
   a. Eclipse integrated development environment (IDE)
   b. Android software development kit (SDK)
   c. Application development tools (ADT) plugin
2. Emulators and devices
   a. Android virtual devices (AVDs)
   b. Connecting Androids to the development platform
   c. USB drivers for Android development
3. Android development lifecycle
   a. "Hello World!"
   b. Running on the emulator
   c. Running on a device
4. XML fundamentals
   a. Trees
   b. Elements
   c. Attributes
   d. Examples
5. Simple interactive programs
   a. View and ViewGroup objects
   b. Layouts using XML and graphics layout editors
   c. Text fields
   d. String resources
   e. Buttons
   f. Activities and manifests
   g. Runtime binding
   h. Intents
6. Activity lifecycles
   a. Callbacks and activity pyramids
   b. Launcher activity
   c. Instantiation
   d. Destroying activities
   e. Pausing, resuming, starting, and stopping activities
   f. Saving and restoring activities
7. Maximizing the audience
   a. Multi-(natural) language support
   b. Multi-resolution support
   c. Multi-version support
8. Dynamic UIs
   a. Fragments
   b. Android support library
   c. Flexible UIs
   d. Fragment communication
9. Interaction with other apps
   a. Sending users to apps
   b. Getting results from apps
   c. Allowing apps to start our activity

10. Location-aware apps
    a. Location manager
    b. Obtaining current location
    c. Displaying addresses
11. Network and cloud operations
    a. Connecting to networks
    b. Managing network usage
    c. Backup API
    d. Cloud messaging
    e. Issuing email
12. Layout hierarchies
    a. ListView
    b. Linear
    c. Relative
    d. Lint
    e. Re-usable layouts
    f. Includes and merges
    g. ViewStub layout
13. Audio
    a. Volume and playback
    b. Audio focus
14. Photos and video
    a. Capturing photos
    b. Simple video record and playback
    c. Controlling the camera
15. Graphics
    a. OpenGL and OpenGL ES
    b. Views
    c. Renderers
    d. 2-D and 3-D shapes
    e. Projection transformations
    f. Motion transformations
    g. Touch events
16. Signing and publishing
    a. Testing for distribution
    b. Icons and labels
    c. Versioning
    d. Obtaining and signing a certificate
    e. Signing up as a developer
    f. User authentication
17. Overview of advanced topics
    a. Bitmaps
    b. TV
    c. Monetizing apps
    d. Accessibility
    e. Connecting devices
    f. Searching the web
    g. Saving user data

Lab Content

1. Familiarization with the Android development platform
    a. Configure the settings of the Eclipse Integrated Development Environment (IDE) for Android development
    b. Use the IDE to create an Android programming project
    c. Write a short introductory program
    d. Connect to an Android device and run the program on the device
2. Writing Android programs
    a. Use the IDE to create an app project that has buttons, string resources, View and ViewGroup objects
    b. Experiment with XML, activities, manifests, and intents in this project
    c. Use the emulator to test the app
    d. Load the app onto an Android device and test it on actual hardware
3. Expanding the app capabilities with lifecycles and more UI options
    a. Use the IDE to create an app that has an activity lifecycle, including some of the following: pause, resume, start, stop, destroy, and restore
    b. Experiment with UI fragments and flexible UIs in this project
    c. Use the emulator to test the app
    d. Load the app onto an Android device and test it on actual hardware
4. Demonstrating inter-app interaction
    a. Use the IDE to create an app that has inter-application interaction
    b. Experiment with sending and getting messages to other apps in this project
    c. Use the emulator to test the app
    d. Load the app onto an Android device and test it on actual hardware
5. Demonstrating location awareness
    a. Use the IDE to create an app that is location-aware
    b. Experiment with the location manager in this project
    c. Use the emulator to test the app
    d. Load the app onto an Android device and test it on actual hardware
6. Building a program that demonstrates layout hierarchies
    a. Use the IDE to create an app that has multiple layout hierarchies
    b. Experiment with ListView, Lint, and ViewStubs in this project
    c. Use the emulator to test the app
    d. Load the app onto an Android device and test it on actual hardware
7. Building a program that demonstrates network operations
    a. Use the IDE to create an app that has network awareness
    b. Experiment network and cloud messaging in this project
    c. Use the emulator to test the app
    d. Load the app onto an Android device and test it on actual hardware
8. Incorporating audio and/or video into app projects
    a. Use the IDE to create an app that has either audio or video (or both) incorporated into its design
    b. Experiment with volume, playback, photo-capture, and/or video control in this project
    c. Use the emulator to test the app
    d. Load the app onto an Android device and test it on actual hardware
9. Building a program around 2-D or 3-D graphics
    a. Use the IDE to create an app that uses OpenGL ES with 2-D and/or 3-D graphics
    b. Experiment with projections, shapes, motion transformations, and renderers in this project

c. Use the emulator to test the app
d. Load the app onto an Android device and test it on actual hardware

Special Facilities and/or Equipment

1. Access to a computer laboratory with Kotlin/Java compilers, Android Studio, Android SDK and ADT plugins, Android devices, and USB connectivity.
2. A website or course management system with an assignment posting component (through which all lab assignments are to be submitted) and a forum component (where students can discuss course material and receive help from the instructor). This applies to all sections, including on-campus (i.e., face-to-face) offerings.
3. When taught via Foothill Global Access on the internet, the college will provide a fully functional and maintained course management system through which the instructor and students can interact.
4. When taught via Foothill Global Access on the internet, students must have currently existing email accounts and ongoing access to computers with internet capabilities.

Method(s) of Evaluation

Methods of Evaluation may include but are not limited to the following:

Tests and quizzes
Written laboratory assignments which include source code, sample runs, and documentation
Project suitable for portfolio
Final examination

Method(s) of Instruction

Methods of Instruction may include but are not limited to the following:

Lectures which include motivation for syntax and use of the Android SDK using Android Studio
Online labs (for all sections, including those meeting face-to-face/on campus), consisting of:
1. A programming assignment webpage located on a college-hosted course management system or other department-approved internet environment. Here, the students will review the specification of each programming assignment and submit their completed lab work
2. A discussion webpage located on a college-hosted course management system or other department-approved internet environment. Here, students can request assistance from the instructor and interact publicly with other class members
Detailed review of programming assignments which includes model solutions and specific comments on the student submissions
In-person or online discussion which engages students and instructor in an ongoing dialog pertaining to all aspects of designing, implementing, and analyzing programs
When course is taught fully online:
1. Instructor-authored lecture materials, handouts, syllabus, assignments, tests, and other relevant course material will be delivered through a college-hosted course management system or other department-approved internet environment
2. Additional instructional guidelines for this course are listed in the addendum of CS department online practices

Representative Text(s) and Other Materials

Griffiths, Dawn, and David Griffiths. Head First Android Development. 2021.

Horton, John. Android Programming for Beginners. 2021.

Philips, Bill. How to Build Android Apps with Kotlin. 2021.

Curriculum and materials provided by Google Education.

Types and/or Examples of Required Reading, Writing, and Outside of Class Assignments

1. Reading:
   a. Textbook assigned reading averaging 30 pages per week
   b. Reading the supplied handouts and modules averaging 10 pages per week
   c. Reading online resources as directed by instructor though links pertinent to programming
   d. Reading library and reference material directed by instructor through course handouts
2. Writing:
   a. Writing technical prose documentation that supports and describes the programs that are submitted for grades

Discipline(s)

Computer Science