C S 63A: DEVELOPING APPLICATIONS FOR IOS

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, 2B or 3B.
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

- Produce clearly written Objective-C code that solves a given problem.
- Write a program that stores user data in between sessions.

Description

An introduction to programming the iPhone and other iOS devices. Covers Swift, Cocoa Touch, and the Model/View/Controller architecture. Students learn the basics of Swift and acquire practical experience with the tools, techniques, and concepts needed to build a basic iOS app from scratch.

Course Objectives

The student will be able to:

1. Use Xcode to develop apps for the iPhone, iPad, and iPod Touch.
2. Analyze a user's needs and create an easy-to-use app that meets those needs.
3. Write a program using the Swift language.
4. Organize a computer program using the Model/View/Controller architecture.
5. Produce clearly written code in an industry standard style.
6. Use many of the Cocoa Touch UI frameworks.
7. Access built-in applications.
8. Use abstraction to solve a specific problem using more general purpose tools.
9. Persist data.
10. Debug and document units of code.
11. Thoroughly test an app on the simulator and on an actual device.

Course Content

1. Using the Xcode integrated development environment
    a. Interface Builder and storyboards
    b. iOS simulator
    c. Debugger
2. Creating a useful app in an efficient manner
    a. User centered design
    b. User stories
    c. The Agile development process
3. Swift
    a. Classes, objects, instances
    b. Instance variables, methods, initializers
    c. Inheritance and events
4. Model/View/Controller
    a. Class diagrams
    b. Target action
    c. Delegation, observation, and notification
5. Programming style
    a. Code reviews
    b. Documentation
    c. Indentation and capitalization
    d. Creating reusable classes
6. Cocoa Touch frameworks
    a. UIKit
    b. File system
7. Multithreading
    a. Application states
    b. Transitions
    c. Notifications
8. Persist data
    a. Table view
    b. Using the device's file system
9. Testing
    a. Simulator
    b. Multi-platform

Lab Content

1. Use the XCode IDE
    a. Use storyboards to construct a user interface
    b. Test an application by simulating it in XCode
    c. Organize projects within an IDE to make submitting labs and switching project environments an orderly process
2. Application development process
    a. Design a user interface
    b. Perform user testing
    c. Incremental development
3. Swift
    a. Write a program that uses a predefined class and that defines a new class
    b. Write a program that uses an initializer
    c. Write a program that uses inheritance
4. Model/View/Controller architecture
    a. Draw a simple class diagram to aid in the design and documentation of a program containing many classes
    b. Use a target action to communicate between the View and the Controller
    c. Delegation, observation, and notification
5. Use good programming style
    a. Participate in code reviews
    b. Write correct and complete documentation

   c. Use consistent and readable indentation and capitalization
   d. Write reusable classes
6. Cocoa Touch frameworks
   a. Write a program that uses the UIKit framework
   b. Deduce how a function or method design impacts the program that invokes it
   c. Use the iOS file system to persist data to the device
   d. Deduce the impact of a function's or method's design on the programs that invoke it
7. Program processes to run in the background
   a. Put program into various application states
   b. Control the program's transition from one state to another
   c. Notify the user when the background process is complete
8. Persist data from one user session to the next
   a. Use the table view to store data on the device
9. Test the app
   a. Simulate the execution in the Xcode IDE
   b. Run the app on different devices/platforms

Special Facilities and/or Equipment

1. Access to a computer laboratory with computers running the latest version of Xcode.
2. Apple devices/systems running the latest version of iOS for developing and testing student apps.
3. 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.
4. 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.
5. 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
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 Swift language and OOP concepts, example programs, and analysis of these programs
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
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. 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

No textbook; course materials developed in partnership with Apple and are fully contained within the course.

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