

Las Positas College  
3000 Campus Hill Drive  
Livermore, CA 94551-7650  
(925) 424-1000  
(925) 443-0742 (Fax)

## Course Outline for CS 16

### MOBILE APPLICATION DEVELOPMENT - IPHONE

Effective: Spring 2018

#### I. CATALOG DESCRIPTION:

CS 16 — MOBILE APPLICATION DEVELOPMENT - IPHONE — 3.00 units

Object-oriented programming in Swift for the iPhone, iPad and related platforms at a beginning to intermediate level. Introduction to the iOS mobile platform. Introduction to Swift syntax and concepts and the iOS application programming interface (API), including: classes, objects, inheritance, protocols, optionals, arrays, dictionaries, and closures; creating user interfaces; using graphics and audio; responding to touch-based user interaction.

2.50 Units Lecture 0.50 Units Lab

#### **Strongly Recommended**

CS 1 - Computing Fundamentals I

#### **Grading Methods:**

Letter or P/NP

#### **Discipline:**

- Computer Science

	<b>MIN</b>
<b>Lecture Hours:</b>	45.00
<b>Lab Hours:</b>	27.00
<b>Total Hours:</b>	72.00

#### II. NUMBER OF TIMES COURSE MAY BE TAKEN FOR CREDIT: 1

#### III. PREREQUISITE AND/OR ADVISORY SKILLS:

**Before entering this course, it is strongly recommended that the student should be able to:**

##### A. CS1

1. Interpret and apply C++ control structures for sequencing, selection and iteration.
2. Interpret and implement programmer-defined functions in C++.
3. Create and interpret expressions involving arithmetic and logical operators;
4. Interpret and apply arrays and simple programmer-defined data structures in C++.
5. Modify and expand short programs that use standard conditional and iterative control structures and functions.
6. Choose appropriate conditional and iteration constructs for a given programming task.
7. Apply the techniques of structured (functional) decomposition to break a program into smaller pieces.
8. Analyze and explain the behavior of simple programs.
9. Describe, interpret and apply the mechanics of parameter passing.
10. Discuss and apply the concept of algorithms in problem-solving processes.
11. Describe and apply effective debugging strategies.
12. Design, implement, test, and debug programs using basic computation, simple I/O, standard conditional and iterative structures, and the definition of functions.

#### IV. MEASURABLE OBJECTIVES:

**Upon completion of this course, the student should be able to:**

- A. Use Xcode fluently for coding and interface creation.
- B. Explain and apply the primary features of the Cocoa Touch framework.
- C. Interpret and apply Swift classes and objects.
- D. Interpret and apply Swift constants, variables and optionals.
- E. Interpret and apply Swift arrays and dictionaries.
- F. Interpret and apply Swift programmer-defined functions and closures.
- G. Create iOS programs of moderate complexity using Swift and Cocoa Touch frameworks.
- H. Explain and apply the model-view-controller (MVC) design pattern.
  - I. Create and manipulate standard view and interface elements within an iOS application.
  - J. Create touch and gesture features within an iOS program.
  - K. Apply asynchronous techniques and callbacks within an iOS application

#### V. CONTENT:

- A. iOS Development Environment
  - 1. iOS (iPhone/iPad operating system) basic features
  - 2. Cocoa Touch (application framework) basic features
  - 3. Using the iPhone simulator
  - 4. Model view controller (MVC) design pattern
- B. Swift – Basic Language Features
  - 1. Data types
  - 2. Optionals
  - 3. Loops
  - 4. Collections
  - 5. Functions
  - 6. Closures
  - 7. Error handling
- C. Swift – Object Oriented Features
  - 1. Classes and objects
  - 2. Inheritance
  - 3. Protocols
  - 4. Polymorphism
  - 5. Structs and Enums
- D. User Interface Concepts and Components
  - 1. Outlets and actions
  - 2. Delegation
  - 3. Standard controls (e.g., buttons, sliders, switches, text fields)
  - 4. Picker views
  - 5. Scroll views
  - 6. Alerts
  - 7. Autolayout
  - 8. Navigation controllers and segues
  - 9. Tab controllers
  - 10. Table views
    - a. Hierarchical/nested tables
  - 11. Map views
    - a. Location geocoding services
- E. Other iOS Application Techniques
  - 1. Timers
  - 2. Custom graphics and animation
  - 3. Audio playback and recording
  - 4. Gestures
  - 5. Introduction to persistent storage
  - 6. Introduction to web requests

#### VI. METHODS OF INSTRUCTION:

- A. **Lecture** -
- B. **Discussion** -
- C. **Projects** -
- D. **Classroom Activity** -
- E. **Student Presentations** -
- F. **Lab** -

#### VII. TYPICAL ASSIGNMENTS:

- A. Create an iPhone application that will maintain a list of user's friends, including names, birthdays, addresses, emails and phone numbers. It should allow the user to enter new friends, and to delete or updating existing ones.
- B. Create an iPhone application that lets the user create and use vocabulary flash cards. The program will allow the user to enter the contents for a each card, including a word and a definition. In practice mode, the program will show the each card's word, asking the user to select showing the definition or advancing to the next card

#### VIII. EVALUATION:

- A. **Methods**
  - 1. Exams/Tests
  - 2. Quizzes
  - 3. Projects
  - 4. Home Work
- B. **Frequency**
  - 1. At least one in-class midterm examination
  - 2. Quizzes approximately every 2 weeks
  - 3. One final project of substantial complexity and/or a comprehensive final exam
  - 4. Several programming assignments of sufficient size and complexity to incorporate all concepts in the course

#### IX. TYPICAL TEXTS:

- 1. Apple, Inc.. *The Swift Programming Language*. Swift 3.1 ed., Apple, Inc., 2015.
- 2. Manning, Jonathon, Paris Buttfield-Addison, and Tim Nugent. *Learning Swift: Building Apps for macOS, iOS, and Beyond*. 2nd ed., O'Reilly Media, 2017.

#### X. OTHER MATERIALS REQUIRED OF STUDENTS:

- A. A portable storage device (e.g., USB drive) is strongly recommended
- B. It is recommended that students have access to an Intel-based Macintosh outside the class.