COURSE OUTLINE OF RECORD

Number: CS G167       TITLE: iPhone Programming with Objective C

ORIGINATOR: Herbert Cohen       EFF TERM: Spring 2011
FORMERLY KNOWN AS:       DATE OF
OUTLINE/REVIEW: 12-18-2010
CROSS LISTED COURSE:       TOP NO: 0707.10

SEMESTER UNITS: 4.0
HRS LEC: 54.0       HRS LAB: 54.0       HRS OTHER: 0.0
CONTACT HRS TOTAL: 108.0
STUDY NON-CONTACT HRS RECOMMENDED: 108.0

CATALOG DESCRIPTION:
This course will cover the fundamentals of iPhone application development using the Objective-C Language and the iPhone SDK (Software Development Kit). The application development process will include: designing for MVC (Model-View-Controller), writing source code, compiling, linking, executing, debugging, and testing. The theory and use of Cocoa and Cocoa Touch, Xcode, Interface Builder, Core Audio/Animation/Data/Location, UIKit (User Interface), GameKit and App design will be discussed in lectures and practiced through lab projects. iPhone applications will be designed and created.

JUSTIFICATION FOR COURSE:

PREREQUISITES:

COREQUISITES:

ADVISORIES:
- Successful completion of at least one course in C programming or one of these Object Oriented (OO) languages, C++, C# (CSharp), or Java, and should understand OO design and development concepts.

ASSIGNED DISCIPLINES:
Computer science

MATERIAL FEE: Yes [ ] No [X] Amount: $0.00

CREDIT STATUS: Noncredit [ ] Credit - Degree Applicable [X] Credit - Not Degree Applicable [ ]

GRADING POLICY: Pass/No Pass [X] Standard Letter [X] Not Graded [ ] Satisfactory Progress [ ]

OPEN ENTRY/OPEN EXIT: Yes [ ] No [X]

TRANSFER STATUS: CSU Transferable[X] UC/CSU Transferable[ ] Not Transferable[ ]

BASIC SKILLS STATUS: Yes [ ] No [X] LEVELS BELOW TRANSFER: Not Applicable

CALIFORNIA CLASSIFICATION CODES: Y - Not Applicable

NON CREDIT COURSE CATEGORY: Y - Not applicable, Credit Course

OCCUPATIONAL (SAM) CODE: C

REPEATABLE ACCORDING TO STATE GUIDELINES: No [X] Yes [ ] NUMBER REPEATS:

REQUIRED FOR DEGREE OR CERTIFICATE: No [X] Yes [ ]

GE AND TRANSFER REQUIREMENTS MET: 

COURSE LEVEL STUDENT LEARNING OUTCOME(S) Supported by this course:
1. utilize Objective-C language to create basic object oriented applications.

2. demonstrate the theory and application of the Model-View-Controller (MVC) methodology for designing applications.

3. utilize the Apple iPhone development environment including Cocoa Touch, Xcode, and Interface Builder.

4. develop applications incorporating iPhone technologies using (but not limited to) Core Audio, Animation, Data, Location, GPS (Global Position Syst.), Audio/Video, Multitasking, and Web Services.

COURSE OBJECTIVES:
1. Utilize Objective-C language to create basic object oriented applications.
2. Demonstrate the theory and application of the Model-View-Controller (MVC) methodology for designing applications.
3. Utilize the Apple iPhone development environment including Cocoa Touch, Xcode, and Interface Builder.
4. Develop applications incorporating iPhone technologies using (but not limited to) Core Audio, Animation, Data, Location, GPS (Global Position Syst.), Audio/Video, Multitasking, and Web Services.

COURSE CONTENT:

LECCTURE CONTENT:

A. Introduction to the iPhone Development environment
   1. Objective-C Language
   2. Cocoa Touch
   3. Xcode
   4. Interface Builder

B. Objective-C Language
   1. Compiling and testing applications
   2. Introduction to Xcode

C. Classes, objects and methods
   1. Theory and description of an object
   2. Defining a class
   3. How objects relate to classes
   4. Class methods and members

D. Data types and expressions
   1. Data types and constants
   2. Operators and expressions

E. Looping and decision making
   1. Loop design and theory
   2. If, if-else, while and do loop design and theory
   3. Conditional operator and switch statements

F. Interfaces and methods
   1. Method structure: theory and design
   2. Self keyword
   3. Allocating and returning objects from methods
   4. Interface theory and design

G. Inheritance and classes
   1. Superclass methods
   2. Overriding methods
   3. Subclasses and new instance variables
   4. Theory of Abstract classes

H. Polymorphism and Dynamic Typing and Binding
   1. Theory of polymorphism
2. Implementation in Objective-C
3. Dynamic Binding and Typing in Objective-C

I. Cocoa and Cocoa Touch
   1. History and background
   2. Mac OS (Operating System) X
   3. Xcode IDE (Integrated Development Environment) and GCC (Gnu C Compiler)
   4. Interface Builder

J. iPhone Project Templates
   1. GUI (Graphical User Interface)
   2. Model
   3. View-Controller

K. iPhone Simulator
   1. Differences between Simulator and actual device
   2. Filesystems
   3. APIs – Application Programming Interfaces
   4. Multi-Touch Technology

L. UI View Controller
   1. Theory and Methods
   2. Windows and Buttons
   3. Views and Rotations
   4. UINavigationController Model

M. UITableView View
   1. Data sources
   2. Delegates

N. Custom UIViews (User Interface), Touch Events, Controls
   1. Custom views with Core Graphics
   2. Touch Events and Multi-Touch Events
   3. UISlider, Tab Bar and Keyboards

O. Core Animation and Open GL ES (Graphics Library Embedded Systems)
   1. Core Animation techniques
   2. Open GL ES Rendering

P. UI Web Kit and UI (User Interface) WebView
   1. Integrating with Safari Browser
   2. HTML5 (HyperText Markup Lang.), CSS (Cascading Style Sheets) and JavaScript

Q. Core Data
   1. Core Data Building Blocks
   2. Schema, Managed Object Model
   3. SQL (Structured Query Language), XML (eXtensible Markup Language), and Flat Files
   4. SQLite (Structured Query Language Lite)

R. Networking and Push Notification
   1. Objective C Classes
   2. URL (Uniform Resource Locator) Request and Connections
   3. Message UI (User Interface) Framework and Sockets
   4. Push Protocol – Sending and Receiving Notifications

S. Game Kit
   1. Peer-to-Peer connectivity
   2. Ad hoc networking
   3. Voice chat

T. New Developments

U. Summary of Technologies
LABORATORY CONTENT:

Lab Content: Developing various applications as projects in support of the lecture material presented each week. Applications to be built may include, but will not be limited to: Welcome App, Tip Calculator, Twitter Search, Quiz Game, Spot Touch Game, Dodge-Ball Game, Sketcher/Painter, Address Book, Route Tracker, Slideshow, Audio Recorder/Player, Discount AirFare Search, Peer-to-Peer Networking, Audio Streaming, Data Base Builder, using Push Notification, using the Game Kit, and using the Web Kit.

METHODS OF INSTRUCTION:

A. Lecture:
B. Lab:
C. Independent Study:

INSTRUCTIONAL TECHNIQUES:

COURSE ASSIGNMENTS:

Reading Assignments

Text
Websites

Out-of-class Assignments

Writing Assignments

Design and development of basic iPhone applications using the set of tools and technologies available for iPhone development.

METHODS OF STUDENT EVALUATION:

Midterm Exam
Final Exam
Short Quizzes
Written Assignments
Objective Examinations
Report
Projects (ind/group)
Problem Solving Exercises
Oral Presentations
Skills Demonstration

Demonstration of Critical Thinking:

Students will be able to make a basic assessment of an iPhone app project to determine the features and requirements to complete this task using Objective-C and Xcode.

Required Writing, Problem Solving, Skills Demonstration:

Design and development of basic iPhone applications using the set of tools and technologies available for iPhone development.

TEXTS, READINGS, AND RESOURCES:

TextBooks:

2. Deitel, Deitel, Deitel, Kern and Morgano. iPhone for Programmers, 1st ed. Prentice Hall, 2010

LIBRARY:

Adequate library resources include:

Comments: