Systems & Network Training

Objective-C programming

A 5 day Hands on training course

Save on expenses. This course is also available on your premises.

Description

A hands on introduction that will allow you to master Objective-C and start using it to write powerful native applications for even the newest Macs and iOS devices! Using The step-by-step approach, will let you get comfortable with Objective-C's unique capabilities and Apple's Xcode 5 development environment. Make the most of Objective-C objects and messaging. Work effectively with design patterns, collections, blocks, foundation classes, threading, Git and a whole lot more. Every session builds on what you've already learned, giving a rock-solid foundation for real-world success!

Key outcomes

By the end of the course delegates will be able to:

- Use Xcode 5.
- Declare classes, instance variables, properties, methods, and actions.
- Use arrays, dictionaries, and sets.
- Expand and extend classes with protocols, delegates, categories, and extensions.
- Use Apple's powerful classes and frameworks.

Training Approach

This structured course uses Instructor Led Training to provide the best possible learning experience. Small class sizes ensure students benefit from our engaging and interactive style of teaching with delegates encouraged to ask questions throughout the course. Quizzes follow each major section allowing checking of learning. Hands on sessions are used throughout to allow delegates to consolidate their new skills.

Details

Who will benefit?
Developers wanting to learn Objective-C

Prerequisites
Software development fundamentals.

Duration: 5 days

Overall rating: ★★★★★

Generic Training	Small Class Sizes	Hands On Training	Our Courseware	Customise Your Course
Generic training compliments product specific courses covering the complete picture of all relevant devices including the protocols "on the wire".	We limit our maximum class size to 8 delegates; often we have less than this. This ensures optimal interactivity between delegates and instructor.	The majority of our courses use hands on sessions to reinforce the theory.	We write our own courses; courseware does not just consist of slides and our slides are diagrams not bullet point text.	Please contact us if you would like a course to be customised to meet your specific requirements. Have the course your way.
"*Friendly environment with expert teaching that teaches the why before the how.*" G.C. Fasthosts	*"Excellent course. The small class size was a great benefit..."* M.B. IBM	*"Not many courses have practice added to it. Normally just the theoretical stuff is covered."* J.W. Vodafone	*"Comprehensive materials that made the course easy to follow and will be used as a reference point."* V.B. Rockwell Collins	*"I was very impressed by the combination of practical and theory. Very informative. Friendly approachable environment, lots of hands on."* S.R. Qinetiq

Objective-C programming

Course Content

PART 1: GETTING STARTED WITH OBJECTIVE-C
The Developer Program: Objective-C, enrolling as an Apple Developer, setting up the development environment, Xcode. Your first project.
OO programming with Objective-C: OO projects, Frameworks, classes and instances, encapsulation, accessors, Inheritance.
OO features in Objective-C: Messages, methods, working with id, nesting messages, method signatures and parameters. allocating and initializing objects.
Using Xcode: Xcode, source code control, git and Xcode, Using a Remote Repository.
Compiler Directives: Projects, Compiler Directives, Prefix headers, main.m, .h files.
PART 2: OBJECTIVE-C BASICS
Messaging in a Testbed App: Setting Up the Testbed Apps, Adding a Text Field and Connecting It to Your Code, Sending a Message to the Text Field, Reviewing the Message Syntax.
Declaring a Class in an Interface File: Xcode and classes, class hierarchies, Superclass methods, declaring Classes, forward references.
Declaring Instance Variables in an Interface File: Context, Creating an Instance Variable with id, What Happens When Execution Stops, dynamic binding, Creating an Instance Variable for with the Class Name and with a Superclass Name, instance variable visibility.
Properties in an Interface File: Interface Variables vs Properties, Declared Properties, Using Attributes. Implementing Properties. @synthesize, @dynamic.
Methods in an Interface File: Methods in a Class, class and instance methods, Method declaration, returning complex data structures from Methods.
Actions in an Interface File: Actions, Actions in OS X and iOS, disconnecting actions.
Routing messages with selectors: Receiver and selector objects in messages, Objective-C Runtime, SEL and @selector (), performSelector, NSInvocation, testing whether an Instance can respond to a selector.
Building on the Foundation: The Foundation Framework, Foundation Classes, Foundation Paradigms and Policies; Mutability, class clusters, notifications.
Defining a Class in Implementation Files: Projects, dynamic typing, creating a new App, implementing a method, expanding Classses with init Methods.
Organizing Data with Collections: Collecting Objects, Property Lists, Runtime, comparing the Collection Classes, Creating a Collection, Objective-C Literal Syntax, Enumerating collections, Testing Membership in a Collection, Accessing an Object in a Collection.
Managing Memory and Runtime Objects: Managing objects in memory, managing reference counts manually and with ARC, variable qualifiers, variable autorelease.
PART 3: EXPANDING AND EXTENDING CLASSES
Protocols and Delegates: Subclassing, Protocols, Delegates, Looking Deeper Inside Protocols.
Categories and Extensions: Comparing categories and protocols, categories vs subclasses, working with categories, class extensions, informal protocols.
Associative References and Fast Enumeration: Objective-C 2.0 Time-Saving Features, Extending Classes by Adding Instance Variables (Sort of), Using Fast Enumeration.
Blocks: Revisiting Blocks, Callbacks, Blocks, Exploring Blocks in Cocoa, Cocoa Blocks and Memory.
PART 4: BEYOND THE BASICS
Handling Exceptions and Errors: Exception and Error classes: NSException, NSError, Identifying exceptions, throwing exceptions, catching exceptions.
Queues and Threading: Getting Started with Concurrency, Introducing Queues, Dispatch Sources, Using Dispatch Queues.
Working with the Debugger: Logging Information, Console Logs, NSLog, Smart Breakpoints, enhancing breakpoints with messages.
Using Xcode Debug Gauges for Analysis: Debug Gauges, Monitoing CPU and memory utilization, monitoring energy, Using Instruments.
PART 5: OPTIONAL TOPICS
C Syntax Summary: Data Types, Control Structures.
Apps, Packages, and Bundles: Project Bundles, Iproj Files, Asset Catalogs, plist Files, Precompiled Header Files (.pch).
Archiving and Packaging Apps for Development and Testing: Archiving.