

This course teaches the fundamentals of programming in Objective-C, the language used to develop iPhone and iPad applications. This course covers Xcode, the integrated development environment used to build Objective-C applications. Primitive and reference variables are covered as well as arithmetic operators, conditional processing, looping, and functions. The object-oriented nature of the language is presented as well as the Foundation framework. An introduction to the Cocoa framework and a discussion of a basic user interface for mobile devices are included. You will use Xcode built-in iPhone and iPad simulators for the exercises and demonstrations in this course.

Course Objectives:

- Learn how to use Xcode for developing Objective-C programs
- Learn how to define primitive and reference variables.
- Understand the use of arithmetic operators.
- Know how to write conditional and looping statements.
- Learn how to write and call a function.
- Understand the object-oriented features of Objective-C.
- Learn how to create objects and methods.
- Understand important classes in the Foundation framework.
- Learn how to build a basic user interface using the Cocoa framework.

Audience: Experienced programmers.

Prerequisites: Minimal programming experience; including some object-oriented and C++ or C# experience.

Number of Days: 4 days

1 Objective-C Overview The Xcode IDE Projects Objective-C History Role in Mobile Device Applications	3 Arithmetic Arithmetic Operators Addition and Subtraction Multiplication, Division, and Modulus Shorthand Notation Typecasting Math Library Functions pow arc4random
2 Variables Numeric Variables Numeric Representations Integers Unsigned Integers Floating Point Constants Nonnumeric Variables Char Boolean Variable Scope	4 Conditional Logic and Looping Conditional Statements Basic if Statement if else Statement if else if Statement switch Statement The Ternary Operator Looping Statements

	while Statement		Runtime Identification of Objects
	do...while Statement		Arrays
	for Statement		The “Classic” C Array
5	Functions		Defining the array
	Purpose		Processing the array
	Declaration		Foundation Framework Arrays
	Header		NSArray
	Body		NSMutableArray
	Calling a Function	10	Foundation Framework
	Passing Parameters by Value		Strings
	Passing Parameters by Reference		Date/Time
	Functions vs. Methods		Numbers
6	Object-oriented Programming		Collections
	According to Objective-C		NSString
	Object-Oriented Programming (OOP)		Replacing char*
	How Objective-C Implements OOP		Initialization
	Encapsulation of Member Variables		NSString Methods
	Class Definition: Interface		NSNumber and NSNumberFormatter
	Class interface and @property		Replacing int, float, and more
	Method vs. Function Syntax		Initialization
	Static Variables and Class Methods		NSNumber and NSNumberFormatter
	The Class Implementation and @synthesize	11	Collections
	Creating an Object from a Class		Arrays
	Sending Messages to Objects		Sets
	The Role of Methods		NSSet
	Visibility of Variables		NSMutableSet
	The Constants nil and NULL		How Duplicate Elements are Detected:
	Reference Variables		isEqual and hash
	The * and & Operators		Dictionaries
7	Inheritance	12	Memory Considerations
	Purpose		Memory Management in Objective-C
	Implementing Inheritance		Manual Retain-Release
	The Root Class		Automatic Reference Counting
	Creating and Processing The Subclass		Garbage Collection
	NSObject Class		Strong and Weak
	Memory Acquisition		Release and Retain
	alloc		Atomic and Nonatomic
	init		Automatic Reference Counting
	Method Override	13	Cocoa Framework
	Where’s Abstraction?		Purpose of Cocoa
8	Polymorphism		User Interface
	Purpose		Using Interface Builder to Create User Interfaces
	Polymorphism in Objective-C		The XIB File
	Placing Objects in Collections		IBAction and IBOutlet
	Using NSMutableArray		



Drag and Drop with the Connections
Inspector
Creating a Simple Mobile Application
Creating the Project
Adding Properties and Synthesized
Methods
Building the User Interface
Testing the Application