

Course Description: Understanding the Objective-C programming language is critical to becoming a successful iPhone developer. This class is designed to teach you a solid foundation in Objective-C 2.0 from the ground up. This course does not require or assume any background in Objective-C, C, C++ or Object Oriented Programming (OOP) in general. You will learn how to enter, develop, and debug your programs under Windows OS X using the Xcode development tool. In the second part of the course, students will learn how to write iPhone Applications with the SDK!

Who Should Attend? This course is for programmers who want to write applications for Windows OS X, the iPhone, the iPod Touch, or the iPad. Some prior programming experience, preferably with an object-oriented programming language.

Prerequisites: Students should have some prior programming experience, preferably with an object-oriented programming language.

Benefits of Attendance: **Upon completion of this course, students will be able to:**

- Define new classes and write programs in Objective-C
- Write subclasses, categories, and understand the principles of inheritance
- Understand how dynamic typing works and the principle of polymorphism
- Use the Foundation Framework classes to work with numbers, strings, collections, and files
- Archive and copy objects and to thoroughly understand the principles of memory management

- Get started writing applications for the iPhone, iPod Touch, and iPad
- Write iPhone and iPad applications
- Use XCode to manage the application development project, and to compile, edit, and debug your applications
- Run your application on a device
- Use Interface Builder to design a user interface
- Integrate iPhone interface elements like buttons, switches, and sliders into your application
- Work with views and view controllers
- Work with navigation controllers, tables, and pickers
- Integrate multimedia into your applications to create sound effects, and to play audio and video clips
- Respond to touch controls and gestures
- Manipulate images
- Perform simple animation
- Save and restore your application's data

Getting Started in Objective-C

1. What is Objective-C?
2. Your First Program
3. Steps for Using Xcode
4. What is a Class, Object, or Method?
5. Defining a Class to Work with Fractions
6. Accessing Instance Variables

Data Types and Expressions

1. Basic Data Types
2. Arithmetic Expressions
3. Defining a Calculator Class

Loops and Making Decisions

1. The for statement
2. The while statement
3. The do statement
4. The if Statement
5. The switch Statement
6. The Conditional Operator and Boolean Variables

More on Classes

1. Properties, Synthesized Accessor Methods, and the dot Operator
2. Multiple Method Arguments
3. Passing Objects as Arguments
4. Local Variables; The self Keyword
5. Allocating and Returning Objects in Methods

Inheritance, Categories, and Protocols

1. Inheriting Methods and Instance Variables
2. Subclasses: Extension through Inheritance
3. Classes Owning Their Own Objects; Overriding Methods
4. Categories
5. Protocols

Polymorphism, Dynamic Typing and Dynamic Binding

1. Polymorphism; Static Typing
2. Dynamic Typing and Dynamic Binding

The Preprocessor and Underlying C Language Features

1. Preprocessor
2. Arrays
3. Functions
4. Structures
5. Pointer
6. Unions
7. Miscellaneous Language Features

Numbers and Strings

1. Number Objects
2. String Objects
 - Collections
1. Array Objects
2. Fast Enumeration
3. Sorting Arrays
4. Dictionary Objects
5. Set Objects

Collections

1. Array Objects
2. Fast Enumeration
3. Sorting Arrays
4. Dictionary Objects
5. Set Objects

Working with Files

1. Managing Files and Directories: NSFileManager
2. Working with Paths: NSPathUtilities.h

Memory Management

1. The Autorelease Pool
2. Reference Counting
3. Summary of Memory Management Rules
4. Garbage Collection

Copying Objects

1. The copy and mutablecopy Methods
2. Shallow Versus Deep Copying
3. Implementing the NSCopying Protocol
4. Copying Objects in Setter and Getter Methods

Archiving Objects

1. Archiving with XML Property Lists
2. Archiving with NSKeyedArchiver
3. Writing Encoding and Decoding Methods
4. Using NSData to Create Custom Archives
5. Using the Archiver to Copy Objects

Writing Your First iPhone Application

1. Introduction to Cocoa Touch
2. The iPhone SDK
3. Your First iPhone Application

Web-based vs. Native Applications

1. The Difference Between the Two
 - Cocoa, Cocoa Touch and the iPhone SDK
1. Definitions

iPhone OS Frameworks

1. What is a Framework?
2. Overview of iPhone SDK Frameworks

iPhone Application Development Overview

1. Overview of the Tools and Process
 - Some Quick Reviews
1. Objective-C
2. Protocols
3. Memory Management

Writing Your First iPhone Application

1. Outlets and Actions
2. Introduction to Interface Builder
3. Using the iPhone Simulator
4. Extending your Application: Adding an Application Icon

Simple Debugging Techniques

1. Using the Preprocessor
2. NSLog and the Description Method
3. Using Breakpoints

Running Your App on a Device

1. The iPhone Provisioning Portal
2. Developer Certificates, App IDs, Device IDs, Provisioning Profiles
3. XCode's Organizer Window

Application Execution and MVC

1. Applications and Events
2. The Model-View-Controller Design Pattern

Screen Geometry

1. The iPhone and iPad Coordinate System
2. The frame, bounds, center, and origin
3. Supporting Structures and Functions
4. Applications and Events

Understanding Views and View Controllers

1. Properties of Views
2. Subviews and Superviews
3. Creating a View Controller
4. Multiple View Controllers
5. Creating Views Programmatically
6. Understanding a View-Based Application
7. Manipulating Views: Moving and Hiding
8. UIImageView – Working with Images
9. Adding Simple Animation
10. Handling Device Rotation and Control
Resizing and Repositioning