

This course is designed for novice programmers as well as for programmers new to the Objective-C programming language. Topics covered will include both the general theoretical concepts of programming - with a focus on Object Oriented programming - as well as the specifics of Objective-C's code syntax. An in-depth exploration of Xcode - Apple's IDE for developing iOS Apps - will also be an integral part of this class as students use it to write many programs that demonstrate the concepts being taught. Students will gain a solid foundation and understanding of how Objective-C works, a strong familiarity of working with Xcode, and be ready to start learning iPhone and iPad App development.

### Course Objectives:

- Write Objective-C programs with variables, primitives, conditionals, loops, and using other building blocks of programming.
- Work with Collection Objects such as NSArray, NSDictionary, and NSSets, and use of Fast Enumeration.
- Create and use custom user-defined Classes.
- Apply Objective-C's Object-Oriented messaging, Inheritance, Polymorphism, Dynamic Binding and Dynamic Typing.
- Explore the Foundation Framework's classes and data structures.
- Write programs that interact with files and the file system

**Audience:** Beginner programmers or programmers new to Objective-C language.

**Prerequisites:** Some programming experience is helpful but is not mandatory.

**Number of Days:** 3 days

### 1 Working with Variables and Data Types

Writing and Running basic programs  
Working with the Console  
Integers  
Floats  
Doubles  
Booleans  
Chars  
NSString

### 2 Arithmetics and Expressions

Addition, Subtraction, Multiplication, Division, Modulus  
Precedence of Operators  
Type conversions through mixed operations  
Type-casting  
Assignment and Comparison operators  
Intro to Math Library functions

### 3 Loops – Automatic Repetition

Addition, Subtraction, Multiplication, Division, Modulus  
Precedence of Operators  
Type conversions through mixed operations  
Type-casting  
Assignment and Comparison operators  
Intro to Math Library functions

### 4 Conditionals & Decision Making

The If statement  
The If-Else statement  
The If-Else-If statement  
Nested Conditionals  
Compound Relationals  
Boolean Variables  
The Switch Statement

5	<p><b>Custom Classes</b>          Creating custom Classes - using @interface          Implementing custom classes - using @implementation          Creating Instance Variables          Creating Instance Methods          Understanding Getters &amp; Setters          Instantiating Objects from Classes          Executing and Messaging Methods onto Objects          Using Self          Declaring Properties and synthesizing Accessor methods</p>	10	<p>File-Paths and NSURL  <b>Introduction to Cocoa Touch &amp; iPhone App Development</b>          Creating the first iPhone App: "Hello World"          Working with Interface Builder          Creating GUI's          Using ViewControllers          Overview of the View Lifecycle          About IBOutlets and IBActions          Creating the second iPhone App: going interactive</p>
6	<p><b>Understanding Inheritance</b>          Root Class, Super-class, Subclass          Sharing Variables and Methods through Inheritance          Locating Methods in Classes          Class Extention through Inheritance          Overriding Methods          Using Super          Class and Object Ownership          Memory Allocation          Variable Scope</p>		
7	<p><b>Polymorphism, Dynamic Binding &amp; Typing</b>          Reusing method names in different classes          Runtime Dynamic Typing and Binding          Static Typing          Runtime Querying of Objects and Classes          The ID type</p>		
8	<p><b>The Foundation Framework</b>          Number Objects          String Objects          Array Objects          Dictionary Objects          Set Objects          Number Objects</p>		
9	<p><b>Working with the File System</b>          The NSFileManager          Working with Directories          Reading, writing and copying files.          Working with NSData</p>		