Programming in *Swift 4.0*

Introducing Swift

Overview

With ***Swift***, Apple has introduced a fast and innovative new programming language aimed specifically at iPhone, iPad and OS X App Development. Modern and concise, Swift represents a major departure from Objective-C in both structure and syntax as it introduces many new programming concepts. This course will thoroughly cover all the major features of Swift, from basic variable and constant declarations to working with Collection Objects, Optionals, Tuples, Closures, Variadic and Curried Functions, Control-Flow statements, and the creation of custom Swift Classes, Protocols, Generic Functions and Types, Error Handling and much more. The course is aimed at creating a solid foundation in Swift programming so as to easily enable moving towards iPhone, iPad and OS X App Development using Swift.

NOTE: This class is recommended as the prerequisite/companion class to the 5-day ***iOS App Development Using Swift*** class also offered by us.

Course Objectives

In this course you will learn how to:

• Use Swift's unique **Optionals**, using Optional Binding, Optional Chaining, Forced Unwrapping and Implicitly Forced Optionals

• Create Swift **Functions**, including Variadic Functions, *inout* Parameter Value Functions, Curried Functions, Nested Functions, Functions Returning ***Tuples***, and more.

• Use Swift **Closures**, with Type Inferrence, Shorthand Argument Names, and Trailing Closures syntax

• Create Swift **Classes** using Stored, Computed, and Lazy Properties, as well as Property Observers

• Create Swift **init** methods including Designated and Convenience Initializers, understand Initializer Chaining and Initializer Delegation, use the ***Self*** keyword

• Create Instance and Type Methods as well as Type Properties

• Use Inheritance to create Subclasses, override Super-class Methods & Properties**,** prevent Overrides

• Use Swift's Standard Library Protocols as well as create custom Protocols, specifying required and optional Properties and Methods, as well as use multiple/chaining Protocol Inheritence

• Use Swift **Generics**, create Generic Functions, Generic Types, and restrict Generics using Protocols

• Create Swift **Structures**, declaring Struct Properties and Methods, work with Default and Memberwise Init's, create Mutating Functions, and Structs containing other Structs

• Use Swift's **Error Handling** mechanism, handling built-in System errors using the **Try-Catch-Defer** block, create and *Throw* custom user defined Errors using the **Error** Protocol

Target Audience - Beginners and experienced programmers who are not familiar with ***Swift***.

Course Duration – 3 days

Lab Setup / Required Tools:

• Mac computers running OS X 10.12 ("Sierra") or greater.
• Xcode 9 – free download from https://developer.apple.com/xcode/downloads/

Course Outline

I. Creating and Running a Swift Project in Xcode

1. Xcode IDE overview
2. Creating and using Playgrounds
3. Building and testing programs – the workflow

II. Variables, Constants, and Swift Data Types

1. Variables and Constants
2. Data Types: Ints, Floats, Doubles, Bools and Characters
3. Strings – Literals, Mutability, Interpolation, Concatenation
4. Type Inference & Type-Safety
5. Type Casting
6. Unicode characters in Variable declaration

III. Working with Collection Objects

1. **Arrays**
   i. Array Literals
   ii. Creating and Initializing Arrays
   iii. Accessing & Modifying Arrays
   iv. Iterating over Arrays
   v. Assignment and Copy Behavior for Arrays
2. **Dictionaries**
   i. Dictionary Literals
   ii. Initializing
   iii. Modifying Dictionaries
   iv. Iterating over Dictionaries
   v. Assignment and Copy Behavior for Dictionaries
3. **Sequence Operations**
   i. map
   ii. filter
   iii. reduce

IV. Control-Flow

1. **Conditionals**
   i. IF and Ternary Statements, Compound Relationals
   ii. Switch Statement – switch with Strings, Range-Matching, Tuples, Value Bindings, WHERE clauses

iii. Control Transfer Statements – Continue, Break, Fallthrough, Return
iv. Guard Statement

2. Loops

i. FOR loops, FOR-IN loops, FOR-Condition-Increment
ii. WHILE loops, REPEAT-WHILE loops

V. Writing Classes in Swift

1. Properties

i. Stored Properties – Variable or Constant
ii. Optional Properties
iii. Computed properties – Read/Write, Read-Only
iv. Lazy Properties
v. Property Observers
vi. Type Properties

2. Methods

i. Instance Methods
   - init methods - Designated Initializer, Convenience Initializer, Initializer Delegation, Initializer Chaining
   - Customizing Initialization

ii. Local and External Parameter Names
iii. The *Self* keyword
iv. Type Methods

VI. Inheritance

1. Base Classes, Creating Subclasses
2. Overriding Methods & Properties
3. Preventing Overrides

VII. Archiving, Serialization, and Key-Value Coding

1. The Codable Protocol
2. Using the JSONEncoder Class
3. Creating KeyPaths for Property References
4. Creating KeyPaths for Function References

VIII. Optionals

1. Forced Unwrapping
2. Optional Binding
3. Implicitly Unwrapped Optionals
4. Optional Chaining

IX. Functions

1. Basic, Argumented, Variadic Parameters Functions, *inout* Parameter Value Functions, Curried Functions, Nested Functions, with Default parameter values, with Return types, Returning Multiple/Tuple Types
2. Constant and Variable Parameters
3. Function Types

4. Functions Returning functions

X. Structures

1. Properties & Methods
2. Memberwise Initializers
3. Mutating Functions
4. Structures vs. Classes
5. Asking Questions about Classes

XI. Enumerations

1. Declaration & Creation
2. Matching with Switch Statements
3. Associated Values
4. Raw Values

XII. Generics

1. Generic Functions
2. Generic Types
3. Type Parameters
4. Extending Generic Types
5. Associated Types

XIII. Protocols

1. Protocol Declarations and Syntax
2. Property and Methods Requirements
3. Requiring Initializers
4. Protocols as Types
5. Swift Standard Library Protocols
6. Protocol Inheritance

XIV. Closures

1. Closure Expressions & Syntax
2. Closure Context Type Inferrence
3. Shorthand Argument Names
4. Trailing Closures

XV. Debugging

1. Creating Breakpoints
2. Advanced Breakpoints
3. Debugging in LLDB

XVI. Error Handling

1. Understanding the Error Protocol.
2. Creating a custom Error Enum
3. Throw-Try-Catch-Defer Statements

**Note this course may be customized as per the client's needs upon request*