

**Introduction to the Swift Programming Language**

**Course Number:** SWFT-100
**Duration:** 3 days

**Overview**

Accelebrate's Swift programming training course teaches Apple developers the basics of the Swift language, including the Swift syntax and object-oriented programming concepts, giving teams a solid foundation to build on. Attendees learn how to use Swift to build mobile apps and create software solutions. After taking the course, students confidently build their own apps and software solutions.

**Prerequisites**

* All attendees must have some previous programming experience (any language)
* Familiarity with object-oriented programming concepts is recommended but not required
* Attendees must have some experience using the macOS operating system (launching applications, working with files, etc.)

**Materials**

All Swift training students receive courseware, a lab manual, and a related textbook.

**Software Needed on Each Student PC**

* Mac running the current or immediately previous version of macOS, with 8 GB RAM or more
* The latest version of Xcode (available for free from the Apple App Store)

**Objectives**

* Understand the purpose and benefits of Swift
* Gain experience using Swift’s data types and standard library
* Learn the proper use of optionals
* Implement a variety of user-defined types in Swift
* Learn about error handling and techniques in Swift
* Understand how to write Swift code that can interoperate with existing code written in C and Objective-C

**Outline**

* Introduction
	+ Purpose of Swift
	+ Evolution of Swift
	+ Goals of Swift
	+ Execution Environment
	+ Swift Package Manager
* Fundamentals
	+ Files and Initialization
	+ Statements
	+ Constants and Variables
	+ Type Annotations
	+ Numeric Types
	+ Booleans
	+ Strings and Characters
	+ Tuples
	+ Conditional Statements
	+ Loops
* Optionals
	+ Introduction
	+ Forced Unwrapping
	+ Optional Binding
	+ Optional Chaining
	+ Implicitly Unwrapped Optionals
* Functions
	+ Introduction
	+ External Parameter Names
	+ Default Parameter Values
	+ Variadic Parameters
	+ InOut Parameters
	+ Function Types
	+ Closures
* Collections
	+ Introduction
	+ Arrays
	+ Sets
	+ Dictionaries
* Object-Oriented Techniques
	+ Enumerations
	+ Structures
	+ Classes
	+ Computed Properties
	+ Property Observers
	+ Automatic Reference Counting (ARC)
	+ Inheritance
	+ Type Casting
	+ Initialization
	+ Deinitialization
	+ Protocols
	+ Extensions
	+ Access Control
* Error Handling
	+ Introduction
	+ Representing and Throwing Errors
	+ Handing Errors
	+ Cleanup Actions
	+ Guards
* Generics
	+ Introduction
	+ Generic Functions
	+ Generic Types
	+ Type Constraints
* Interoperability
	+ Introduction
	+ Objective-C Frameworks
	+ C Libraries
	+ Mixed Projects
* Conclusion