Swift Programming Essentials - TT4725

Master Swift's powerful syntax and features to craft clean, efficient code for iOS and MacOS applications.

Duration: 4 Days
Skill Level: Introductory
Available Format: Instructor-Led Online ; On Public Schedule

Swift Programming Essentials for OO Developers is an intensive four-day program designed for experienced object-oriented developers (from Java, C++ or C# background, etc.) transitioning to Swift.

What You'll Learn

Overview

Swift is a modern programming language that allows you to deliver highly optimized, safer, and more user-friendly applications, contributing to an improved user experience and better performance for your organization's software products.

Swift Programming Essentials for OO Developers is an intensive four-day program designed for experienced object-oriented developers (from Java, C++ or C# background, etc.) transitioning to Swift. Throughout the course, you'll gain valuable experience learning a wide variety of core skills to gain proficiency with Swift.

You'll start off learning Xcode and running code in the iOS simulator, and then move to exploring type safety and inference, control flow statements, different collection types, function and closure paradigms, as well as classes and structures. You'll then dive deep into Swift's error-handling mechanisms, ensuring you can develop and debug robust Swift code confidently.

This hands-on course dedicates half of the time to lab exercises where you'll actively write Swift code, manage data structures, control flows, and create reusable components. Using tools like Xcode and Playgrounds, you'll develop robust skills, crucial for crafting software within the iOS and broader Apple ecosystem. The course provides a comprehensive understanding of Swift's features to boost code performance and safety, while enabling you to design and implement high-performing applications. Upon completion, you'll confidently produce efficient and maintainable Swift code, ready to apply your newfound skills seamlessly to any Swift-based project.

Objectives

This course combines engaging instructor-led presentations and useful demonstrations with valuable hands-on labs and engaging group activities. Throughout the course you'll learn how to:

- **Master Swift Fundamentals:** Develop a thorough understanding of Swift's syntax, semantics, data structures, and unique features, laying a strong foundation for all your future iOS and MacOS projects.
- **Write and Debug Swift Code:** Learn to write clean, efficient Swift code using Xcode, along with troubleshooting techniques to keep your code bug-free and running smoothly.
- **Explore Control Flow and Collection Types:** Dive into conditional statements, loops, and collection types in Swift. Master the manipulation of program execution and data to build more dynamic and efficient applications.
- **Grasp Object-Oriented Principles in Swift:** Gain a solid understanding of Swift's approach to classes, structures, and enumerations. Apply object-oriented principles to design robust and reusable components.
- **Utilize Functions, Closures, and Error Handling:** These skills will empower you to write safer, more robust, and higher-performing code.
- **Get Hands-On with Xcode and Playgrounds:** Gain practical experience working with Apple's development tools - Xcode and Playgrounds.

If your team requires different topics, additional skills or a custom approach, our team will collaborate with you to adjust the course to focus on your specific learning objectives and goals.

Audience

This introductory-level technical course is ideally suited for experienced object-oriented programmers (coming from Java, C++, C#) seeking to transition to Swift for iOS and MacOS development. **This course is not for non-developers.**

Pre-Requisites

To ensure a smooth learning experience and maximize the benefits of attending this course, you should have the following prerequisite skills:

- Hands-on experience with another OO programming language, such as Java, C# or C++.
- Understanding of core programming concepts such as variables, loops, and control flow is essential to grasp the fundamentals of Swift swiftly.
- Basic knowledge of data structures like arrays, sets, and dictionaries will be beneficial when learning Swift's powerful data management capabilities.
- Understanding of what functions and closures are and how they work is important, as they're integral to Swift programming.

Agenda

Please note that this list of topics is based on our standard course offering, evolved from typical industry uses and trends. We'll work with you to tune this course and level of coverage to target the skills you need most. Topics, agenda and labs are subject to change, and may adjust during live delivery based on audience skill level, interests and participation.

1. Getting Started

- Getting Started with Xcode
- Playing in a Playground
- Running Your Code in the iOS simulator
- Troubleshooting Playgrounds
- Varying Variables and Printing to the Console
- Adding Comments

2. Types, Constants, and Variables

- Exploring Types
- Understanding type inference and type safety
- Using type annotation to specify a type
- Constants and Variables
- String Interpolation

3. Conditionals

- if/else statements
- Ternary Operator

- Nested ifs;
- else if

4. Numbers

- Integers
- Creating Integer Instances
- Operations on Integers
- Integer division
- Operator shorthand
- Overflow operators
- Converting Between Integer Types
- Floating-Point Numbers
- Ranges of Numbers

5. Switch

- Switch Syntax
- Ranges
- Value binding
- where clauses
- Tuples and Pattern Matching
- switch vs if/else

6. Loops

- for-in Loops
- where
- while Loops
- repeat-while Loops
- Control Transfer Statements in Loops

7. Strings

- Working with Strings
- Characters
- Unicode
- Unicode scalars
- Canonical equivalence

8. Arrays

- Creating an Array
- Accessing and Modifying Arrays

- Combining Arrays
- Array Equality
- Immutable Arrays
- Documentation

9. Optionals

- Optional Types
- Optional Binding
- Implicitly Unwrapped Optionals
- Optional Chaining
- Modifying an Optional in Place
- The Nil Coalescing Operator

10. Dictionaries

- Creating a Dictionary
- Accessing and Modifying Values
- Adding and Removing Values
- Looping over a Dictionary
- Immutable Dictionaries
- Translating a Dictionary to an Array

11. Sets

- What Is a Set?
- Getting a Set
- Working with Sets
- Iterating over a set
- Exploring Set operations
- Exploring set membership and equality
- Unions
- Intersections
- Disjoint
- Moving Between Types

12. Functions

- A Basic Function
- Function Parameters
- Parameter names
- Default parameter values
- In-out parameters

- Returning from a Function
- Nested Function Definitions and Scope
- Multiple Returns
- Optional Return Types
- Exiting Early from a Function
- Function Types

13. Closures

- Closure Syntax
- Closure Expression Syntax
- Functions as Arguments
- Closures Capture Their Enclosing Scope
- Functional Programming
- Higher-Order Functions
- map(_:)
- filter(_:)
- reduce(_:_:)

14. Enumerations

- Basic Enumerations
- Enumerations with Raw Values
- Methods
- Associated Values

15. Structs and Classes

- Structures
- Instance Methods
- Mutating methods
- Classes
- A Monster class
- Inheritance
- Making a subclass
- Overriding a superclass method
- Comparing value types and reference types
- Deciding between classes and structures

16. Properties

- Basic Stored Properties
- Nested Types

- Lazy Stored Properties
- Computed Properties
- A getter and a setter
- Property Observers
- Type Properties
- Access Control
- Controlling getter and setter visibility

17. Protocols

- Formatting a Table of Data
- Protocols
- Creating a protocol declaration
- Protocol Conformance
- Protocol Inheritance
- Protocols as Types
- Protocol Composition
- Mutating Methods

18. Extensions

- Extending an Existing Type
- Extending Your Own Type
- Using extensions to add protocol conformance
- Adding an initializer with an extension
- Nested types and extensions
- Extensions with methods
- Adopting a protocol via an extension

19. Error Handling

- Classes of Errors
- Lexing an Input String
- Catching Errors
- Parsing the Token Array
- Handling Errors by Sticking Your Head in the Sand
- Swift Error-Handling Philosophy

Related Courses

TT4713	Mobile Development Boot Camp: Android Development Essentials

TT4710	Mobile Development Boot Camp: Kotlin / Android (Part 1): Kotlin Programming Essentials
TT4721	Mobile Development Boot Camp for Swift / iOS: iOS Foundations with Swift
TT4720	Mobile Development Boot Camp for Swift / iOS: Swift Programming Essential

Setup Made Simple! Learning Experience Platform (LXP)

All applicable course software, digital courseware files or course notes, labs, data sets and solutions, live coaching support channels and rich extended learning and post training resources are provided for you in our “easy access, no install required” online Learning Experience Platform (LXP), remote lab and content environment. Access periods vary by course. We’ll collaborate with you to ensure your team is set up and ready to go well in advance of the class. Please inquire about set up details and options for your specific course of interest.

For More Information

Please contact us or call 844-475-4559 toll free for more information about our training services (instructor-led, self-paced or blended), coaching and mentoring services, public course enrollment or questions, partner programs, courseware licensing options and more.