

## Functional Programming in Swift: Thinking Functionally in Swift 4.0

### Course Summary

#### Description

At the end of this day, you'll look at functions differently.

We begin by looking more closely at functions you use already. Next we explore higher order functions that either accept or return closures. We play with four canonical examples from the Swift Standard Library: `map()`, `filter()`, `reduce()`, and `flatMap()` and then create `map()` functions of our own. We finish with a look at MVC, MVVM, and maybe even a reactive style.

We assume that you are an experienced programmer who has spent some time looking at Swift but you are new to Functional Programming. This course shows you how to add elements of a functional style to Object-Oriented and Protocol-Oriented Programming.

#### Topics

- Understanding Functions
- Higher Order Functions
- Swift Standard Library
- More with Map
- Architecture

#### Audience

This course is designed for experienced Swift developers who are new to Functional Programming.

#### Prerequisites

There are no prerequisites for this course.

#### Duration

One day

## Functional Programming in Swift: Thinking Functionally in Swift 4.0

### Course Outline

- I. Understanding Functions**
  - A. Function Parameters
  - B. Favor Immutability
  - C. Value Types
  - D. Mutating Methods
  - E. Non-mutating Methods
  - F. Embracing Generics
  
- II. Higher Order Functions**
  - A. Closures
  - B. Returning Closures
  - C. Consuming Closures
  - D. Custom Operators
  - E. Capture Lists
  
- III. Swift Standard Library**
  - A. Map
  - B. Filter
  - C. Reduce
  - D. FlatMap
  - E. Functors and Monads
  
- IV. More with Map**
  - A. Collections
  - B. Optionals
  - C. Error
  - D. Result
  - E. Non-Containers
  - F. FlatMap (Optional)
  
- V. Architecture**
  - A. MVC
  - B. MVVM
  - C. Reactive Style (Optional)