



iOS Programming

- **Introduction to Swift**
 - Swift vs. Objective-C
 - Swift language principles
 - The REPL and Playgrounds
 - LAB: Getting started with Swift
- **Variables, Types, and Control Flow**
 - Variables and constants
 - Built-in types
 - Conditional statements
 - Basic loops
 - LAB: Basic control flow
- **Optional Types**
 - Optional variables
 - Testing and unwrapping
- **Collections**
 - Tuples
 - Arrays
 - Dictionaries
 - Sets
- **Functions and Closures**
 - Basic functions
 - Higher-order functions, nested functions
 - Closures
- **Classes and Structures**
 - The Swift type system
 - Properties
 - Initializers
 - Methods
 - Property observers
 - Access modifiers
 - Inheritance and polymorphism
 - Structures



- **Enumerations**
 - Using enumerations
 - Associated values (sum types)

- **Pattern matching**
 - Basic pattern matching
 - Expression patterns
 - Custom expression matching

- **Memory Management**
 - Automatic reference counting (ARC)
 - Reference cycles
 - Weak and unowned references
 - Reference cycles with closures

- **Protocols**
 - Protocol inheritance and casting
 - Some useful protocols

- **Extensions** - Extending protocols

- **Generics**
 - Generic functions
 - Generic classes
 - Constraints
 - Protocols and associated types
 - Complex constraints

- **Operators**
 - Subscripts
 - Overloading operators
 - Custom operators
 - Associativity and precedence

- **Error Handling**
 - Objective-C vs. Swift error handling
 - Throwing errors
 - Calling functions that throw
 - Deferred execution



- **Interop with Objective-C**
 - Bridging basic types, classes, methods
 - Bridging strings and collections
 - Using id and optional
 - Error handling
 - API availability
 - Mix and match in the same project