DC ACM – 6/30/2014

Intro to Swift/Elections

Where are programming languages going?

Welcome - Agenda

- Welcome and introductions (10 mins)
- Intro to Swift Talk (30 mins)
  - Swift Demo
  - Discussion on Programming Languages
- Officer Elections
  - Nominations Chair: Andrew Conklin
  - Teller: Roger Fujii
- SIG-BEER
  - Networking

DC ACM/ACM

- Association for Computing Machinery (ACM)
  - An association for computing professionals
  - 100,000+ members, globally
  - Turing Award
- District of Columbia (DC ACM)
  - Local chapter, local events
  - Meetup group – meetup.com/dc-acm
  - Follow us on Twitter @dcacm

Intro to Swift

- A new programming language!
  - In beta, tools are under NDA
- The Swift developer experience
  - Modern language syntax
  - Native code
  - Specialized tool chain
    - LLVM Compiler (llvm.org)
    - XCode

Motivations - Language innovation

- Improve the app developer experience
- Improve upon established languages
  - Objective-C
  - C/C++
  - Java
  - Python
  - Rust
  - Ruby
- Modern compilation tools
  - Fast compilation within an IDE

Audience

- Audience
  - Developers for iOS/Mac apps
  - Individuals new to programming
  - Objective-C developers
    - Migrate to Swift

What does it look like?

- https://developer.apple.com/library/prerelease/ios/documentation/Swift/Conceptual/Swift_Programming_Language/TheBasics.html

Modern Language Trends

- Offer more than just a grammar
- Compatible with IDE features such
- A tool to make apps

- Favor designing the tools to enact complex work to offer a powerful, simple experience to the programmer
  - Extremely sophisticated compiler

More than a grammar

- LLVM compiler
- Standard library
  - Functions
  - Classes (arrays, dictionary, string)
- Playground
- User Interface Design
  - Interface Builder
  - UIKit compatible

Modern IDE Features

- Auto-completion
- Real-time compilation and execution
- Library loading
- Multilanguage (Swift w/objective-c, c)
- Live debugging
- Live Rendering of UI

A tool to make apps

- Type-safe
- Rapid learning curve
  - Emphasis on readability
  - Flexibility
- Instant on
- Native
- No raw memory access (no pointers)
- No non-nil references
- Automatic Reference Counting

Swift Classification

- Object-Oriented
- Imperative/Functional hybrid
    - Closures
    - Functions
- Native
- Concurrent

Flexibility...in types

- Classes
  - Single Inheritance, Multiple Protocols
- Structs
  - Can have methods
- Enums
  - Can have methods
- Generics
  - Array, Dictionary, or Custom
- Tuples

Where are programming languages going?

- Rethought for a wider experience range
  - First time programmers and Non-Programmers
  - Expert Programmers
- Combining open-source code with project code
- App building
  - Especially Games
- Continuous advancement
  - Grammar and tools are 1.0, rapid evolution

Other modern language

- Java (java.org)
  - New classes, language features continuously added
- Python (python.org)
  - Non-programmer friendly
- C# (mdsn.microsoft.com)
  - Type-safety
- Go (golang.org)
  - Fast, native compilation
- Julia (julialang.org)
  - Real time evaluations, tuples
- Ruby (ruby-lang.org)
  - No semi-colons!

From Here

- Language Guide
  - https://developer.apple.com/library/prerelease/ios/documentation/Swift/Conceptual/Swift_Programming_Language/TheBasics.html
- Swift/Objective-C Guide
  - https://developer.apple.com/library/prerelease/ios/documentation/Swift/Conceptual/BuildingCocoaApps/BuildingCocoaApps.pdf
- Blogs (a TON of them)
  - http://practicalswift.com/2014/06/14/the-swift-standard-library-list-of-built-in-functions/

Open Discussion

- Questions??
- Insights??
- Controversies??

Time for Elections

- Andrew Conklin
  - Past Chair (2012-2013)