CS 371L - Mobile Computing (iOS)

Dr. William C. Bulko

CS 371L – Mobile Computing (iOS)
Introduction

2014

The Evolution of Computing

- Mainframes
- Minicomputers - fridge-size
- PCs - desktop and deskside
- Laptops and tablets
- Phones and PDAs
- iWatch and other wearable computers
- Virtual and augmented reality
- Coming up: devices embedded in our bodies

The Internet of Things

1999:

- Referred to “tagging things”
- Idea was to equip everything with machine-readable information

Now:

- Term applies to more than just stored information
- Really means devices connected to the Internet with varying amounts of computation abilities
  - Smaller
  - More powerful
  - Less expensive
  - More connected

IP addresses

According to Gartner: there will be over 20 billion devices on the Internet of Things by 2020.

IPv4 (“old-style”):

- 128.101.68.110
- 2^{32} = 4,294,967,296 combinations (in practice, much fewer, because some are reserved)

IPv6

- 2001:0db8:0000:0000:0000:ff00:0042:8329
- 2^{128} = 340,282,366,920,938,463,463,374,607,431,768,211,456 combinations (enough to assign an IP address to each atom on the surface of the earth)

What you will need to be successful in this class:

- Ready access to a Mac computer
- Experience with an object-oriented programming language, such as C++, Java, or Python
- Xcode (Apple’s free IDE)
- An ability to teach yourself: initiative, diligence, and patience!

What you will not need:

- An iPhone or iPad
- An Apple Developer’s license

What we will cover in this class:

- Xcode
- Swift
- User Interface Essentials
- Some iOS frameworks (audio, graphics, etc.)

What we will not cover:

- How to use a Mac
- Programming concepts (basic, object-oriented)
- Objective-C
- Android
- Differences between versions of Xcode and Swift
- *Everything* you can do in iOS
- Provisioning profiles and publishing your app on the app store

Swift vs. Objective-C:

- Very different
- Swift is considered easier to learn
- Swift contains several features that make programming more productive
- Swift “feels like Python”

Swift vs. Objective-C:

Swift “hello world” command line application:

```
import Foundation
print("Hello, World!”)
```

Objective-C “hello world” command line application:

```
#import <Foundation/Foundation.h>
int main(int argc, const char * argv[]) {
   @autoreleasepool {
      NSLog(@"Hello, World!");
   }
   return 0;
 }
```

Swift vs. Objective-C:

- Very different
- Swift is considered easier to learn
- Swift contains several features that make programming more productive
- Swift “feels like Python”
- Swift is definitely the future of iOS programming, but Objective-C is here to stay
- Knowing both would give you an edge for employment opportunities after graduation

More on Swift

- Swift is a (relatively) new programming language created by Apple for developing OS X and iOS applications.
- The intent was to take the best of C and Objective-C, and not worry about C compatibility.
- According to Apple, it supposedly makes programming
  - easier
  - more flexible
  - more fun
- Swift provides seamless access to Cocoa frameworks
- It provides mix-and-match interoperability with Objective-C code
- Apple considers Swift to be a systems programming language – which means you can do system-level stuff like OS code
- Everything is an object

Frameworks

A *framework* is an abstraction in which software providing generic functionality can be selectively changed by additional user-written code, thus providing application-specific software.

We write our code to fit into how a framework is designed and works, in order to have a functioning iOS application.

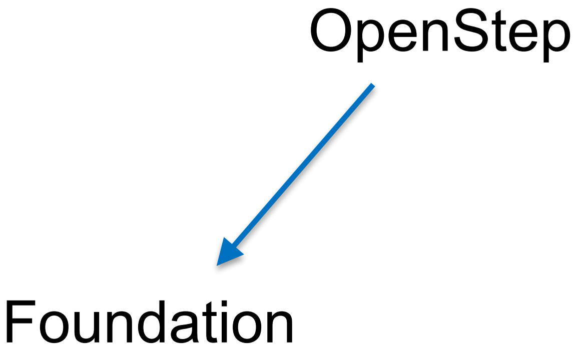
Frameworks

OpenStep

OpenSTEP is an object-oriented API specification for NeXT.

It can be split into two parts:

Frameworks

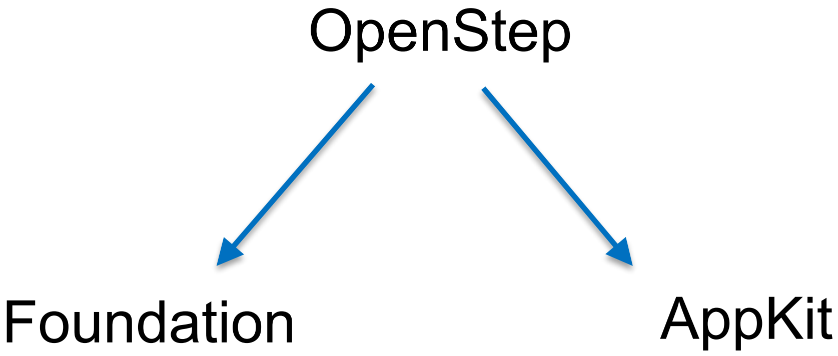


OpenStep

Foundation

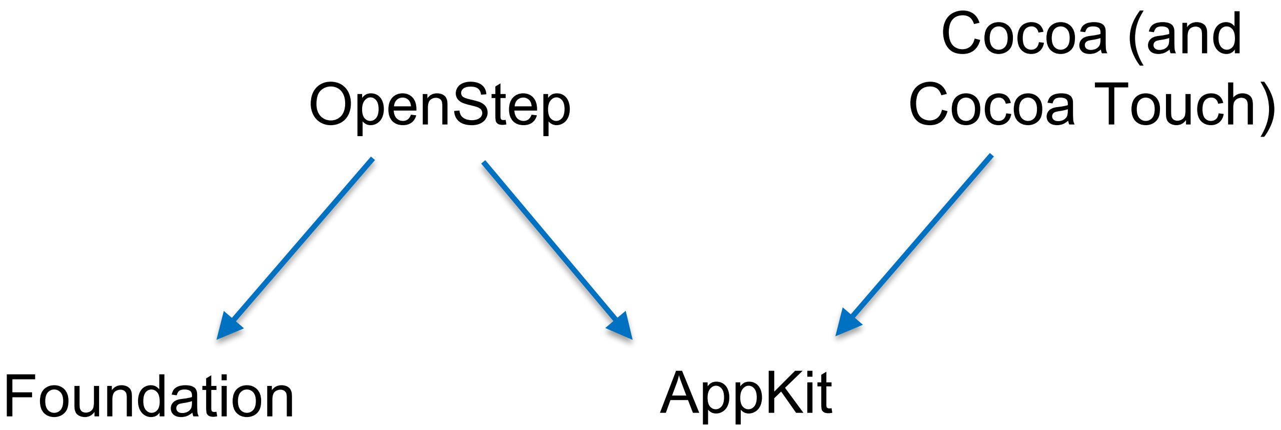
Foundation is an Objective-C framework under OpenSTEP. It contains low-level object definitions such as NSString, NSNumber, etc.

Frameworks



AppKit is a collection of classes within OpenSTEP, such as NSApplication, NSWindow, NSView, NSResponder, etc.

Frameworks



OpenStep

Cocoa (and
Cocoa Touch)

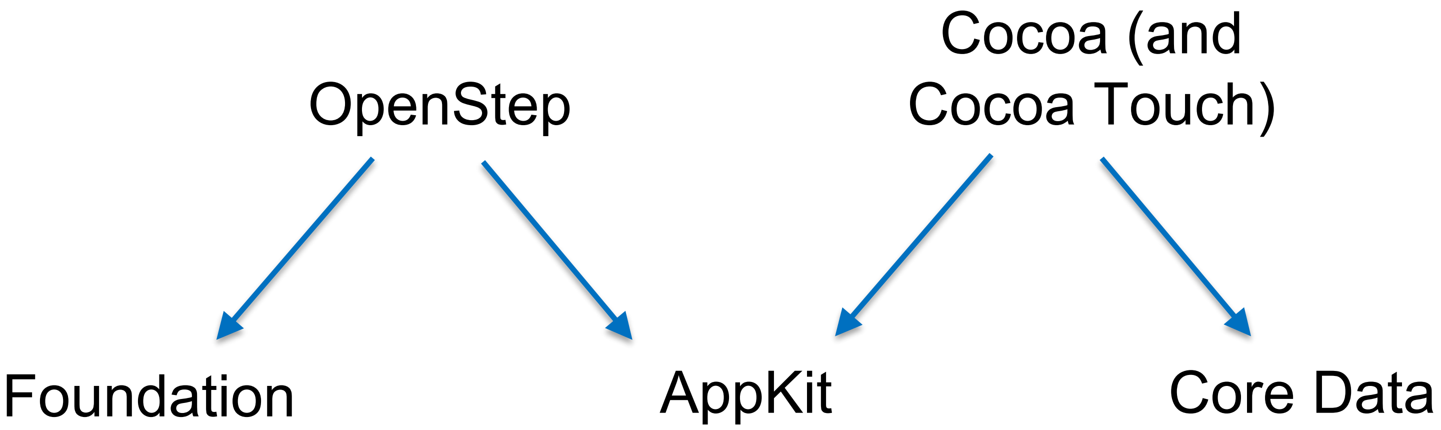
Foundation

AppKit

Cocoa is Apple’s native object-oriented API for OS X.

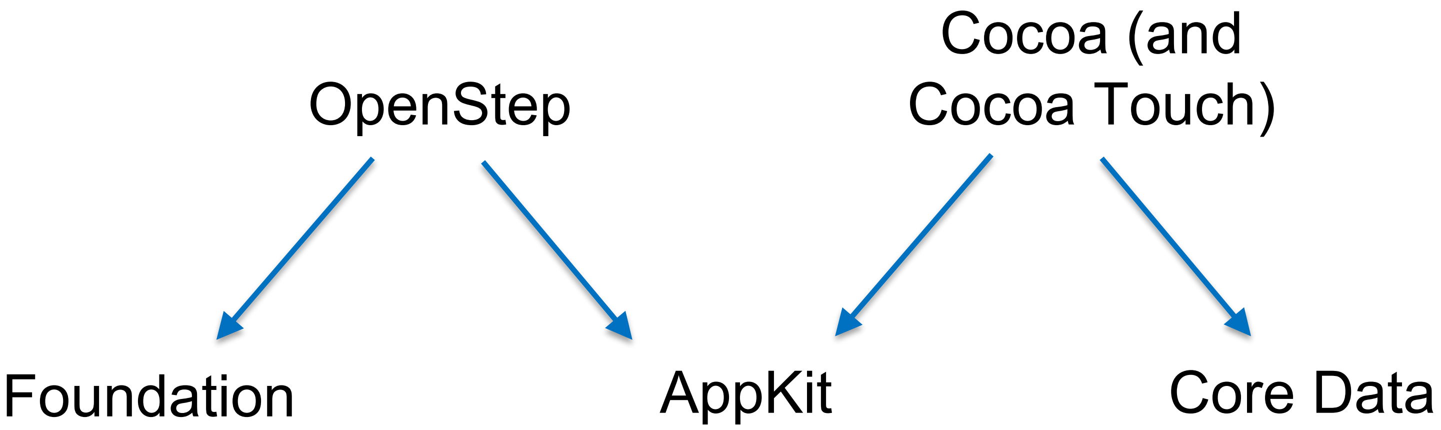
For iOS, tvOS, and watchOS, a similar API exists called **Cocoa Touch**.

Frameworks



Core Data is a framework provided by Apple that enables data to be organized by the relational entity-attribute model.

Frameworks



OpenStep

Cocoa (and
Cocoa Touch)

Foundation

AppKit

Core Data