

CS 329E – Elements of Mobile Computing

Introduction



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
- Amazon Echo

IP addresses

According to Gartner the number of devices on the Internet of Things will reach over 46 billion in 2021, triple of what it was in 2016.

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 “looks like C++ but 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