

CS 329E

Elements of Mobile

Computing

Spring 2018
University of Texas at Austin

Lecture 1

Agenda

- Introduction
- Syllabus
- Swift versus Objective-C
- Playgrounds
- Xcode beginnings
- Anatomy of an IOS application
- To do before next class

Introduction

Introduction

- Who I am
- Migration of computing
- What you need to succeed in this class
- What you do not need
- What we will cover
- What we will NOT cover
- Class deliverables
- Coding standard

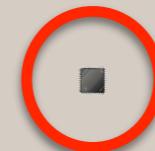
Introduction

Who I am:

- Bob Seitsinger
- Started in high tech in 1977
- Developed software for main frames (room size), mini computers (closet size), PCs (box size), mobiles (hand size), wearables (watch size)
- Developed desktop apps, device drivers, server-side stuff, mobile apps, etc
- Programmed in Swift, Objective-C, C++, C, Java, C#, Python, etc

Introduction

Migration of computing - changes during my career



Next step - implants?

...



I can't imagine the changes you'll see during your career?

Introduction

Coming in 2016



Virtual/Augmented Reality



The world's first social robot for the home.

Introduction

Coming 2017?

Autonomous vehicles - Tesla's Summon feature



Introduction

Artificial Intelligence

ANI - Now

AGI - ~2025

ASI - ~2040

Introduction

Increasingly, the world is becoming an intelligent, digitally enabled mesh of people, things and services.

Technology will be embedded in *everything*.

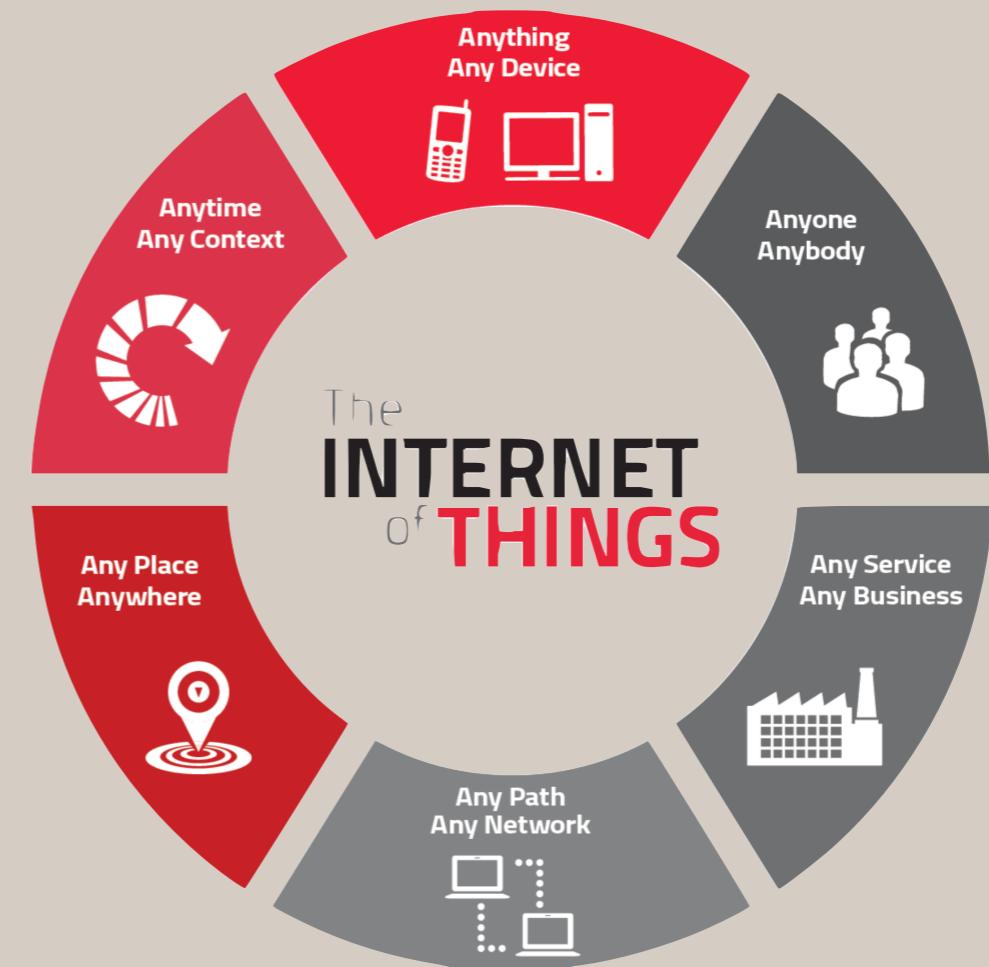
People will experience a world where the lines between what is real and what is digital blur.

Gartner, Inc.

Introduction

The trend continues unabated; the pace accelerating:

- Smaller
- More powerful
- Less expensive
- More connected



IPv6 supports 340,282,366,920,938,000,000,000,000,000,000,000 unique IP addresses.

The IPv6 address space is large enough to address EVERY ATOM ON THE SURFACE OF THE PLANET, and still have enough addresses left over to do another 100+ earths.

Introduction

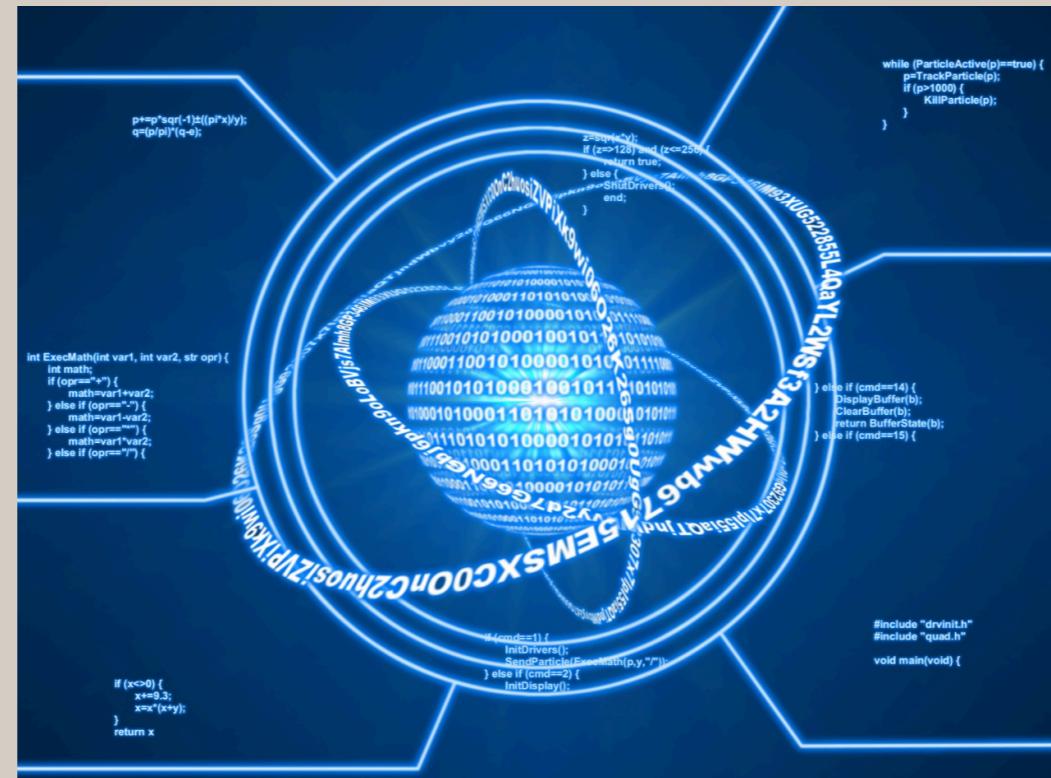
Some trends for 2018:

- Artificial Intelligence
 - CoreML
- IoT
- Cryptocurrencies
- Blockchain
 - Smart contracts
- Augmented Reality
 - ARKit
- Autonomous vehicles

Introduction

And we haven't even mentioned...

Quantum Computing



Introduction

Final point (for now):

Some are suggesting we are in the throws of technological advancement that will bring about...

The Singularity

The next significant evolutionary step of intelligence.

Where we are no longer the dominant intelligence.

Introduction

All this stuff needs software!!

Mobile is just one of many, many domains that need software.

Introduction

- What you need to succeed in this class:
 - Ready access to a Mac
 - There are some Macs in GDC 5.710A
 - Not a hangout room; grad students and researchers also using the room
 - **Mark L on the sign-in sheet if you need access to this room**
 - Knowledge/experience with an object-oriented language
 - Experience with OO capabilities - classes, objects, inheritance, etc
 - Some OO-capable languages would be C++, Java, C#, Python
 - Xcode (Apple's free IDE), latest currently version 9
 - You can download and install for free
 - A sense of curiosity, desire to explore and diligence
 - You WILL encounter problems that will be frustrating to solve

Introduction

Xcode downloads:

<https://developer.apple.com/download/>

The screenshot shows the Apple Developer website's 'Downloads' section. At the top, there is a navigation bar with links for 'Discover', 'Design', 'Develop', 'Distribute', 'Support', and 'Account'. A search icon is also present. Below the navigation bar, the user 'Bob Seitsinger' is logged in, with a 'Sign out' link. The main heading is 'Downloads', followed by the sub-instruction 'Get the latest beta releases of Xcode, macOS, iOS, watchOS, tvOS, and more.' A table lists software releases, with the first row highlighted by a red oval. The table columns are 'Release Software', 'Build', and 'Date'. The first row shows 'Xcode 9.2' with build '9C40b' and date 'Dec 4, 2017'. A blue 'Download' button is next to the date. A note below the table states 'Includes macOS, iOS, watchOS, and tvOS SDKs'. At the bottom of the page, a message requires a paid Apple Developer Program or Enterprise account to access certain software downloads, with a 'Download' link. A note at the very bottom asks if the user is signed in with the wrong account.

Release Software	Build	Date
Xcode 9.2	9C40b	Dec 4, 2017

Includes macOS, iOS, watchOS, and tvOS SDKs

Your Apple ID must be associated with a paid Apple Developer Program or Apple Developer Enterprise Program to access certain software downloads. You can download Xcode or sign in with a different account.

Are you signed in with the wrong account? [Sign out](#) and sign back in.

Introduction

You'll need an Apple ID:

<https://appleid.apple.com/account#!&page=create>

Introduction

- What you do NOT need:
 - An iOS developer license:
 - All deliverables only need to work on the simulator - which comes with Xcode
 - If you're interested in a license:
 - \$99 a year
 - developer.apple.com
 - An iOS device
 - Experience developing iOS applications
 - That's what this class is for!

Introduction

What we will cover:

- Xcode
 - Project creation
 - Creation of various types of apps
 - UI Creation via storyboards, manually and 3rd party tools
 - Debugging
- Swift, with a focus on version 4
- UI essentials
 - Views (Labels, Buttons,...), View Controllers, etc
- Various iOS frameworks
 - Foundation, UIKit, Core Graphics, MapKit, etc

Introduction

What we will NOT cover:

- How to use a Mac
- Basic programming *concepts* - variables, data types, functions, etc
- Object-oriented *concepts* - classes, objects, inheritance, polymorphism, etc
- *Everything* you can do in iOS - there's just too much
- Objective-C - we'll be using Swift!
- How to create provisioning profiles - requires an iOS developer license
- How to publish your app on the app store - involves a non-trivial approval process

Introduction

Assumptions:

- You know how to write non-trivial programs
- You understand how to design and program using an object-oriented paradigm
- You have never done any iOS application development before

Introduction

Class deliverables:

- This is a project based class, with some homeworks
 - You can discuss stuff with others but all homeworks must be your own work
 - The project will be a group/team effort - teams of 3
 - Make sure you contribute your portion to the project
 - I will be asking for input from each team member about each other team member
- ** You can't learn if you don't do! **

Introduction

Coding standard:

- I will be posting a short document on Canvas with some rules for how I expect you to write your code
 - You want to get into the habit of writing code that not only works, but is designed well and is easy to read and understand
 - You could get dinged on grading if you don't follow the standard
- The code you write is not for you. It's for the developer that picks up your code after you!

Syllabus

Syllabus

Available online in Canvas in the Syllabus folder.

Syllabus

The intent for each class is to introduce some amount of new material and allow time - interspersed/at the end - to work with the new material at some level

- typically in preparation for use in a homework

Swift versus Objective-C

Swift versus Objective-C

- Swift is easier to learn
- Comes with features that make programming more productive and fun - like the Playground
- How different is Swift from Objective-C? Very!
- Feels very Python-y
 - For example, implicit data typing
- A valid Swift program can be as small as one line of code!

Swift versus Objective-C

Example:

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 ‘hello world’ command line application:

```
print("Hello, World!")    ← NO semicolon!!!!
```

Swift versus Objective-C

Should I learn or forget about Objective-C?

Learn - Depends! Forget - No!

Why?

Clearly, Swift is the future for iOS application development but Objective-C is here to stay for years to come. There are tons of apps and lots of existing Objective-C code. It will take time to rewrite/replace all that code; all those applications.

Knowing both gives you an edge for employment opportunities.

Playgrounds

Playgrounds

What is a playground?

- It is an option in Xcode that allows you to play with code and get immediate results
 - File -> New -> Playground
- It is an interactive environment (similar to Python's) that allows developers to interactively write Swift code and see both final and intermediate results
- Provides an easy way to experiment with Swift code

Playgrounds

Demo

Xcode Beginnings

Xcode Beginnings

Brief Xcode how-to:

- Where to find Xcode after install
 - In the *Applications* folder
- Launching Xcode and keeping it in the dock
 - Right click Xcode icon in dock -> Options -> Keep in Dock
- Basics of the IDE
- How to create and run an iOS project
- How to debug in Xcode
 - Breakpoints

Anatomy of an iOS application

Anatomy of an iOS application

- Storyboard
 - Where the user interface is defined
- Delegate
 - At least one per application - the app delegate
- View Controller
 - At least one per application
- Views - the elements of your User Interface
 - Labels - text
 - Buttons
 - Etc
- The MVC (model, view, controller) pattern is used

Wrap-up

Wrap-up

What you learn in this class will allow you to write apps for:

- iPhones
- iPads
- iWatches
- Apple TV
- Apple Car(?)

Apple has a very powerful integrated development and execution eco-system

Wrap-up

Mac Lab:

Anyone needing access to the Mac lab on GDC 5th floor - **Mark L on the sign-in sheet in the Lab column.**

Wrap-up

To Do:

Get your environments set up:

- If you haven't already, download and install Xcode
- Play around with Xcode and Swift playgrounds to get familiar with them

** Bring your macs to class **

Next class we'll be going over some basics of Swift and the first homework will be assigned.