

Introduction to iOS

PROG31975 – Week 1

Jawaad Sheikh

Jawaad.Sheikh@sheridancollege.ca

Requirements

- **Prerequisites/Corequisites : please see course outline**
- **Required Hardware & Software:**
 - Intel-based Macintosh, running Mac OS X 10.7 Lion or higher
 - iPhone SDK
- **iPhone or iPod touch is not required**
 - Assignments may be completed using the iPhone Simulator

Embedded Programming

- Since semester 1, you've been learning how to program primarily for the desktop
- Now we move to the embedded or mobile platform.

Embedded Programming

- An embedded device is a scaled down computer usually consisting of a CPU, tiny motherboard, RAM and not much more.
- A mobile platform is an operating system that runs on an embedded device.

Embedded Programming

- Examples of embedded devices include cellphones, PDAs, netbooks, game consoles, set top boxes, modems.....and more!
- Examples of mobile platforms include Blackberry, iOS, Android, Windows Mobile...etc.

Embedded Programming

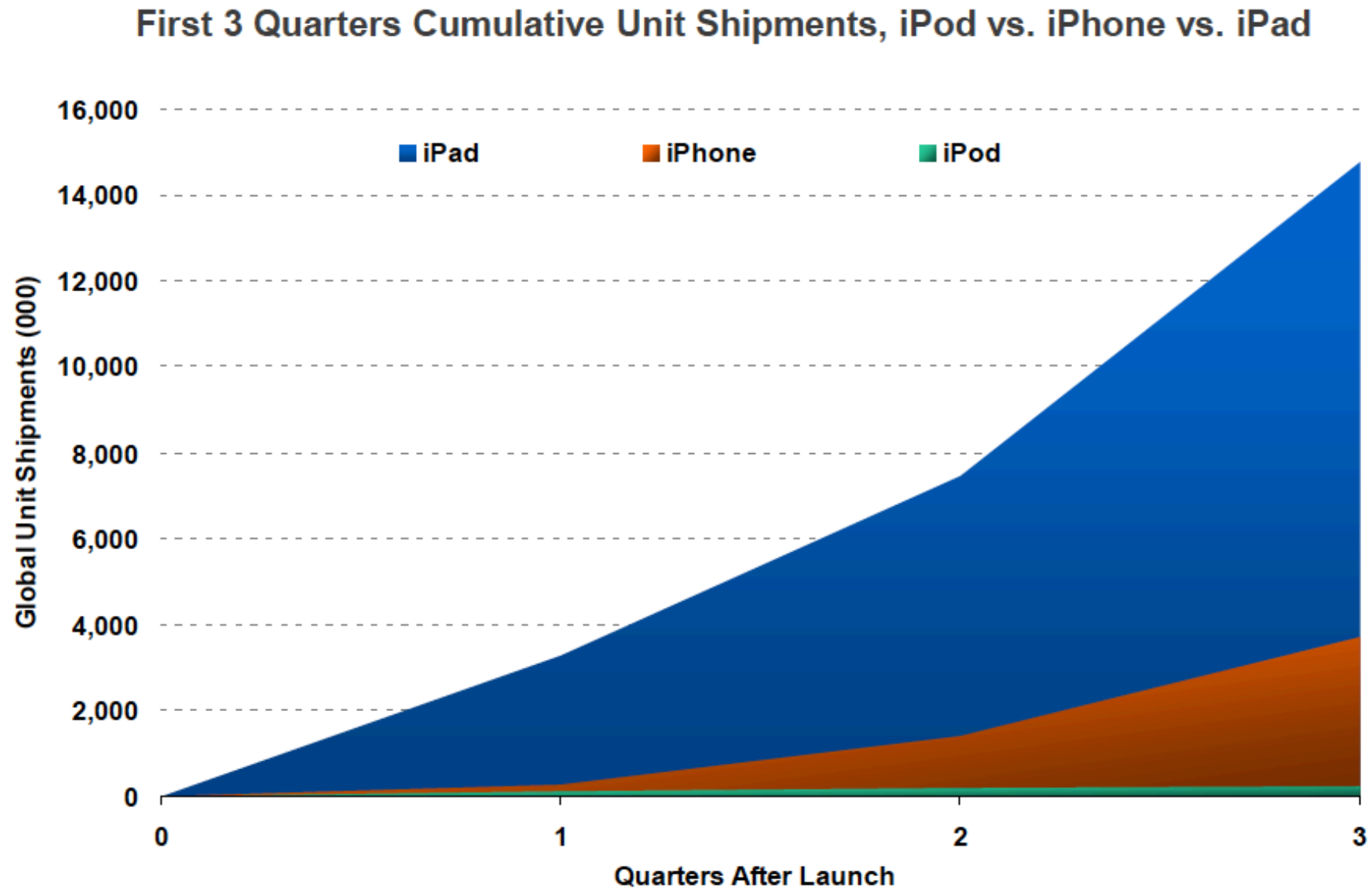
- In desktop programs, you have a wealth of resources and memory to work with.
- In mobile / embedded programming, you don't!
- Memory is scarce and reuse / compactness of code is necessary.
- Battery life is concern
- Network connectivity is an issue too.

Which devices are we going to focus on?

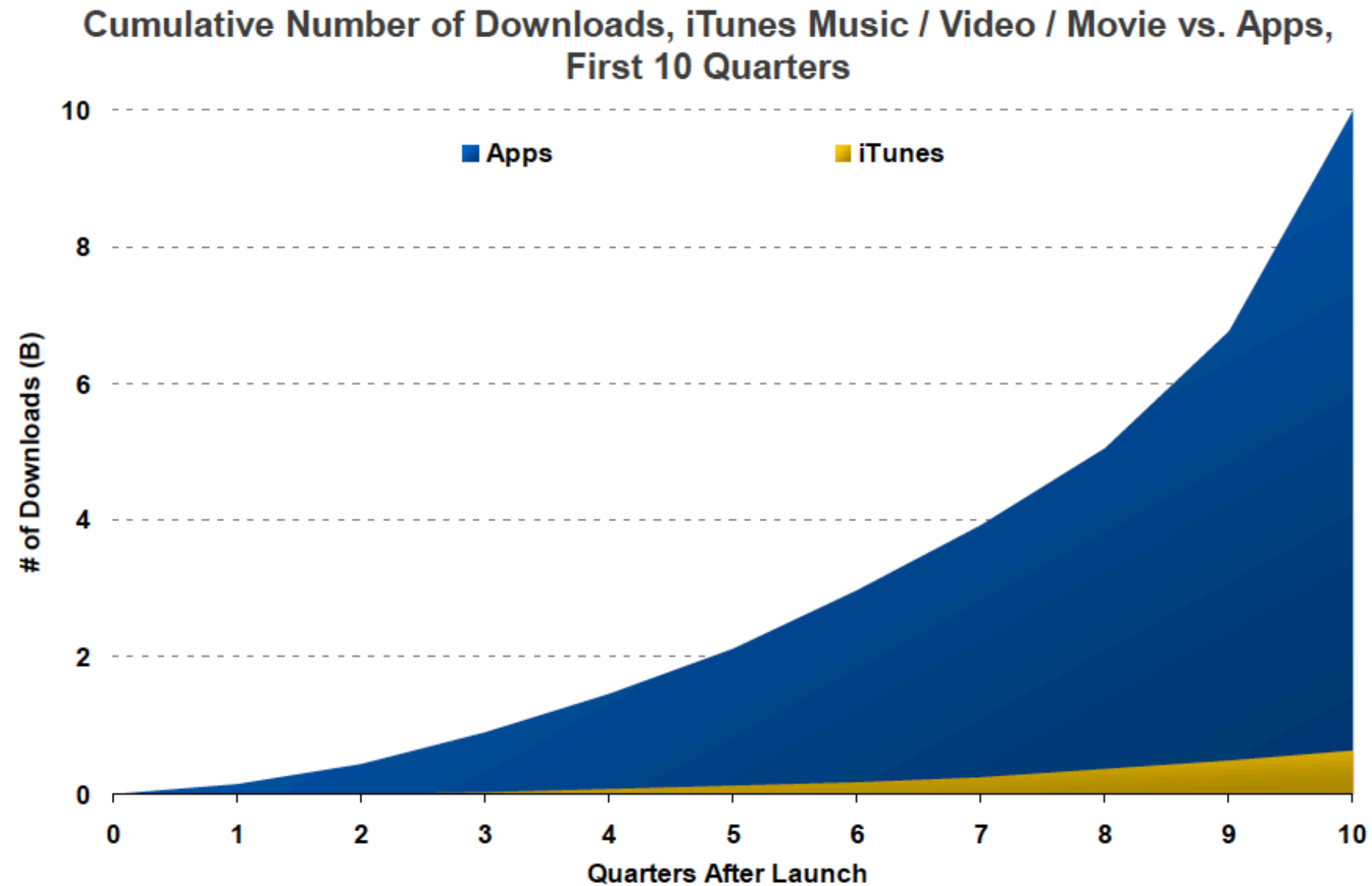


iPhone, iPod touch & iPad

Why Are We Here?



Why Are We Here?



Why Are We Here?

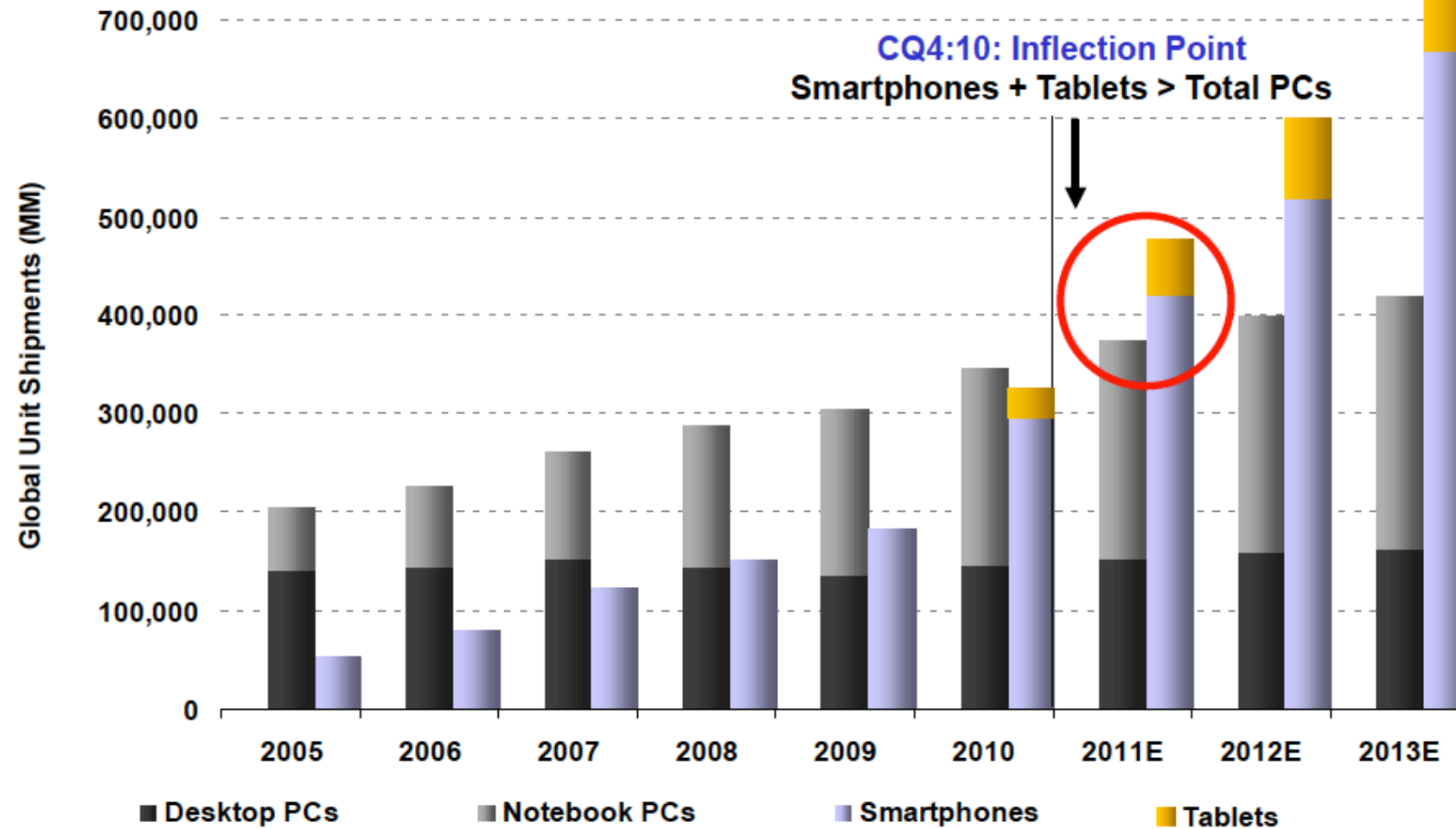
**Apple iPad =
Among One of the Fastest Growing New Consumer Computing Devices Ever**

Number of Days to Reach 1MM Units Sold

						
Nintendo Wii*	Nintendo DS*	iPad	iPhone	Netbooks	BlackBerry	iPod
~13	~15	28	74	~180	~300+	~360+

Why Are We Here?

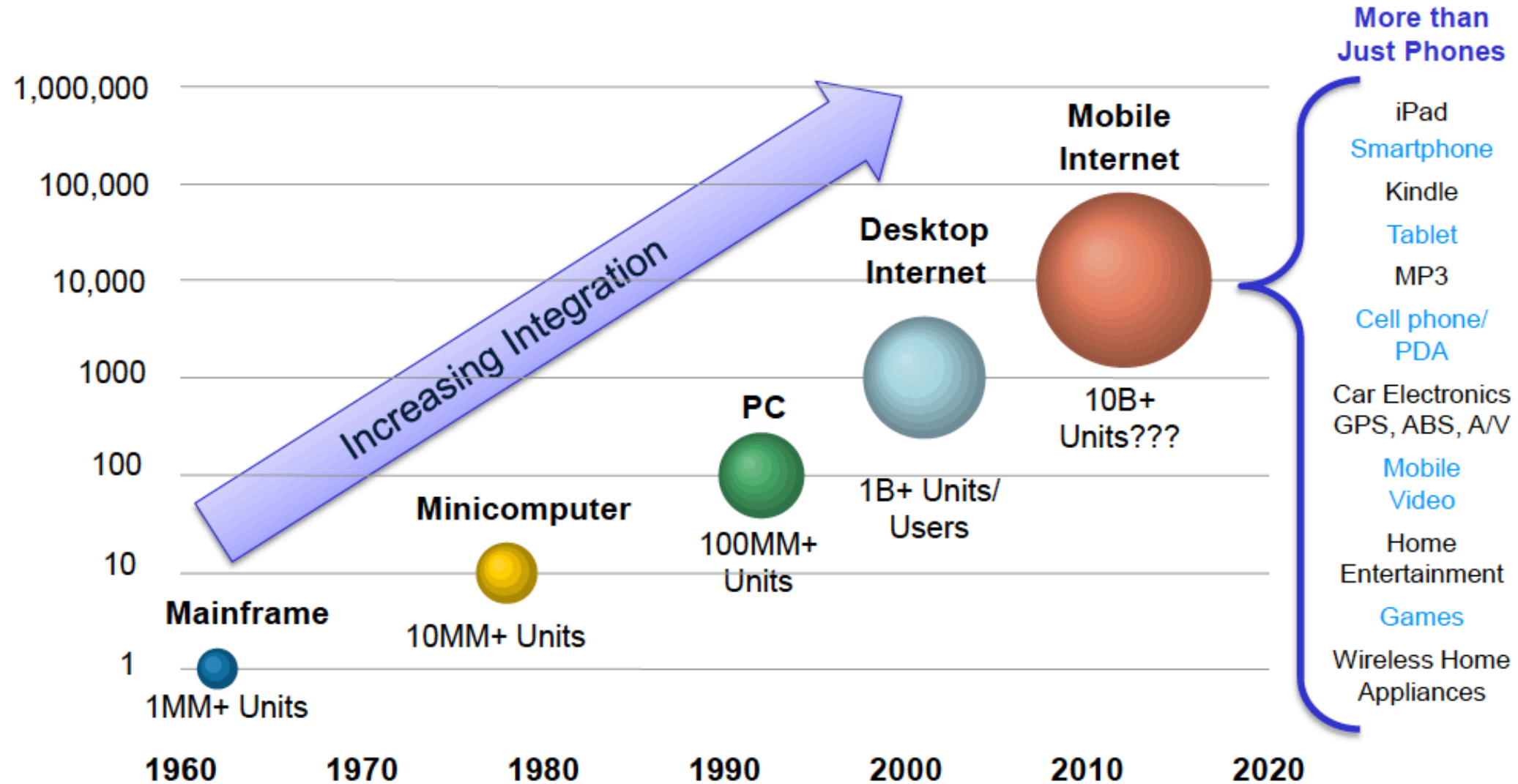
Global Unit Shipments of Desktop PCs + Notebook PCs vs. Smartphones + Tablets, 2005-2013E



Note: Notebook PCs include Netbooks. Source: Katy Huberty, Ehud Gelblum, Morgan Stanley Research.
Data and Estimates as of 2/11

Why Are We Here?

Computing Growth Drivers Over Time, 1960-2020E



Note: PC installed base reached 100MM in 1993, cellphone/Internet users reached 1B in 2002/2005 respectively;
Source: ITU, Mark Lipacis, Morgan Stanley Research.

iPhone Developer University Program

- Sheridan is part of the iPhone Developer University Program
 - If you want to test your apps on your iPhone/iPod touch/iPad Xcode 8 and above allow you to sideload the app
 - If you want to develop your own apps for the App Store, pay the \$99 to be an Application Developer

Why Are We Here?

- The course is not just about the iPhone, Cocoa Touch or Swift
- It's about Software Engineering, as well as object oriented architecture and design
- Exposure to problems and solutions that you might not see in other classes

Cocoa Touch & iPhone SDK

- Based on Cocoa
 - Mature, polished, highly consistent APIs
 - consists of a suite of object-oriented software libraries, a runtime, and an integrated development environment.
- Provides a very rich starting point for exploring app design
- Shows “real-world” implementations of OO design patterns
- Designs learned on iPhone translate directly to Mac OS X

Things we'll be covering in the course:

- Tools
- Frameworks
- Language (& Runtime)

Tools



Xcode



Interface Builder

Frameworks

Language
(& Runtime)

Tools



Xcode



Interface Builder

Frameworks



Foundation



UIKit

Language (& Runtime)

Tools



Xcode



Interface Builder

Frameworks



Foundation



UIKit

Language (& Runtime)

```
textView.text = "hello"
```

Swift

Why Swift?

- Exposure to other languages is always good
- Swift is a language focused on simplicity and the elegance of OO design
 - Shares a lot of commonalities with Java and Javascript
 - Brings many object oriented principles, but with a minimal amount of syntax
- A data point to compare with designs of C, C++ and Java

What We Will Cover

- Application design patterns
- View controllers
- Presenting data
 - Table views
- Loading & saving data
- Text input
- And if we have time...
 - MapKit, Video & Photo APIs, & more
 - Threading and Performance
 - Address Book and other system integration

Final Projects

- 5-7 weeks to work on them in groups of 3-4
- Never too early to think of something propose it to me
- Categories to consider:
 - Student life apps
 - Educational tools
 - Games
 - Social / location-aware software
- Find a project that you or your friends would actually like to use!











Exercise 1 – App Design

- Think back to the last 3 apps you downloaded
- What did they have in common?
- How did they look and feel?
- Compare it to the apps from your colleagues at your table

iPhone

- SmartPhone with multimedia capabilities.
- iPhone OS X- based on Mac OS X
 - Difference is multi-touch interface and accelerometer support.
- Unix kernel.
- Touch screen.
- Supports 3rd-party applications.

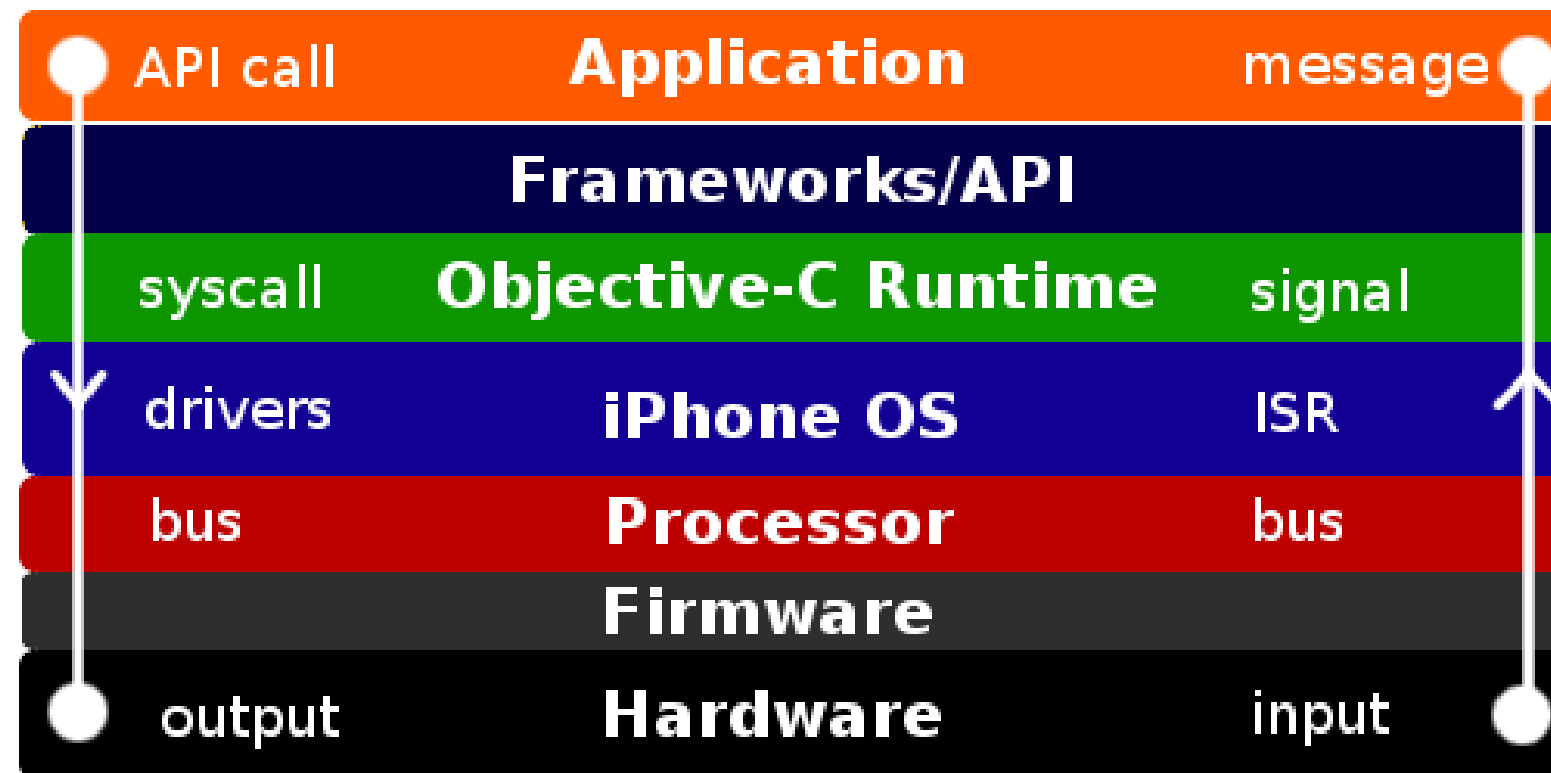
iPhone specifications

Device	Retina	Portrait (px)	Landscape (px)
iPhone X		1125 x 2436	2436 x 1125
iPhone 6+, 6S+, 7+, 8+		1080 x 1920	1920 x 1080
iPhone 6, 6S, 7, 8		750 x 1334	1334 x 750
iPhone 5, 6SE <i>5, 5S, 5C, 6SE</i>		640 x 1136	1136 x 640
iPhone 4 <i>4, 4S</i>		640 x 960	960 x 640
iPhone <i>1st, 2nd & 3rd Generation</i>		320 x 480	480 x 320
iPad Air / Retina iPad <i>1st & 2nd Generation / 3rd & 4th</i>		1536 x 2048	2048 x 1536
iPad Pro		2048 x 2732	2732 x 2048
iPad Mini <i>2nd, 3rd & 4th Generation</i>		1536 x 2048	2048 x 1536
iPad <i>Mini, 1st & 2nd Generation</i>		768 x 1024	1024 x 768

iPhone OS Overview

- It is a 7 layer architecture

iPhone Architecture



iPhone OS

- The OS takes up about 2 GB of the device's total storage!
- Supports bundled apps from apple and third party developers.
- Designed to only run software with apple approved cryptographic software.
- >Millions of apps on Apple's AppStore

Tools



Xcode



Interface Builder

Frameworks



Foundation



UIKit

Language (& Runtime)

```
textView.text = "hello"
```

Swift

iOS Overview

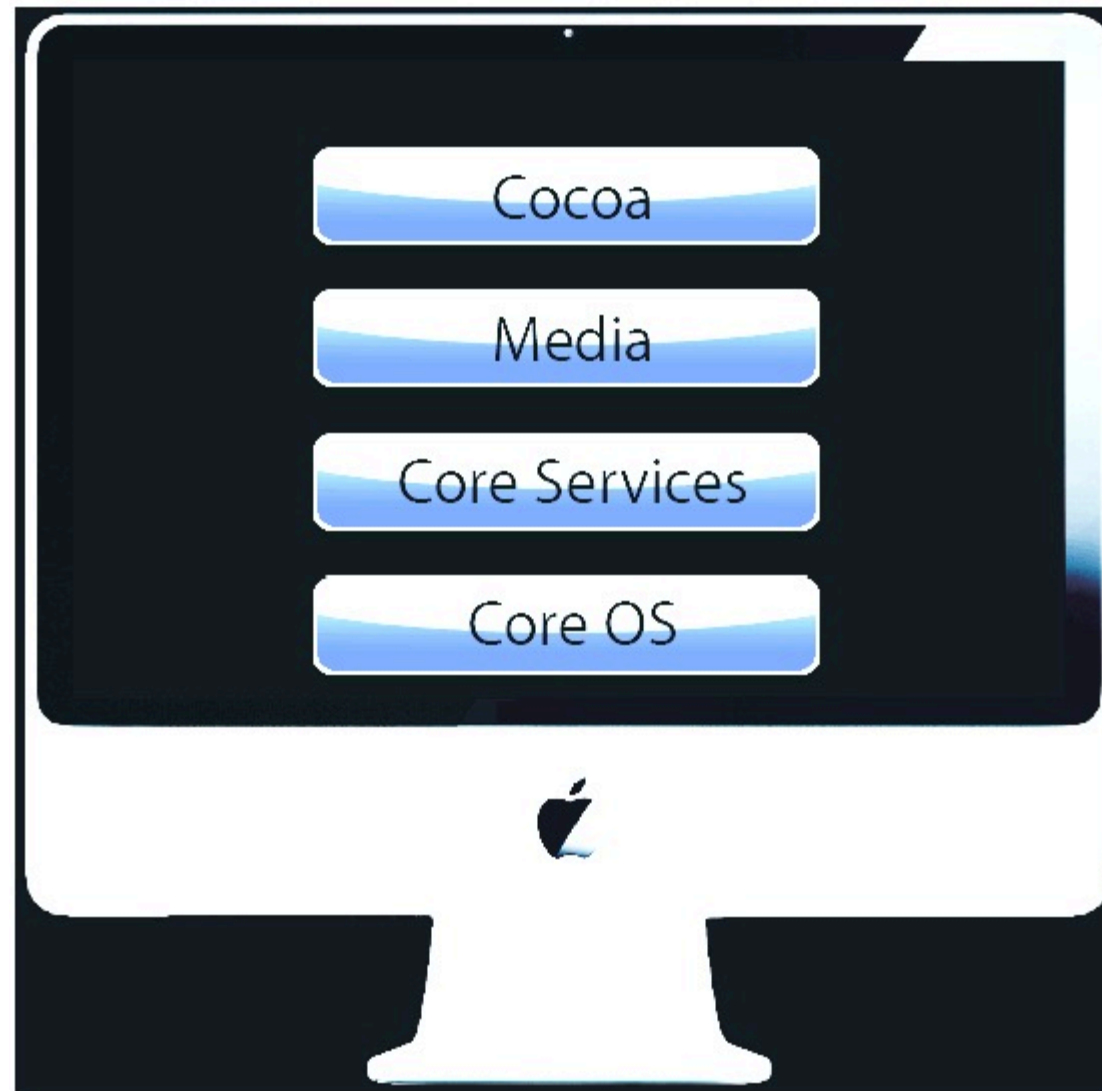
iPhone



Mac OS X



Mac OS X



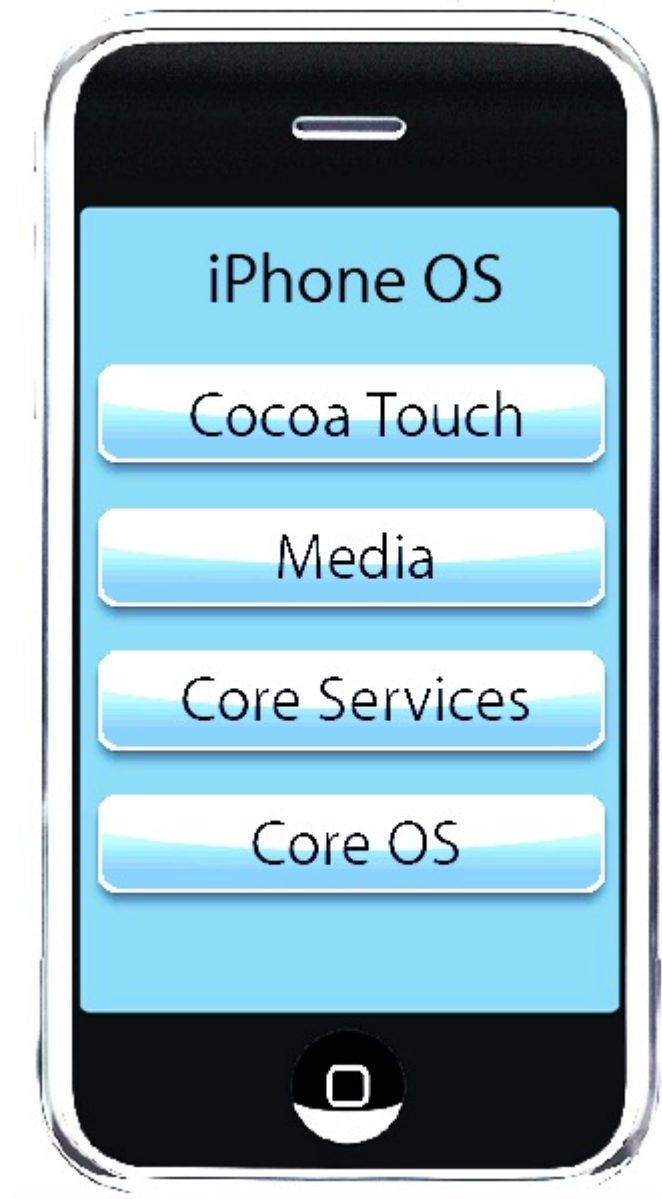
Cocoa

Media

Core Services

Core OS

iPhone OS



4 abstraction layers:

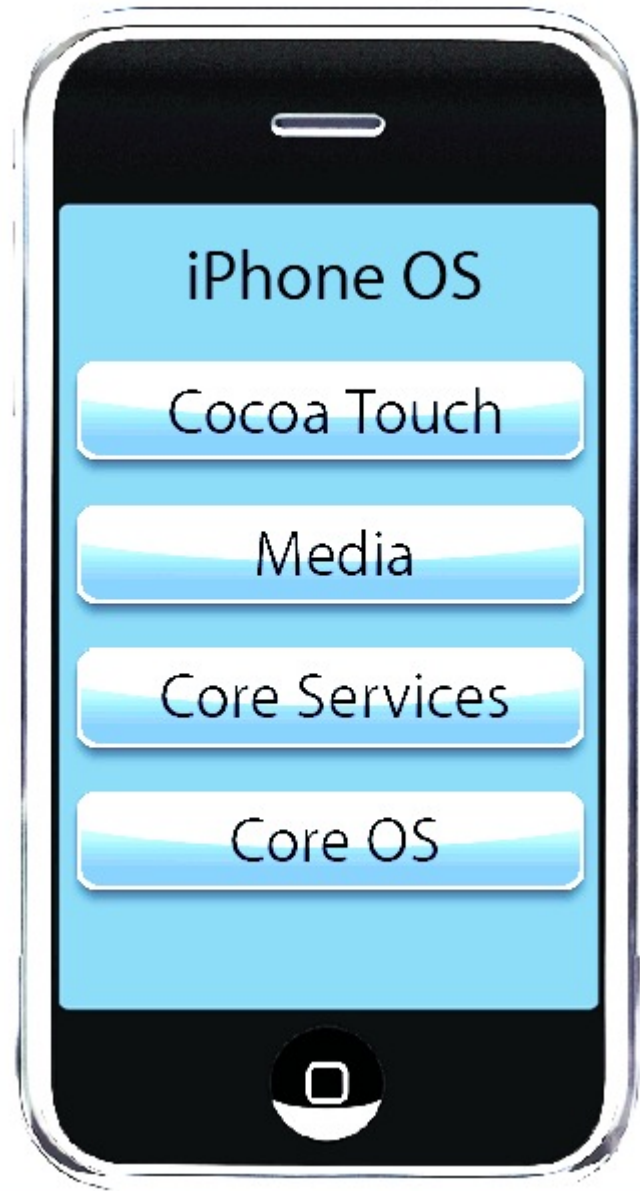
Cocoa Touch

Media

Core Services

Core OS

iPhone OS



Core OS:

sits directly on top of the device hardware and provides low level system interface support

OS X Kernel

Power Mgmt

Mach 3.0

Keychain

BSD

Certificates

Sockets

File System

Security

Bonjour

iPhone OS



Core Services:

provides the fundamental system services that all applications use

iCloud	File Access
ARC	SQLite
Block Objects	Core Location
Data Protection	Net Services
Address Book	Threading
Networking	Preferences
	URL utilities
	and many more...

iPhone OS



Media:

provides the iPod, iPhone and iPad audio, video, animation and graphics capabilities

Core Audio

JPG, PNG, TIFF

OpenAL

PDF

Audio Mixing

Quartz (2D)

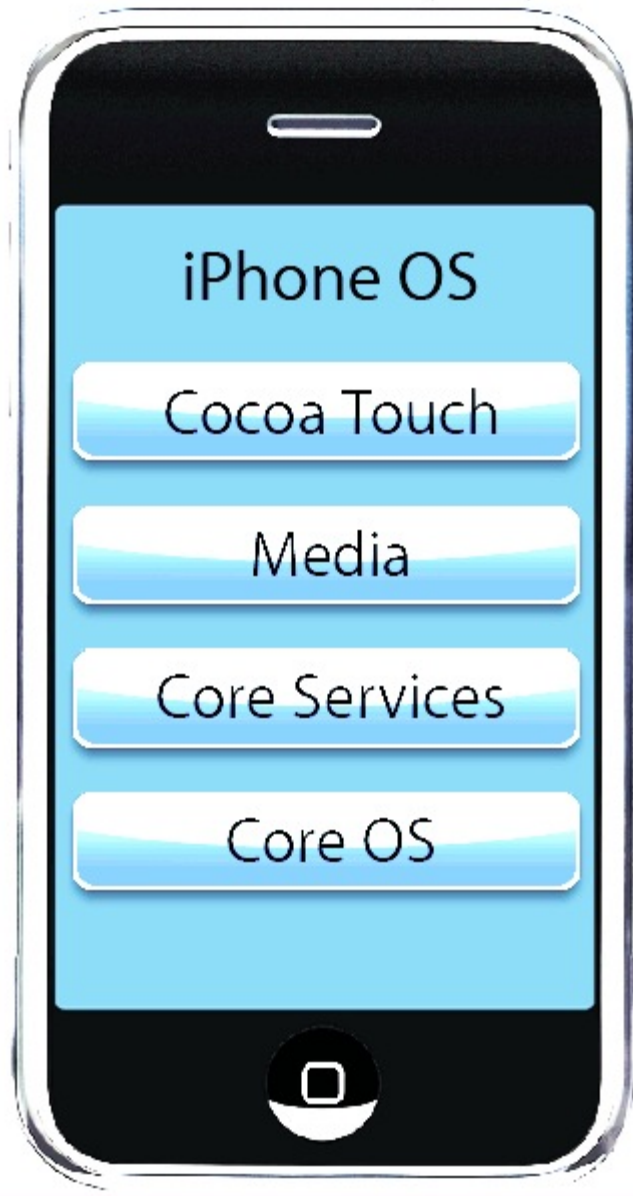
Audio Recording

Core Animation

Video Playback

OpenGL ES

iPhone OS



Cocoa Touch:

iOS development framework, principally consisting of Foundation, UIKit, and Core Animation, Core Audio, and Core Data.

Multi-Touch Events

Multi-Touch Controls

Accelerometer

View Hierarchy

Localization

Alerts

Web View

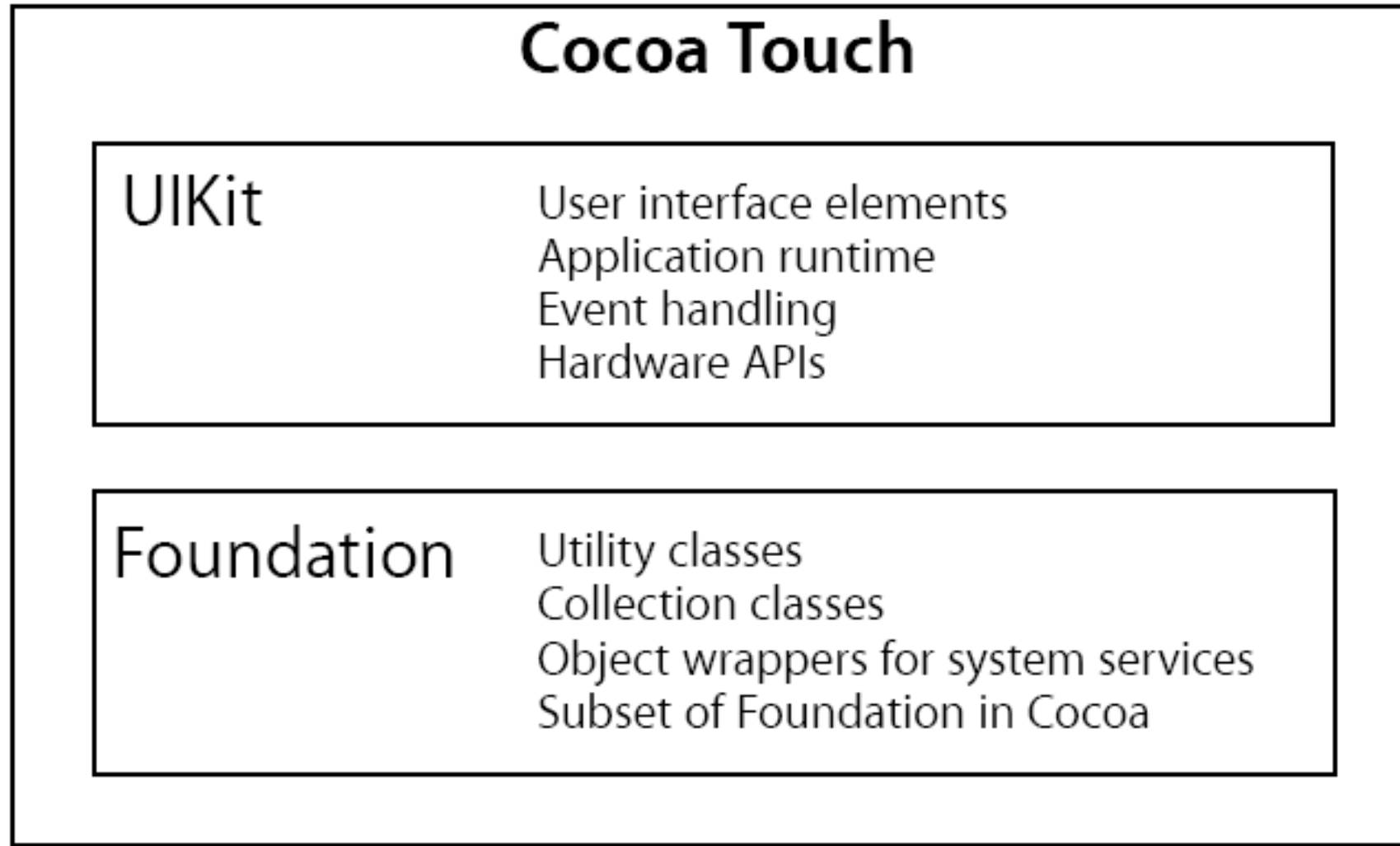
Photo Library

Image Picker

Camera

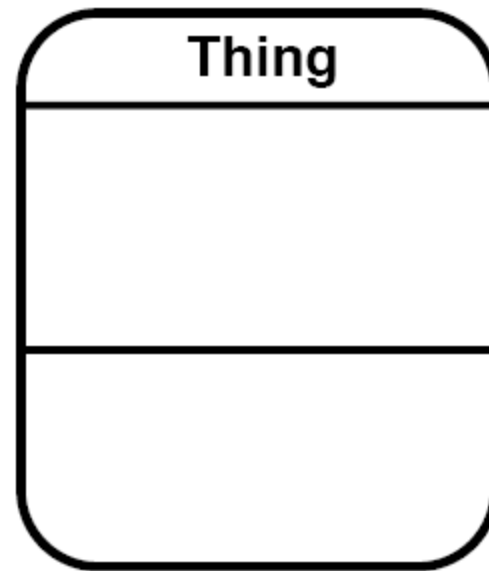
<http://developer.apple.com/technologies/ios/cocoa-touch.html>

Cocoa Touch Architecture

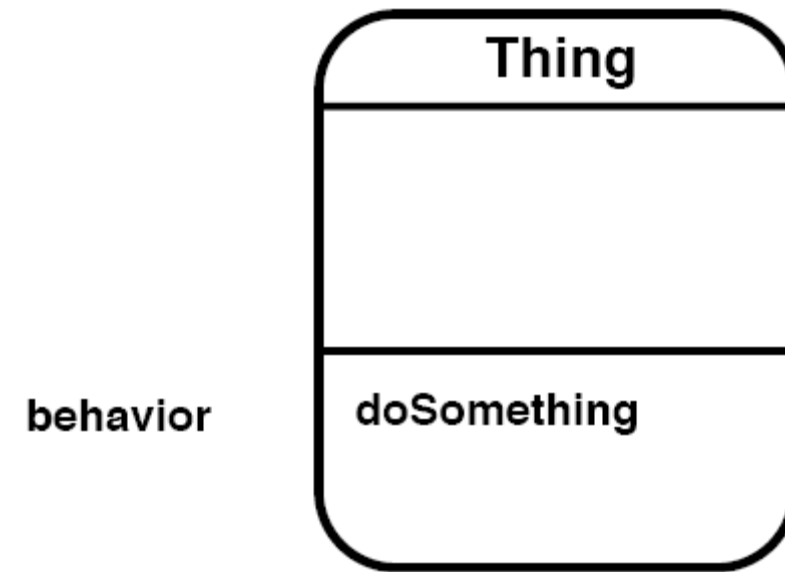


Objects

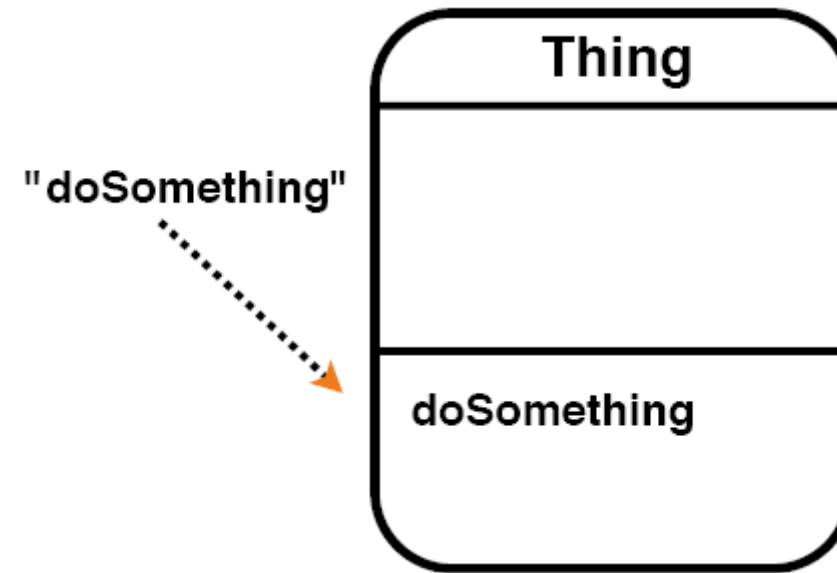
Object



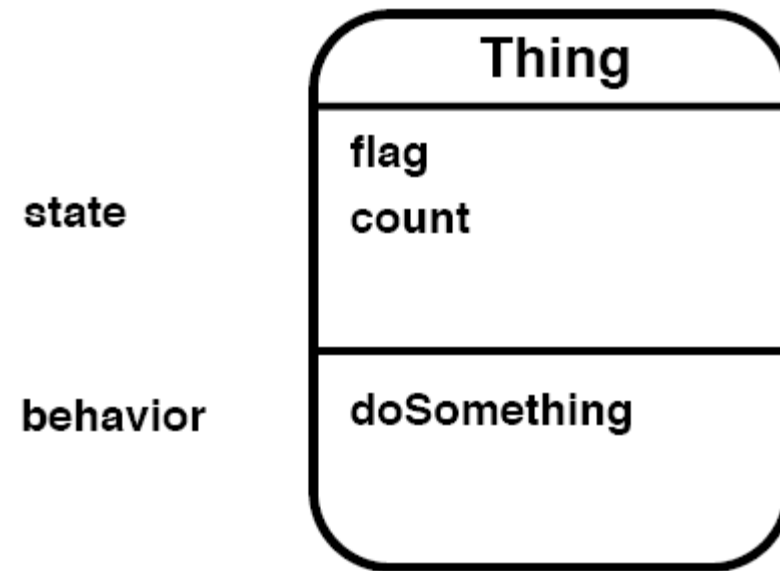
Behavior



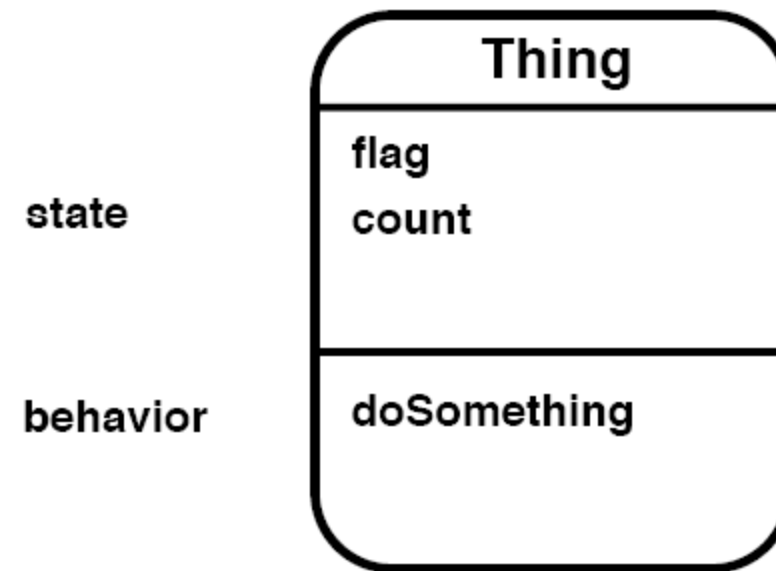
Message



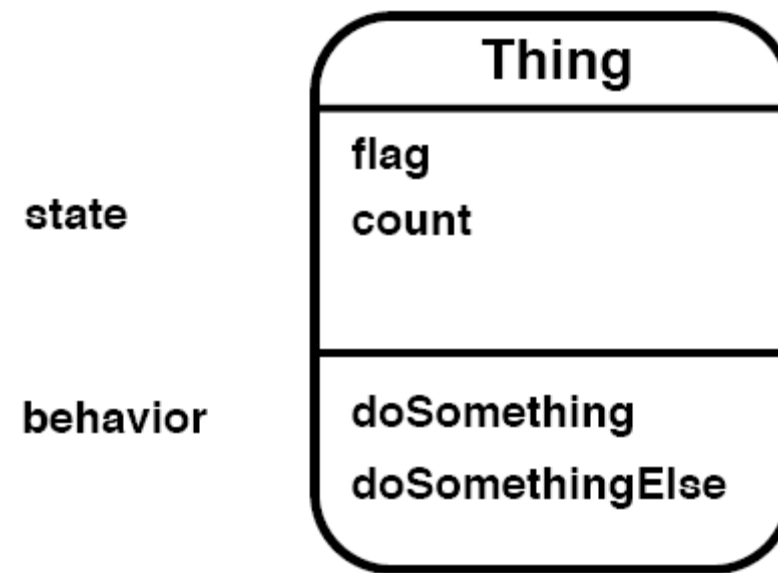
State



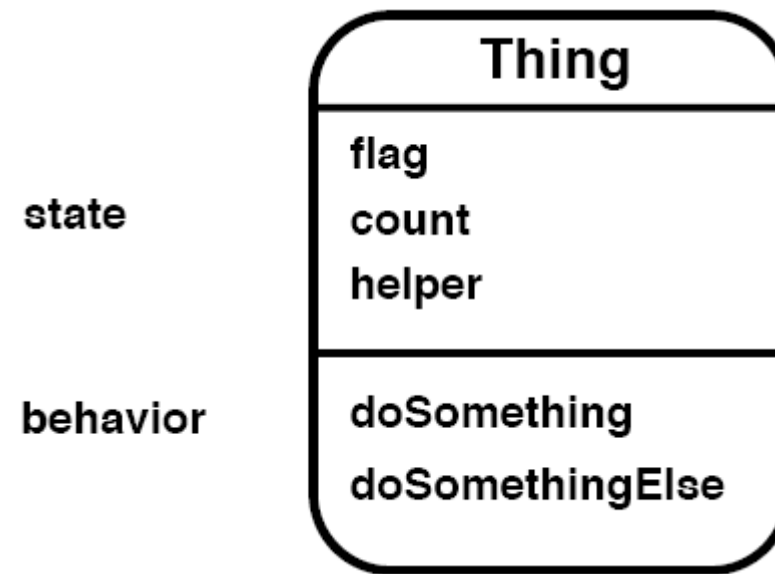
Other Objects As State



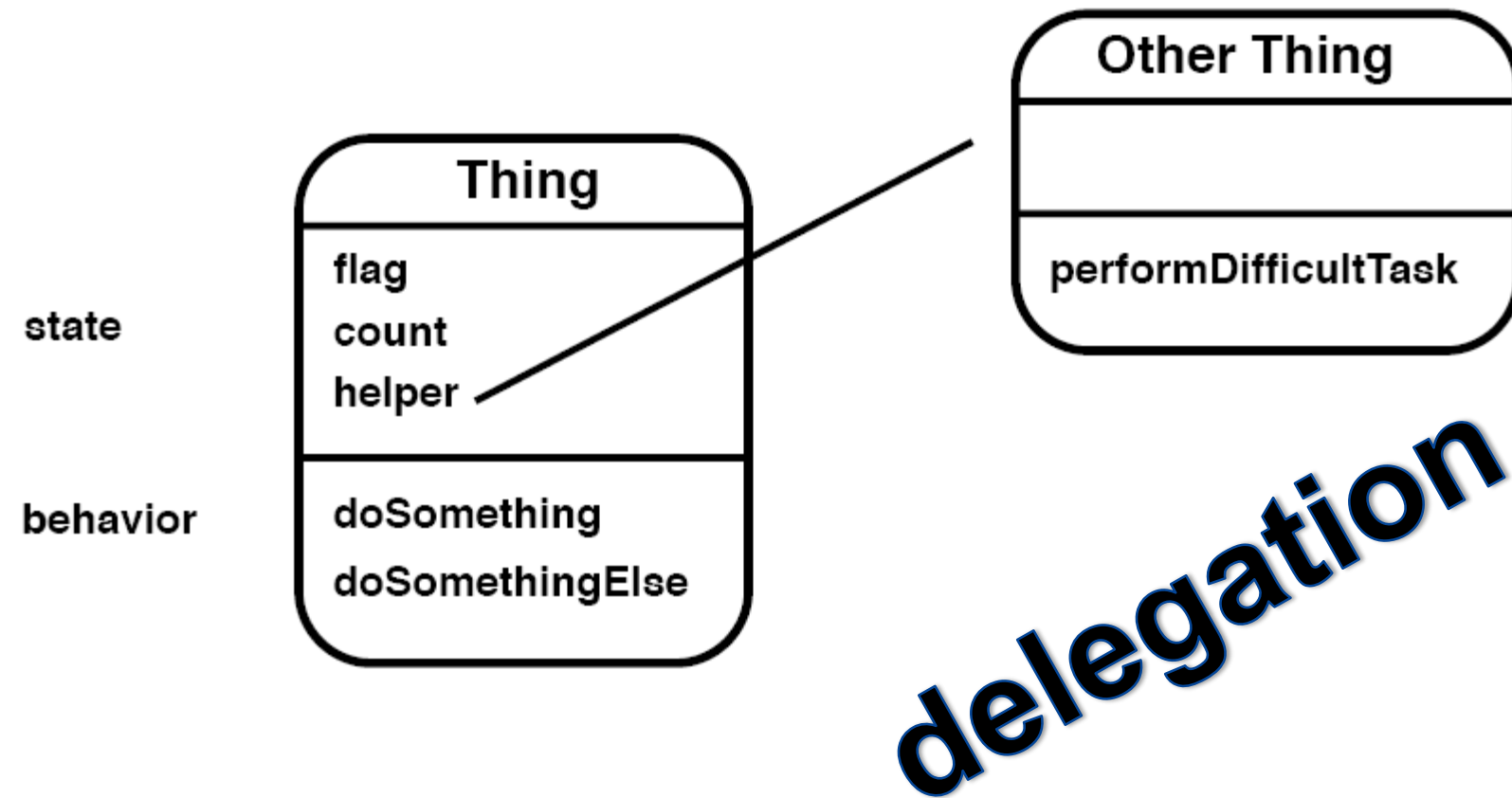
Other Objects As State



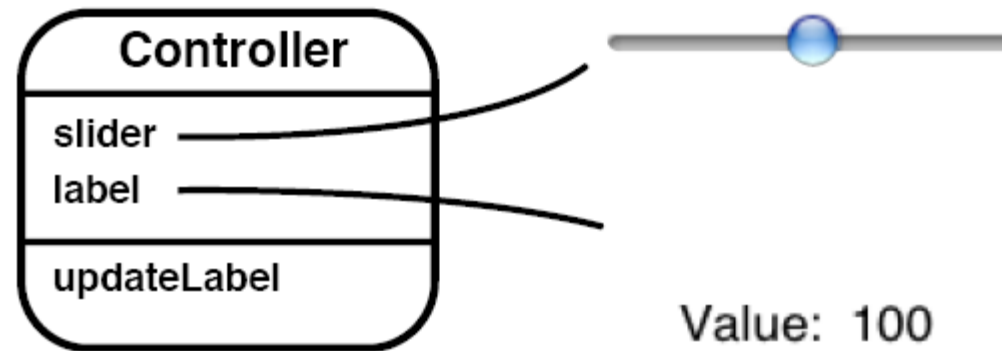
Other Objects As State



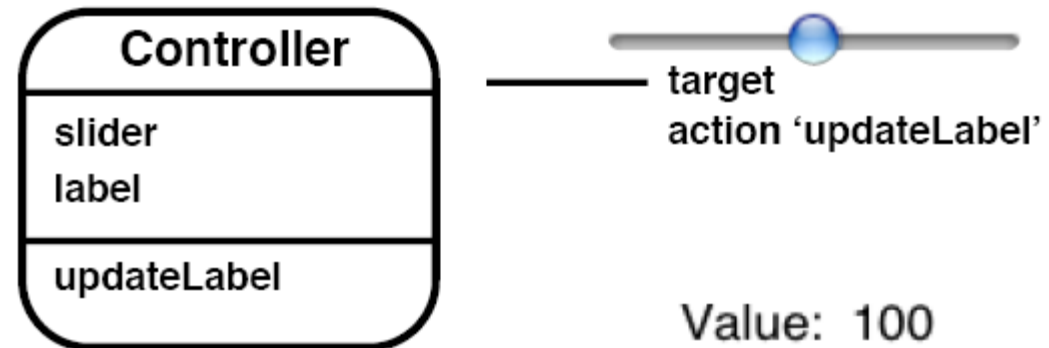
Other Objects As State



Outlets

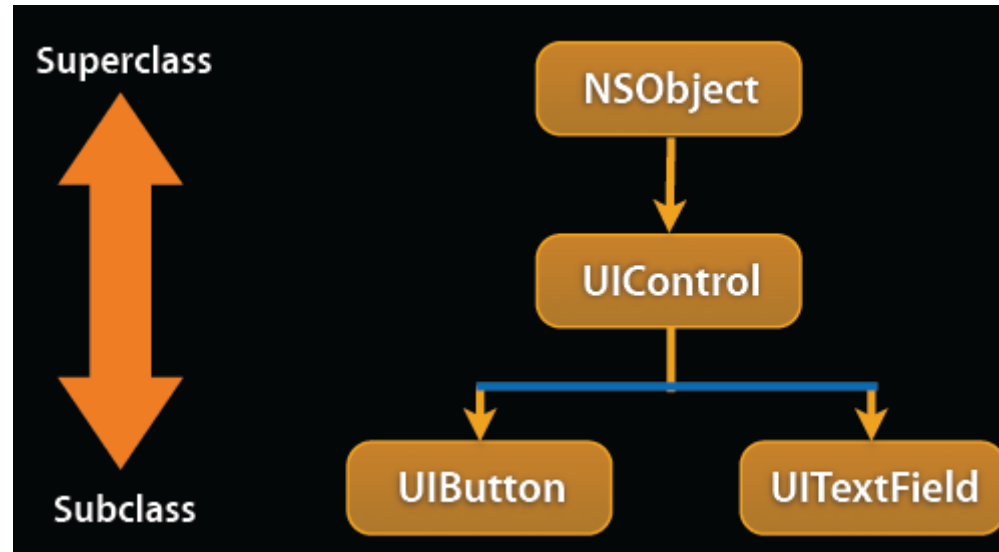


Target / Action



Inheritance

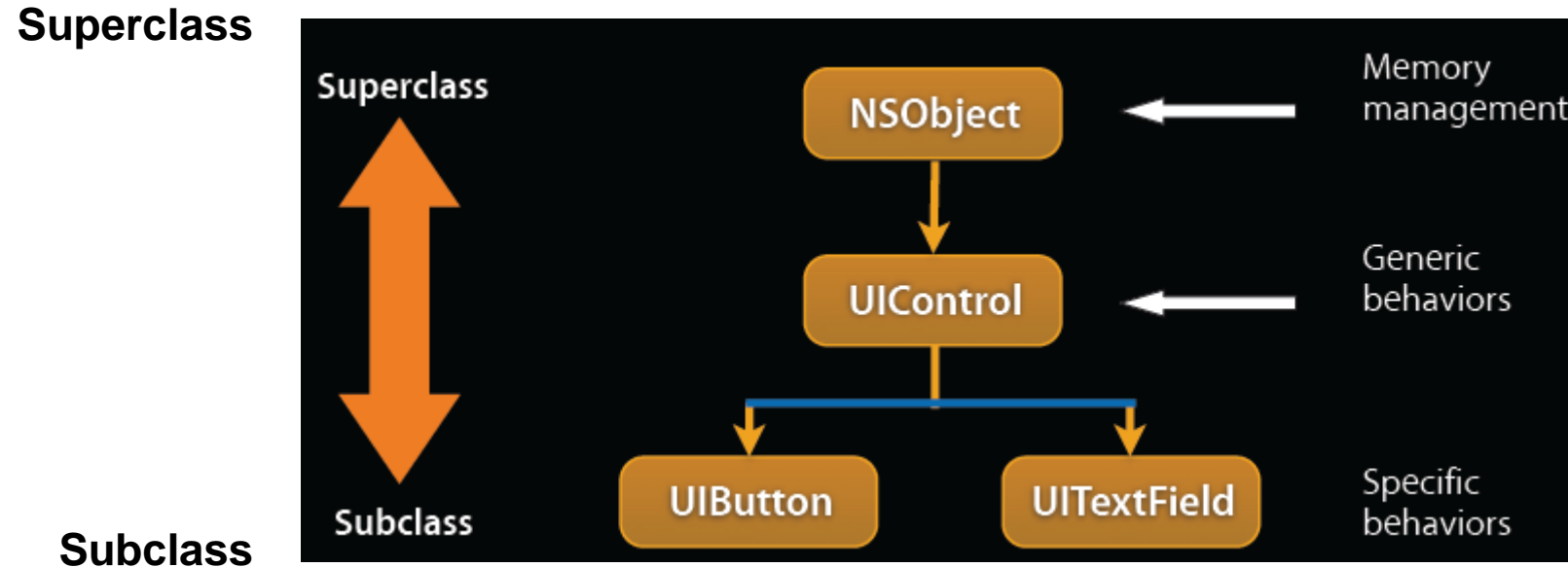
Superclass



Subclass

- Hierarchical relation between classes
- Subclass “*inherit*” behavior and data from superclass
- Subclasses can use, augment or replace superclass methods

Inheritance



- Hierarchical relation between classes
- Subclass “inherit” behavior and data from superclass
- Subclasses can use, augment or replace superclass methods

Questions?

Exercise 2 – App Design

- Go back to the 3 apps you looked at in the last class.
- What widgets did they use?
- Come up with a block diagram of how each app page is laid out and how those widgets were used.
- Compare and contrast with your classmates at your table.