

Today

- What is this class all about?

Why am I here?

- Prerequisites

You must be a strong object-oriented programmer.

- iOS Overview

What's in iOS?

- Show me!

A demo with a thousand words is worth tens of thousands of words.



What will I learn in this course?

• How to build cool apps

Easy to build even very complex applications.

Result lives in your pocket or backpack!

Very easy to distribute your application through the AppStore.

Vibrant development community.

• Real-life Object-Oriented Programming

The heart of Cocoa Touch is 100% object-oriented.

Application of MVC design model.

Many computer science concepts applied in a commercial development platform:

Databases, Graphics, Multimedia, Multithreading, Animation, Networking, and much, much more!

Numerous students have gone on to sell products on the AppStore.



Prerequisites

• Prior Coursework

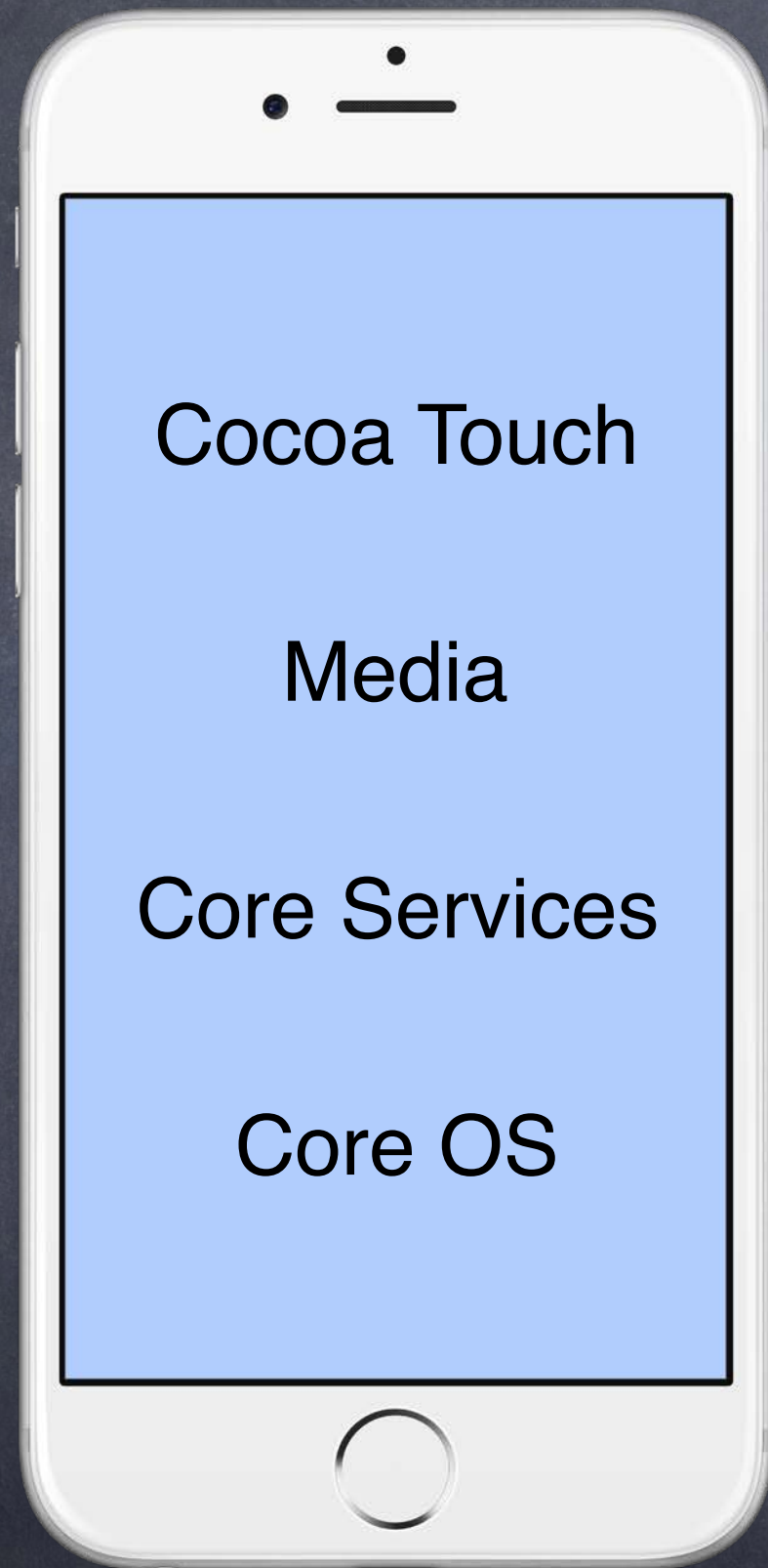
Object-Oriented Programming experience mandatory.

CS106A&B (or X) required & CS107 or CS108 or CS110 also (at a minimum) required.

(or equivalent for non-Stanford undergrads)



What's in iOS?



Core OS

OSX Kernel Power Management

Mach 3.0 Keychain Access

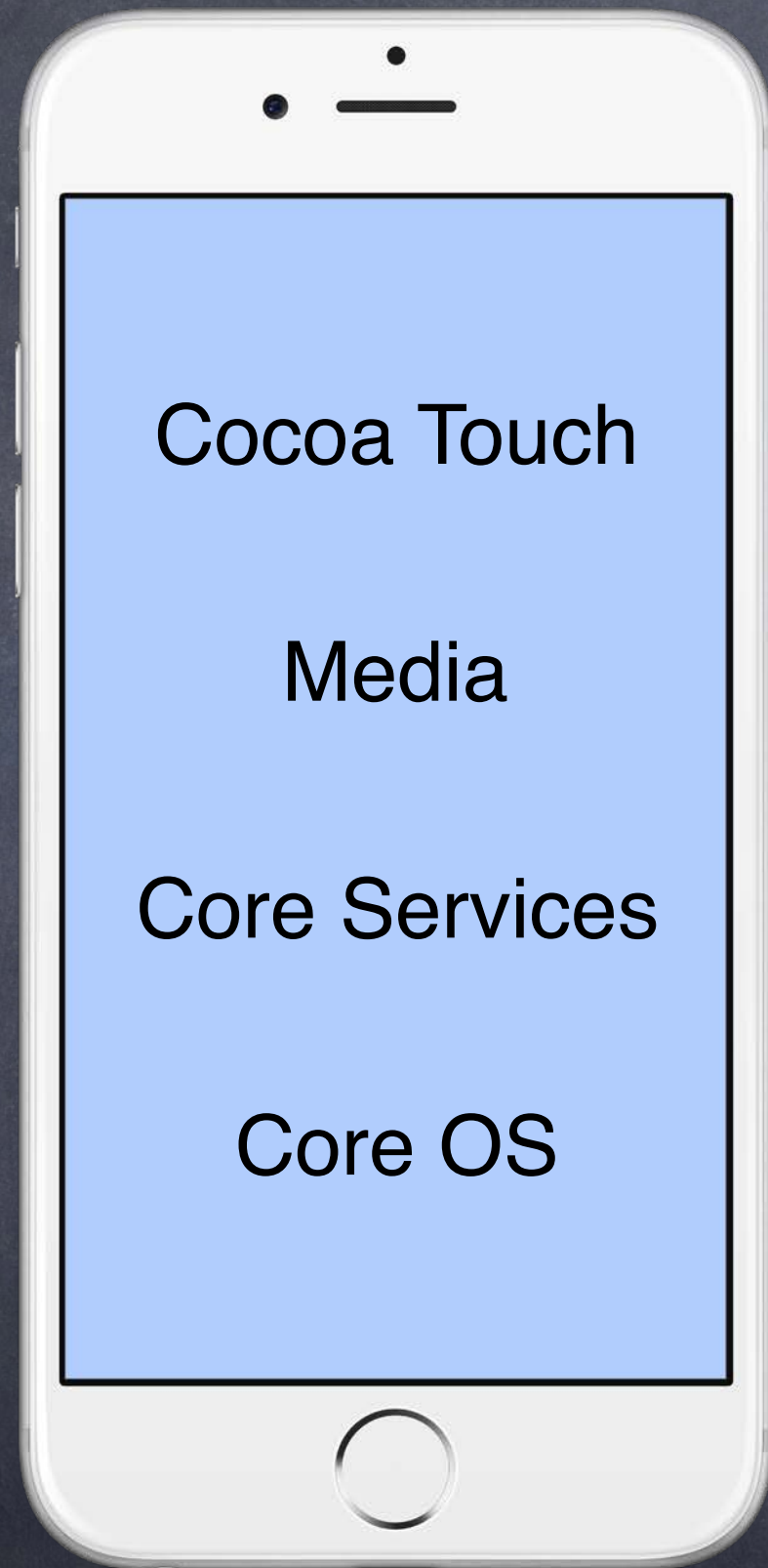
BSD Certificates

Sockets File System

Security Bonjour



What's in iOS?



Core Services

Collections

Core Location

Address Book

Net Services

Networking

Threading

File Access

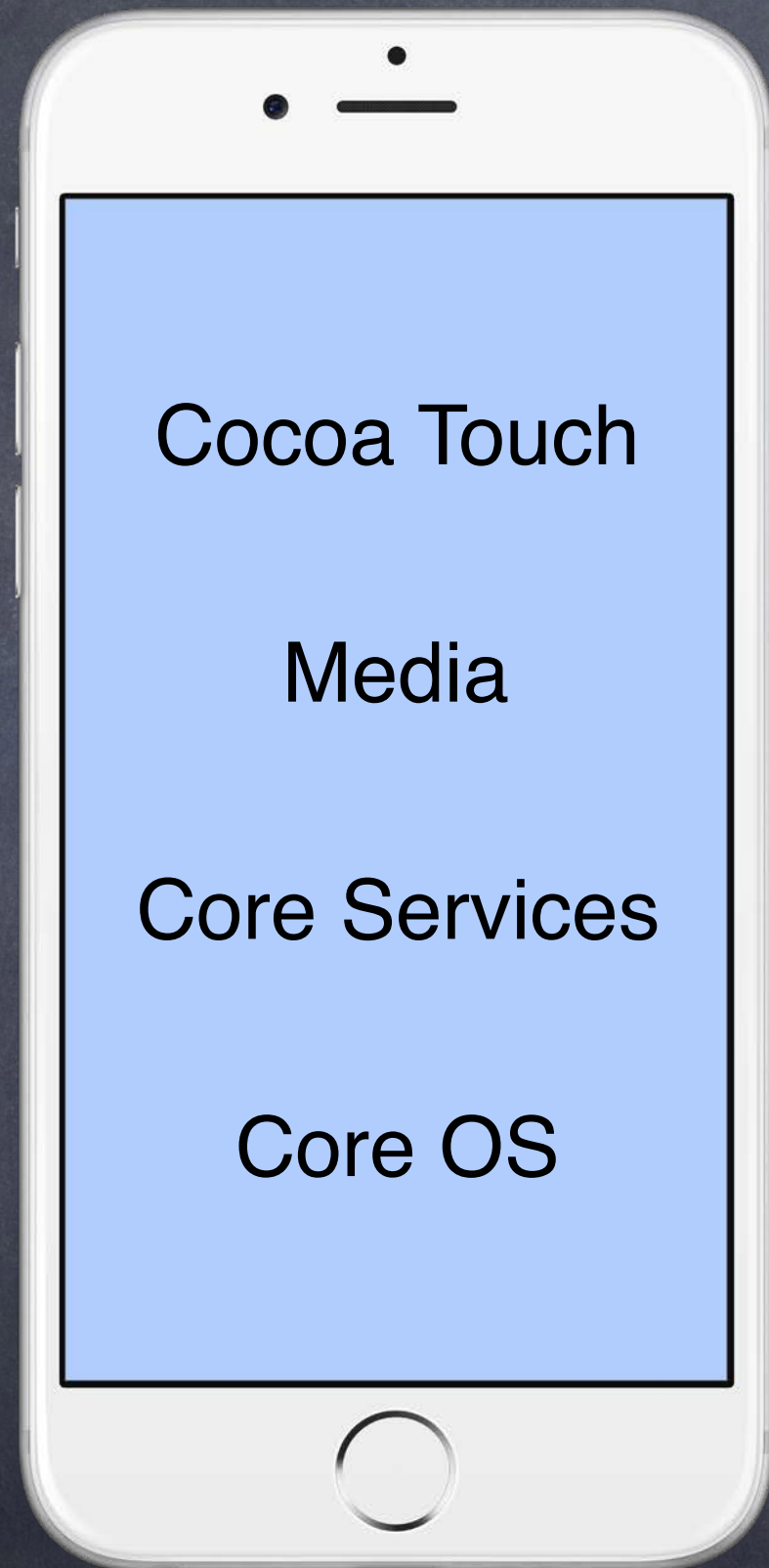
Preferences

SQLite

URL Utilities



What's in iOS?



Media

Core Audio

JPEG, PNG, TIFF

OpenAL

PDF

Audio Mixing

Quartz (2D)

Audio Recording

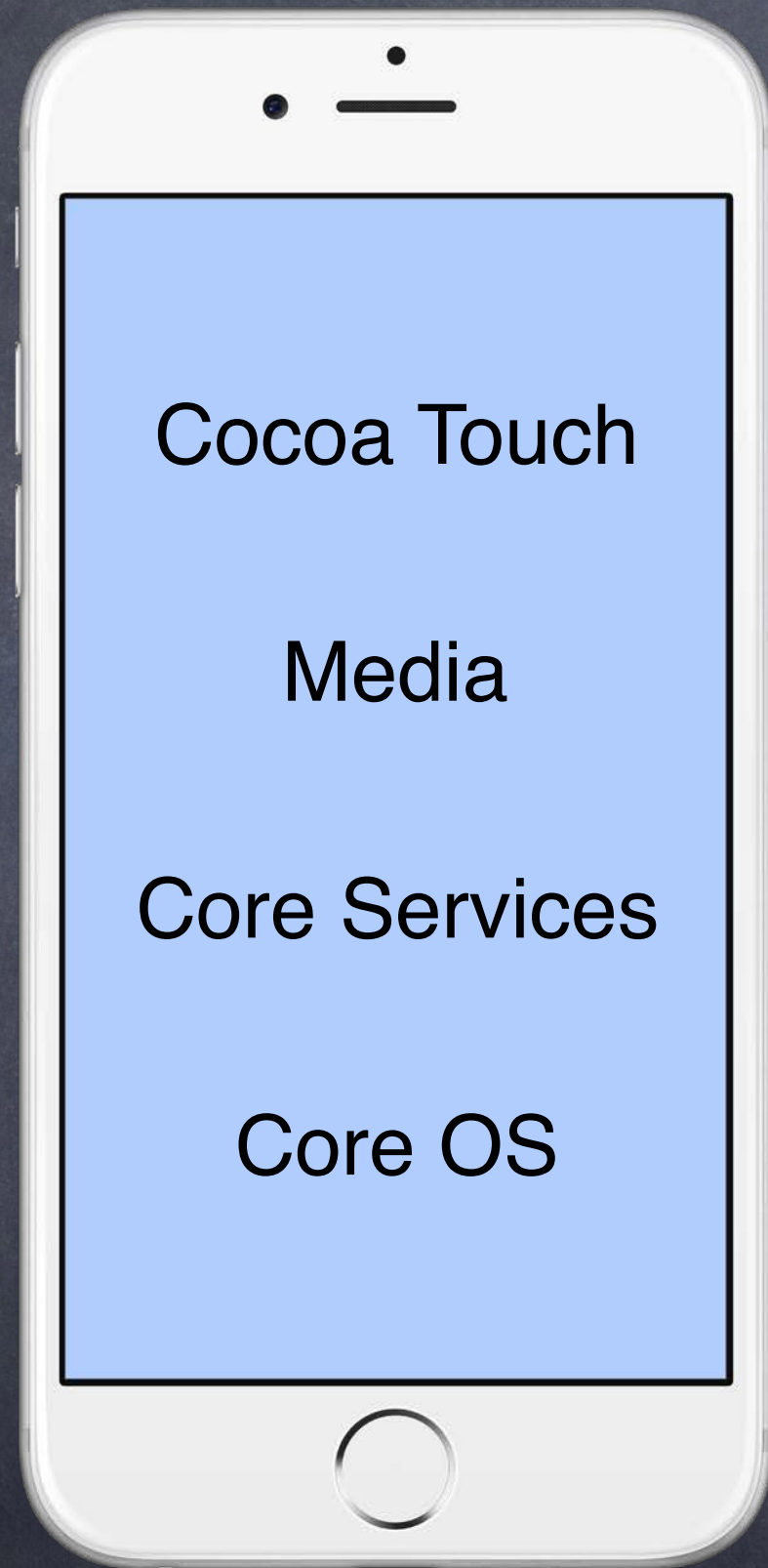
Core Animation

Video Playback

OpenGL ES



What's in iOS?



Cocoa Touch

Multi-Touch

Alerts

Core Motion

Web View

View Hierarchy

Map Kit

Localization

Image Picker

Controls

Camera

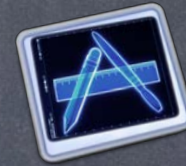


Platform Components

- Tools



Xcode 9



Instruments

- Language(s)

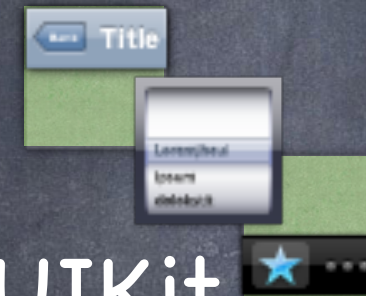
```
return cards.indices.filter { cards[$0].isFaceUp }.oneAndOnly
```

- Frameworks



Foundation

Core Data



UIKit

Core Motion

Map Kit

- Design Strategy

MVC



Demo

👁 Concentration Game

All this stuff can be very abstract until you see it in action.

We'll start getting comfortable with Swift 4 and Xcode 9 by building something right away.

Two part demo starting today, finishing on Wednesday.

👁 Today's topics in the demo ...

Creating a Project in Xcode 9, including building a UI and running in the iOS Simulator

Subclassing in Swift, including how to specify instance variables and methods

Connecting UI elements to invoke methods in our Swift code (actions)

print (outputting to the console using \() notation)

Connecting properties (instance variables) from our Swift code to the UI (outlets)

Accessing iOS documentation from our code

Automatically doing something every time a property's value changes

Array

Optionals

