Mobile Application Development in iOS

School of EECS

Washington State University

Instructor: Larry Holder
Course Overview

• Overview of iOS
• Language: Swift
• Development environment: Xcode
• Lifecycle: Design, implement, test, deploy
• Model-View-Controller (MVC) paradigm
Course Topics

- Swift
- Storyboarding and UI design
- Navigation and segues
- Tables
- Settings
- Alerts and notifications
- Gestures
Course Topics (cont.)

• Sensors
• Communications
• Data storage
• Graphics and animation
• Multimedia
• Apple Watch
Course Outcomes

• Proficient with iOS development environment
• Design, implement, test and deploy iOS app
• Able to use various iOS frameworks and kits
Course Details

• Course website
  – www.eecs.wsu.edu/~holder/courses/MAD/

• Prerequisites
  – Advanced Data Structures
  – Object-oriented design

• Grading (curved)
  – 12 homeworks (80%)
  – 1 final project (20%)
  – Submissions via Blackboard Learn (learn.wsu.edu)
Mobile App Development: Platforms

- iOS (Swift)
- Android (Java)
- Windows (C#)
  - Xamarin (C# → iOS/Android/Windows)
iOS Devices

iPad

iPhone

Apple Watch
Overview of iOS: Layers

- **Cocoa Touch Layer**: Frameworks that define appearance of app (e.g., GameKit, MapKit, UIKit)
- **Media Layer**: Graphics, Audio and Video technologies (e.g., CoreAudioKit, GLKit, SpriteKit, SceneKit)
- **Core Services Layer**: Services and basic types (e.g., Network, CoreData, CoreLocation, CoreMotion, CloudKit, HealthKit, HomeKit, StoreKit, WebKit)
- **Core OS Layer**: Low-level services (e.g., file I/O, networking, security)
Overview of iOS: Frameworks

- ARKit
- CloudKit
- HealthKit
- HomeKit
- GameKit
- MapKit
- SceneKit
- MusicKit
- WebKit
- UIKit
- CoreData
- CoreLocation
- CoreMotion
- CoreML
- WatchOS/WatchKit
- Over 50 more...
Development Environment: Xcode
First App

Mobile Application Development in iOS
School of EECS
Washington State University
Instructor: Larry Holder
Outline

• Xcode 11 projects

• Anatomy of an iOS app
  – iOS 13 (iPhone 6s or higher)

• Storyboarding

• Object attributes

• Auto-layout constraints
Xcode Projects

- File → New → Project
Xcode
Anatomy of an iOS App

- **AppDelegate**
  - Entry point
- **SceneDelegate**
  - Lifecycle management
- **ViewController**
  - Controls a view
- **Launch & Main storyboards**
- **Assets.xcassets**
  - Icons, images, videos
- **Info.plist**
  - App properties
App Lifecycle

- Unattached
- Foreground Inactive
- Foreground Active
- Suspended
- Background
- Foreground Inactive

Mobile Application Development in iOS 17
Storyboard
Storyboard: Adding Elements to View

1. Drag element into view.
Storyboard: Adding Elements to View

2. Add layout constraints.
Storyboard: Adding Elements to View

3. Connect element to view controller.

- Create IBOutlet to get/set properties of element
- Create IBAction to detect interaction with element
Storyboard: Adding Elements to View

```swift
import UIKit

class ViewController: UIViewController {

    @IBOutlet weak var imageView: UIImageView!
    
    @IBAction func flipTapped(_ sender: UIButton) {
    }

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view.
    }
}
```
Resources

• developer.apple.com

• swift.org

• www.raywenderlich.com

• stackoverflow.com