

# **Professional iOS Developer Course Batch II**



#13B, Htar Nar St.  
Ahlone Township, Yangon.  
Ph:+959965473380  
Email: [contact@techplussolution.com](mailto:contact@techplussolution.com)  
Web: [www.techplussolution.com](http://www.techplussolution.com)

# **IOS DEVELOPMENT PART I**

## **Introduction to iOS Development**

# **Table of Content**

## **I. Apple ECO System ( Practical lab and demo )**

## **II. Introduction to iOS Application**

1. Human Interface Guide Line
2. iOS App Architecture
3. Framework
4. App Store and App Distribution Channel

## **III.Mastering Standard iOS UI Element**

1. iOS Application Life Cycles
2. Creating Storyboard & Scenes
3. Containers: Windows & Views
4. Understanding UIView and its inheritance UI Elements
5. Autolayout & Constrains

## **IV.Swift Programming Part I**

1. Introduction
2. Variables & Constants
3. Swift Operators
4. Arrays
5. Dictionaries
6. Control Flow: Decision Making
7. Swift Loops
8. Swift Functions

## I. Apple Eco System

Understanding of Mac OS - Hand on Demo

Installation of Xcode & Mac Application -Hand on Demo

Creating Apple Developer Accounts -Hand on Demo

Using Xcode IDE & Simulator - Demo

First iOS App (Hello World )

## II.Introduction to iOS Application

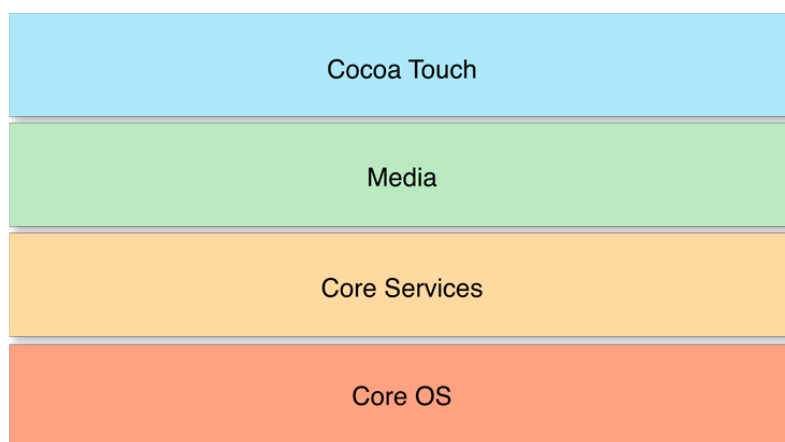
### 1. Human Interface Guide Line

- Refer to <https://developer.apple.com/design/human-interface-guidelines>
- Keynote (Summary )

### 2. iOS App Architecture

At the highest level, iOS acts as an intermediary between the underlying hardware and the apps you create. Apps do not talk to the underlying hardware directly. Instead, they communicate with the hardware through a set of well-defined system interfaces.

The implementation of iOS technologies can be viewed as a set of layers, which are shown in Figure I-1. Lower layers contain fundamental services and technologies. Higher-level layers build upon the lower layers and provide more sophisticated services and technologies.



**(Fig 2-1)**

### **3. Framework**

The iOS technologies are packaged as frameworks. Apple delivers most of its system interfaces in special packages called frameworks. A framework is a directory that contains a dynamic shared library and the resources (such as header files, images, and helper apps) needed to support that library. To use frameworks, you add them to your app project from Xcode.

eg. Foundation.framework, UIKit.framework, MapKit.framework

### **4. App Store and Distribution Channel**

The App Store is a digital distribution platform for mobile apps on iOS. The apps can be downloaded directly to iOS devices or onto a personal computer via iTunes and offers an update service, e.g. if one application is updated, it will notify to update to existing users and allow to install.

Enterprise Apps are in-house iOS apps developed for a certain organization and intended to use within organization

**Register as iOS Developer:**

**<https://developer.apple.com/programs/enroll/>**