Quadrant II – Transcript and Related Materials

Programme	**: Bachelor of Science (Second Year)**
Subject	**: Electronics**
Paper Code	**: ELS 104**
Paper Title	**: Smart phones app development**
Unit	**: I**
Module Name	**: iOS and Windows phone 8 operating system, Comparison of Android, iOS and Windows phone 8**
Module No	**: 02**
Name of the Presenter	**: Ms. Meryl Celian Lopes**

Notes:

iOS Introduction

- iOS(formerly iPhone OS) is a mobile operating system created and developed by Apple Inc..
- iOS is originally unveiled in 2007 for the iPhone and then extended to support other Apple devices such as the iPod Touch (September 2007) and the iPad(January 2010).
- WatchOS is the OS specifically for Apple Watch.

•Benefits of iOS

1. Longer support
2. Security and privacy
3. User friendly
4. High hardware and software integration

Architecture of iOS

Hierarchical structures

- Lower layers contain fundamental services and technologies.

- Higher level layer build upon the lower layers and provide more sophisticated services and technologies.
- The Kernel in IOS is based on same variant of the basic mac Kernel that is found in MAC OSX.
- iOS is the world's most advanced mobile operating system.
- In iOS there are four abstraction layers.

1. iPhone hardware

Hardware devices are **managed by iPhone OS** and provide the technologies needed for implementing native applications on the phone.

2. Core OS Layer

•Layer is **built from the low-level features** that most other technologies are built upon.

OSX Kernel

•This important layer is in charge of

Power Management-Conserves the power by shutting down any hardware features that are not currently being used.

Provides a handful of frameworks that your application can use directly, such as:

- Accelerate Framework.
- External Accessory Framework.
- Security Services framework.
- Local Authentication framework.

3. Core Services

•Provides **fundamental system services** required for apps.

•Peer-to-Peer services

•**iCloud storage** –it let your documents to upload it in iCloud storage directly.

•**Automatic Reference Counting** –manages the lifetime of application

•**SQLite libraries** -This lets you embed a lightweight SQL database into your application without running a separate remote database server process.

Using Frameworks

•**Address Book framework -**Provides access to the user's **Address Book contacts** on the iOS device

•**Core location framework:** Used for determining the **location and orientation** of an iOS device.

4. Media Layer

Provides iPhone OS with **audio, video, animation and graphics capabilities.**

•The **technologies** in this layer **make it easy** for you to **build apps** that **look and sounds great**.

•Based on various frameworks

5. Graphics Framework

- UIKit Graphics –high level support for designing images and also used for animating the content of your views.
- Core Graphics framework –It is the native drawing engine for iOS apps and gives support for custom 2D vector and image-based rendering.
- Core Animation –It is an initial technology that optimizes the animation experience of your apps.
- Core Images –gives advanced support for controlling video and motionless images.
- OpenGlES and GLKit–manages advanced 2D and 3D rendering.

6. Audio Framework

•Media Player Framework –It is a high level framework which gives simple use to a user's iTunes library and support for playing playlists.

•AV Foundation –It is an Objective C interface for handling the recording and playback of audio and video.

•Open AL–is an industry standard technology for providing audio.

7. Video Framework

•**AV Kit –**framework gives a collection of easy to use interfaces for presenting video.

•**AV Foundation –**gives advanced video playback and recording capability.

•**Core Media –**framework describes the low level interfaces and data types for operating media.

8. Cocoa Touch Layer

Sits at the top of the iPhone OS stack

•Framework in Cocoa touch are commonly used in iOS Apps.

1.UIKitframework

•Multitasking support (cut, copy and past support; app management, etc.)

2.Map kit framework

•gives a scrollable map that you can include into your user interface of app

3.EventKitframework

•gives view controllers for showing the standard system interfaces for seeing and altering calendar related events

•Support for **Multi-Touch capabilities**.

Windows OS Introduction:

Windows mobile is a mobile operating system developed by **Microsoft**.

•Windows mobile is designed for **use on smartphones and tablets running** on **ARM** processor architectures.

•The future of Windows mobile is still **uncertain.**

Architecture of Windows 8 phone OS:

- You can run two different types of apps: a **Windows Desktop app** and a **Windows Store app**.
- A **Windows Desktop app** is traditional Windows application with a new name. It is **run on the traditional desktop** and uses either native Win32 / COM API or Microsoft .NET Framework API for it's communication with the system.
- A **Windows Store app (former "Metro-style app")** is a new type of application that **only runs on Windows 8 devices**. A Windows Store app resembles the apps that are run on smartphones. It **uses new Windows Runtime (WinRT) application programming interfaces (APIs)**.

Core OS Services:

•The core operating system (OS) services consist of the **kernel (Windows CE 7) and other features**.

•Core OS services **enable low-level tasks, such as process, thread, and memory management.**

•**Basic device drivers** are also part of the Core OS services.

WinRT APIs:

•WinRT APIs **provide various features to enable the system services** such as **data communication, graphics and media and etc**.

•This API is written in C++. It is exposed out to other languages (in particular C++, C#, Visual Basic and Javascript).

Metro Style Applications:

•A Metro style app is an app built using HTML or XAML

•HTML /CSS (Java)

•XAML (C# or VB)

•XAML (C++ or C)

•Note that **this is cross-platform compatible** (Windows 8 will support ARM with metro apps) and that standard x86 apps built the "old way" will still continue to work, but are not considered metro apps, nor are they cross-platform compatible.

Comparison:

Parameters	Android	iOS	Windows
Developed by	Google	Apple	Microsoft
OS family	Linux	Darwin	Microsoft Windows
Language	C, C++, Java	C, C++, Objective C, Swift	C#, VB.NET, C++
Application Store	Google Play	App Store	Windows Phone Store
Cross Platforming	Android supports cross platforming	iOS don't support cross platforming	Windows support cross platforming
Browser	Chrome based	Mobile Safari	Internet explorer
Voice Assistant	NA	Siri	NA
Security	Susceptible to malware	Yes (no need for security suites)	Yes
Cloud support	Google sync	iCloud	Skydrive