**Practical 1**

**Operating Systems**

An operating system (OS) is a program that acts as an interface between the system hardware and the user. Moreover, it handles all the interactions between the software and the hardware

Some Functions of OS are:

1. Memory Management
2. Processor Management/Scheduling
3. Device Management
4. File Management
5. Security
6. Many Other Functions

There are many mobile operating systems currently in use, each with its own set of features and applications. Some of the most popular mobile operating systems are:

*Android:* Developed by Google, Android is the most widely used mobile operating system in the world. It is open-source, meaning that anyone can develop and publish applications for it. Android's app ecosystem is vast, with millions of apps available in the Google Play Store.

*iOS:* Developed by Apple, iOS is the operating system used on iPhones and iPads. It is a proprietary system, meaning that only Apple can develop and publish apps for it. iOS has a strong app ecosystem, with millions of apps available in the App Store.

*Windows Phone:* Developed by Microsoft, Windows Phone is a mobile operating system that is available on a limited number of devices. It has a smaller app ecosystem compared to Android and iOS, with many popular apps not available for the platform.

*BlackBerry:* Developed by BlackBerry Limited, BlackBerry is a mobile operating system that is primarily used on BlackBerry smartphones. It has a small but dedicated user base, and its app ecosystem is limited compared to other mobile operating systems.

Each of these mobile operating systems has its own strengths and weaknesses, and the best one for you will depend on your personal preferences and needs.

There are following 8 – components of Operating System:

1. Process Management
2. I/O Management
3. File Management
4. Network Management
5. Main Memory Management
6. Secondary Storage Management
7. Security Management
8. Command Interpreter System

**Differences**

Android, iOS, and Windows are all mobile operating systems that are used on smartphones and tablets. Each of these operating systems has its own unique features and capabilities, and they are designed to appeal to different types of users.

Android is developed by Google and is the most widely used mobile operating system in the world. It is open-source, meaning that anyone can develop and publish applications for it. Android's app ecosystem is vast, with millions of apps available in the Google Play Store. It is known for its customizable user interface and its support for a wide range of device types.

iOS is developed by Apple and is the operating system used on iPhones and iPads. It is a proprietary system, meaning that only Apple can develop and publish apps for it. iOS has a strong app ecosystem, with millions of apps available in the App Store. It is known for its intuitive user interface and its focus on security and privacy.

Windows Phone is developed by Microsoft and is a mobile operating system that is available on a limited number of devices. It has a smaller app ecosystem compared to Android and iOS, with many popular apps not available for the platform. It is known for its integration with other Microsoft products and services, such as Office and OneDrive.

In summary, the main differences between these mobile operating systems are their availability, app ecosystem, user interface, and focus on security and privacy.

**Native Applications**

A native application is a software program that is developed for a specific platform or device and is installed directly onto the device. Native applications are typically written in the programming language that is supported by the platform, such as Objective-C or Swift for iOS, Java for Android, or C# for Windows Phone.

Native applications are designed to take advantage of the features and capabilities of the specific platform or device that they are installed on. This means that they can provide a more seamless and user-friendly experience compared to non-native or web-based applications.

However, native applications are platform-specific, so they must be developed separately for each platform. This can be time-consuming and costly, especially for developers who want to create applications that are available on multiple platforms. Additionally, native applications must be downloaded and installed onto the device, which can be a barrier for some users.=

In summary, native applications are tailored to the specific platform or device that they are installed on, providing a user-friendly experience and access to the full range of device capabilities. However, they require separate development for each platform and must be downloaded and installed onto the device.

***Difference between Native Applications and Mobile Apps***

The terms "native application" and "mobile app" are often used interchangeably, but they refer to slightly different things. A native application is a software program that is developed for a specific platform or device and is installed directly onto the device. In contrast, a mobile app is a general term that refers to any application that is designed to be used on a mobile device, regardless of the platform or how it is installed.

In other words, all native applications are mobile apps, but not all mobile apps are native applications. For example, a web-based application that is accessed through a web browser on a mobile device is not a native application, but it is still considered a mobile app.

The main difference between native applications and mobile apps is how they are developed and installed. Native applications are written in the programming language that is supported by the platform, and they are installed directly onto the device. Mobile apps, on the other hand, can be developed using a variety of technologies and can be installed onto the device in different ways, such as through an app store or by downloading an installer file.

In summary, native applications are specific to a particular platform and are installed directly onto the device, while mobile apps are a general term that refers to any application that is designed to be used on a mobile device.

***Tools and Technologies required for different OS***

Android: To develop a native application for Android, you will need to use the Android Software Development Kit (SDK), which includes the tools and libraries that you need to build, test, and debug Android applications. You will also need to use the Java programming language and the Android Studio development environment.

iOS: To develop a native application for iOS, you will need to use the Xcode development environment and the iOS SDK, which include the tools and libraries that you need to build, test, and debug iOS applications. You will also need to use the Swift or Objective-C programming languages to develop your application.

Windows Phone: To develop a native application for Windows Phone, you will need to use the Windows Phone SDK, which includes the tools and libraries that you need to build, test, and debug Windows Phone applications. You will also need to use the C# programming language and the Visual Studio development environment.

BlackBerry: To develop a native application for BlackBerry, you will need to use the BlackBerry Native Development Kit (NDK), which includes the tools and libraries that you need to build, test, and debug BlackBerry applications. You will also need to use the C/C++ programming languages to develop your application.

***Advantages of Native Applications***

Improved performance: Native applications are designed to take advantage of the features and capabilities of the specific platform or device that they are installed on. This means that they can provide a faster and more responsive user experience compared to non-native applications.

Seamless integration with device features: Native applications have access to the full range of device capabilities, such as the camera, GPS, and sensors. This allows them to provide a more seamless and user-friendly experience compared to non-native applications, which may not have access to these features.

Better security: Native applications are installed directly onto the device, so they are more secure compared to non-native applications that are accessed through a web browser. This makes them suitable for handling sensitive information, such as financial or personal data.

Improved user experience: Native applications are designed specifically for the platform or device that they are installed on. This allows them to provide a user interface and user experience that is tailored to the device and is more intuitive and user-friendly compared to non-native applications.

***Disadvantages of Native Applications***

Platform-specific development: Native applications are written in the programming language that is supported by the specific platform or device that they are installed on. This means that they must be developed separately for each platform, which can be time-consuming and costly, especially for developers who want to create applications that are available on multiple platforms.

Limited flexibility: Native applications are tied to the specific platform or device that they are installed on, so they cannot be easily transferred or used on other platforms or devices. This can limit their flexibility and make it difficult for users to access their applications on multiple devices.

Dependence on app stores: Native applications must be downloaded and installed onto the device through an app store, such as the Google Play Store or the App Store. This can be a barrier for some users, and it also means that the availability of native applications is dependent on the app store's policies and procedures.

Limited access to updates: Native applications are updated through the app store, so users must go through the app store to access the latest version of the application. This can make it difficult for developers to release updates and new features quickly, and it can also make it difficult for users to access the latest version of the application.

*By:*

**20BCM015**

**Devdeepsinh Zala**