**Assignment-1**

1. **What is Android explain version and application?**

* **Android:** Android is an operating system developed by Google for mobile devices, such as smartphones and tablets. It is based on the Linux kernel and is designed primarily for touchscreen-based mobile devices. Android provides a robust framework for building and running applications, allowing developers to create a wide range of software, from simple apps to complex, feature-rich applications.
* **Android Versions:** Android comes in various versions, with each version having a codename and a version number. These versions represent different iterations and improvements of the Android operating system. For example, some well-known Android versions include:
* Android 4.4 (KitKat)
* Android 5.0 (Lollipop)
* Android 6.0 (Marshmallow)
* Android 7.0 (Nougat)
* Android 8.0 (Oreo)
* Android 9.0 (Pie)
* Android 10
* Android 11
* Android 12 (as of my last knowledge update in September 2021)
* Each Android version brings new features, enhancements, and improvements to the operating system, as well as changes to the Android API (Application Programming Interface) that developers use to build apps. Developers need to consider the Android version when developing apps to ensure compatibility with different devices. Android applications, often simply called "apps," are software programs developed to run on Android devices. These apps can serve a wide range of purposes, from productivity tools and social media platforms to games and utility applications. Android apps are typically written in programming languages like Java or Kotlin.
* **Android applications consist of various components, including:**
* **Activities:** These represent the screens or user interfaces of the app.
* **Services:** Background processes that can run independently of the app's user interface.
* **Broadcast Receivers:** Components that respond to system-wide broadcast announcements.
* **Content Providers:** Components that manage and share data with other apps.
* Apps are distributed through the Google Play Store and other app stores, and users can download and install them on their Android devices.

1. **Explain the history of Android in short?**

* **The history of Android is a fascinating journey that spans several decades and involves multiple key players and milestones. Let's explore the history of Android in detail:**
* **Early Beginnings (1980s-2003):** The story of Android begins with several individuals and companies laying the groundwork for what would eventually become the Android operating system. Notable contributions include:
* In the 1980s and 1990s, visionaries like **Andy Rubin, Rich Miner, and Nick Sears** started working on mobile operating systems.
* In 2003, Andy Rubin founded Android Inc., a company with the goal of developing a new, advanced operating system for digital cameras. This venture was backed by Google, which saw the potential for mobile applications.
* **Acquisition by Google (2005):** In August 2005, Google acquired Android Inc. The exact motivations behind the acquisition were not initially clear, but it marked a significant turning point in Android's history.
* **Android Beta (2007):** In November 2007, the Open Handset Alliance (OHA) was formed, consisting of several major technology companies, including Google, HTC, Samsung, and more. Together, they announced the Android platform, an open-source operating system for mobile devices.
* **First Android Device (2008**): In September 2008, the HTC Dream (also known as the T-Mobile G1) was released. It was the first commercially available device running the Android operating system. Android 1.0 was the initial version, and it laid the foundation for subsequent versions.
* **Android Market (2008):** In October 2008, the Android Market (now Google Play Store) was launched, allowing users to download and install applications on their Android devices.
* **Version Updates (2009-2010):** Android saw rapid development and expansion during this period, with the release of Android 2.0 (Eclair), Android 2.2 (Froyo), and Android 2.3 (Gingerbread). Each update introduced new features and improvements.
* **Tablet and Honeycomb (2011):** Android 3.0 (Honeycomb) was introduced, specifically designed for tablet devices. It marked Android's foray into the tablet market.
* **Android's Dominance (2012-2014):** Android became the dominant mobile operating system worldwide during this period. Android 4.0 (Ice Cream Sandwich), Android 4.1 (Jelly Bean), and Android 4.4 (KitKat) were notable releases.
* **Material Design and Lollipop (2014):** Android 5.0 (Lollipop) brought the Material Design language, which focused on a more cohesive and visually appealing user interface.
* **Nougat and Oreo (2016-2017):** Android 7.0 (Nougat) and Android 8.0 (Oreo) introduced improvements in performance, security, and multitasking capabilities.
* **Android Pie and Beyond (2018-2021):** Android 9.0 (Pie), Android 10, and Android 11 continued to refine the user experience, enhance privacy controls, and integrate artificial intelligence features.
* **Android's Versatility (2020s):** Android expanded its presence beyond smartphones and tablets, powering devices like smart TVs, smartwatches, and automotive infotainment systems.
* This is Basic on information available up to September 2021. Android's history continues to evolve, with new versions and innovations being introduced regularly.

1. **Describe Background about Mobile technology?**

* **Mobile technology** has a rich and rapidly evolving history, transforming the way we communicate, work, and live. Here's a brief background on the development of mobile technology:
  1. **Early Mobile Phones (1970s-1980s):** The concept of mobile phones emerged in the 1970s, and the first commercially available mobile phone, the Motorola DynaTAC 8000X, was introduced in 1983. These early mobile phones were large, expensive, and primarily designed for voice calls.
  2. **2G and Text Messaging (1990s):** The 1990s saw the introduction of 2G (second generation) cellular networks, which allowed for digital voice transmission and the introduction of text messaging **(SMS).** Nokia's iconic "Nokia 3310" and "Nokia 5110" became popular during this era.
  3. **Mobile Internet (Early 2000s):** The early 2000s marked the transition to mobile internet access. WAP (Wireless Application Protocol) enabled limited web browsing on mobile devices. Nokia's "Nokia 7110" was one of the first phones with WAP capabilities.
  4. **3G and Multimedia (2000s):** The advent of 3G networks brought faster data speeds, enabling multimedia features like video calling and mobile web browsing. The introduction of the iPhone in 2007 by Apple revolutionized the smartphone industry.
  5. **App Stores and Smartphones (Late 2000s):** Apple's App Store, launched in 2008, and Google's Android Market (now Google Play Store) paved the way for a new era of mobile applications. Smartphones, such as the iPhone and Android devices, became more prevalent, offering a wide range of apps for various purposes.
  6. **4G LTE and Mobile Data (2010s):** The rollout of 4G LTE (Long-Term Evolution) networks provided significantly faster data speeds, enabling high-definition video streaming, online gaming, and more. Mobile data usage skyrocketed.
  7. **Mobile Ecosystems (2010s):** Apple's iOS and Google's Android emerged as dominant mobile operating systems, each with its ecosystem of apps, services, and developer communities.
  8. **Rise of Mobile Commerce (2010s):** Mobile devices became essential for e-commerce, with mobile payments, mobile shopping apps, and digital wallets gaining popularity.
  9. **5G and Beyond (2020s):** The rollout of 5G networks began, promising even faster data speeds, lower latency, and greater connectivity. 5G is expected to drive innovations in IoT (Internet of Things) and augmented reality.
* Mobile technology has become an integral part of modern life, enabling remote work, social networking, navigation, entertainment, health monitoring, and much more. The convergence of mobile devices with other technologies, such as AI and IoT, continues to shape our future.

1. **Why Android is called for Mobile world?**

* **Android is called the mobile world** because it is the most popular mobile operating system in the world. As of August 2023, Android has a market share of over 70% of the global smartphone market. This means that out of every 10 smartphones sold in the world, 7 are Android phones. There are a number of reasons why Android is so popular. It is an open source operating system, which means that anyone can develop apps for it. This has resulted in a huge ecosystem of Android apps, with over 3 million apps available on the Google Play Store.
* **Market Dominance:** Android holds the largest market share among mobile operating systems. It powers the majority of smartphones and tablets worldwide. This widespread adoption makes Android synonymous with the mobile device ecosystem.
* **Open Source Nature:** Android is an open-source operating system. This means that device manufacturers and developers have the freedom to modify and customize it to suit their needs. This openness has led to a diverse range of Android devices, contributing to its ubiquity in the mobile world.
* **Versatility:** Android is not limited to smartphones. It is used in various types of devices, including tablets, Smartwatches, smart TVs, in-car infotainment systems, and more. Its adaptability and versatility across different form factors further establish its presence in the mobile world.
* **App Ecosystem:** The Android ecosystem includes the Google Play Store, which offers a vast library of mobile applications for various purposes. This extensive app ecosystem plays a significant role in making Android the go-to platform for mobile devices.
* **Global Reach:** Android's presence extends to markets around the world. It's not limited to a particular region or country, making it a global standard for mobile technology.
* **Accessibility:** Android's open nature has made it accessible to a wide range of device manufacturers, including both established companies and emerging players. This accessibility has contributed to its popularity.
* **Customization:** Android allows users and device manufacturers to customize the user experience to a great extent. This flexibility has enabled Android to cater to diverse user preferences and market demands.
* **Innovation:** Android has been at the forefront of mobile innovation, introducing new features, technologies, and form factors. This continuous innovation keeps Android at the center of the evolving mobile world.
* **Google Services:** Android integrates with various Google services, including Google Search, Google Maps, YouTube, and more. These services are an integral part of the mobile experience for many users.
* Android is often associated with the "mobile world" because it is the leading mobile operating system in terms of market share, and it has established itself as a versatile, open-source platform that powers a wide range of mobile devices and offers a vast ecosystem of mobile applications. Its widespread adoption and global reach make it a key player in the mobile technology landscape.

1. **What is mean by Native Android Application?**

* **A native Android application,** often referred to as a "native app," is a mobile application that is developed specifically for the Android operating system (OS) using the platform's native development tools and programming languages. In the context of Android, native apps are typically written in Java or Kotlin, which are the officially supported programming languages for Android development.
* **Here are the key characteristics and advantages of native Android applications:**

|  |
| --- |
|  |

* **Performance:** Native apps are optimized for the Android platform, which means they tend to offer superior performance compared to apps developed using cross-platform or web-based technologies. They can directly access the device's hardware and take full advantage of its capabilities.
* **Access to Native Features:** Native Android apps have full access to the device's native features and APIs (Application Programming Interfaces). This means they can utilize the camera, GPS, sensors, contacts, and other hardware and software functionalities, resulting in feature-rich and highly integrated applications.
* **Native User Experience:** Native apps adhere to Android's design guidelines and user interface components, providing a consistent and native user experience. This makes them look and feel like a natural part of the Android ecosystem, ensuring user familiarity and comfort.
* **Offline Functionality:** Native Android apps can offer robust offline functionality, allowing users to use certain features or access content even when they are not connected to the internet.
* **Security:** Native apps can leverage the security features provided by the Android OS, ensuring a high level of security for both data and user interactions.
* **App Stores:** Native Android apps can be published and distributed through app stores like the Google Play Store, making them easily accessible to a vast audience of Android users.
* **Optimized for Android Updates:** Native app developers can quickly adapt their apps to support new Android versions and features, ensuring compatibility and staying up to date with the latest OS releases.
* It's important to note that native development requires expertise in Java or Kotlin, and it often involves more development effort for creating separate codebases for different platforms (e.g., Android and iOS). However, the advantages in terms of performance, access to native features, and a seamless user experience make native Android app development a preferred choice for many developers and organizations, especially when targeting the Android platform exclusively.

1. **What are the features / Tools of Android SDK Explain?**

* **The Android SDK (Software Development Kit)** is a comprehensive set of tools, libraries, and resources that developers use to create Android applications. It provides everything needed to develop, test, and deploy Android apps. Here are key Features/Tools of the Android SDK**:**
* **Android Studio:** Android Studio is the official integrated development environment (IDE) for Android app development. It offers a powerful code editor, debugging tools, an emulator for testing, and a visual layout editor for designing user interfaces.
* **Android Emulator:** The Android SDK includes an emulator that allows developers to test their apps on virtual Android devices with various screen sizes, resolutions, and Android versions. This helps ensure that the app works correctly on a wide range of devices.
* **API Libraries:** Android SDK provides a rich set of API libraries that cover various aspects of app development, including graphics, multimedia, data storage, networking, user interface, and more. These libraries simplify common tasks and provide a consistent framework for building Android apps.
* **Device Profiles:** Android SDK includes device profiles and resources for different screen densities and sizes, which helps developers create responsive user interfaces that adapt to various Android devices.
* **Debugging and Profiling Tools:** Android SDK offers robust debugging and profiling tools for identifying and fixing issues in apps. Developers can use tools like the Android Debug Bridge (ADB) for debugging and tools like Android Profiler for performance optimization.
* **Testing Framework:** The Android SDK includes testing frameworks like JUnit and Espresso for automated testing of Android applications. These frameworks help ensure the reliability and correctness of an app's functionality.
* **Documentation and Sample Code:** Android SDK provides extensive documentation and sample code to help developers learn the platform, understand best practices, and implement specific features. This valuable resource accelerates the development process.
* **Gradle Build System:** The Android SDK uses the Gradle build system to manage project dependencies, automate build processes, and create APK (Android Package) files for distribution. It simplifies project configuration and management.
* **Support for Multiple Android Versions:** Android SDK allows developers to target a wide range of Android versions, ensuring backward and forward compatibility for their apps. Developers can specify the minimum and target Android versions to reach a broader audience.
* These features of the Android SDK provide developers with the tools and resources they need to create high-quality, feature-rich Android applications that are compatible with a wide range of devices and Android versions.

1. **Draw and explain Android Architecture?**

|  |
| --- |
| android architecture |

* **Android Architecture:**
* **Linux Kernel:** Linux is the heart of Android architecture. It provides a level of abstraction between the hardware devices and the upper layers of the Android software stack. The Android operating system is based on the Linux kernel. The Linux kernel is responsible for various device drivers such as Camera driver, Display driver, Bluetooth driver, Keypad driver, Memory management, Process management, Power management, etc.
* **Native Libraries:** The native libraries such as Media, WebKit, SQLite, OpenGL, FreeType, C Runtime library (libc) etc. are situated on the top of a Linux kernel. Media library is responsible for playing and recording audio and video formats, FreeType is for font support, WebKit is for browser support, SQLite is for database, SSL is for Internet security etc.
* **Android Runtime:** Android Runtime is the third section of the architecture and situated on the second layer from the bottom. Android Runtime includes core libraries and Dalvik Virtual Machine (DVM) which is responsible to run android application. Dalvik Virtual Machine (**DVM**) is like Java Virtual Machine (**JVM**) in Java, but DVM is optimized for mobile Devices. DVM makes use of the Linux core features like memory management and multi-threading, which are essential in the Java language. DVM provides fast performance and consumes less memory.
* **Application Framework**: Application framework is situated on the top of the Native libraries and Android runtime. Android framework provides a lot of classes and interfaces for Android application development and higher level services to the applications in the form of Java classes. It includes Android API's such as Activity manager, Window manager, Content Provider, Telephony Manager, etc. Activity manger is responsible for controlling all the aspects of the application lifecycle and activity stack, Content provider is responsible for allowing the applications to publish and share the data with the other applications, View system is responsible for creating application user interfaces, etc.
* **Applications:** Applications are situated on the top of the Application framework. The applications such as Home, Contact, Alarm, Calender, Camera, Browsers, etc. use the Android framework which uses Android runtime and libraries. Android runtime and Native libraries use Linux kernel. The user can write his/her application to be installed on this layer only.

1. **What is mean by application framework?**

* **The Application Framework** is a key component of the Android operating system that provides a structured and organized environment for developing Android applications. It offers a set of high-level building blocks, APIs (Application Programming Interfaces), and services that developers can leverage to create feature-rich and interactive mobile applications. Here's a more detailed explanation of the Android Application Framework:
* **Component-Based Architecture:** The Application Framework follows a component-based architecture, where the building blocks of Android applications are defined as components. The main components include:
* **Activities:** These represent individual screens or UI elements within an app.
* **Services:** Services are background processes that can perform tasks independently of the UI, such as playing music or handling network requests.
* **Content Providers:** Content providers allow apps to share and access data with other apps, enabling data exchange and storage.
* **Broadcast Receivers:** These components listen for system-wide or app-specific broadcast messages and respond accordingly.
* **UI Toolkit:** The Application Framework includes a rich and flexible UI toolkit that provides a wide range of user interface elements and widgets. Developers can use these elements to create visually appealing and responsive user interfaces for their applications. This toolkit supports layouts, views, and event handling for user interactions.
* **Intent System:** Android's Intent system allows components within an app and between different apps to communicate and interact. Intents are used to initiate activities, start services, and send broadcast messages. This system enables seamless inter-component communication and action triggering.
* **Resource Management:** The framework includes resource management capabilities, allowing developers to separate resources like images, layout files, and localization strings from the code. This separation simplifies the localization and adaptation of applications to different device configurations.
* **Activity Lifecycle Management:** Android manages the lifecycle of activities, ensuring that apps respond appropriately to changes in state, such as when the user navigates away from an app or when the device orientation changes. Developers can override specific methods to control the behavior of activities during their lifecycle.
* **Background Processing:** Services and background threads provided by the framework allow developers to perform time-consuming tasks without blocking the UI. This ensures a smooth user experience while running tasks like downloading data or processing computations in the background.
* **Content Providers:** The Content Provider component facilitates data sharing and storage among apps. It defines a structured way to access and manipulate data, making it possible for apps to securely share data with each other.
* **Notification Management:** Android's framework includes a Notification Manager, which allows apps to create and manage notifications. Notifications are used to inform users about events or updates within an app, even when the app is not actively in use.
* **Integrated Services:** Android provides integrated services for common functionalities, such as location services (via GPS or network triangulation), multimedia playback, and access to device sensors (accelerometer, gyroscope, etc.).

The Android Application Framework is a comprehensive set of tools and services that streamline the development process of Android applications. It simplifies tasks like UI design, inter-component communication, background processing, and resource management, allowing developers to focus on creating innovative and user-friendly applications for Android devices.

1. **What is mean by Android library and run time?**

* **Android Library:** An Android library is a collection of pre-written code and resources that developers can include in their Android applications to perform specific tasks or add functionality without having to write everything from scratch. Libraries are designed to be reusable, which helps save development time and effort.
* Android libraries can encompass a wide range of functionalities. For example, a library might provide image loading and caching, networking capabilities, user interface components, database management, or even machine learning capabilities. Developers can add these libraries to their Android projects by including them as dependencies in their build files.
* Popular Android libraries include Retrofit for networking, Picasso for image loading, Gson for JSON serialization/deserialization, and AndroidX libraries for UI components and architecture components.
* **Android Runtime (ART):** Android Runtime (ART) is the runtime environment within the Android operating system responsible for executing Android applications. It is part of the Android operating system and is responsible for running the compiled bytecode of Android apps.
* ART uses a technique known as Ahead-of-Time (AOT) compilation. When an Android app is installed, the app's bytecode (compiled Java or Kotlin code) is converted into native machine code for the specific device architecture. This native code is stored on the device, improving app performance by reducing the need for just-in-time (JIT) compilation during runtime.
* ART replaced the Dalvik Virtual Machine (DVM) in later versions of the Android operating system (Android 5.0 Lollipop and newer). ART offers several advantages over DVM, including improved app performance, reduced memory usage, and enhanced security.

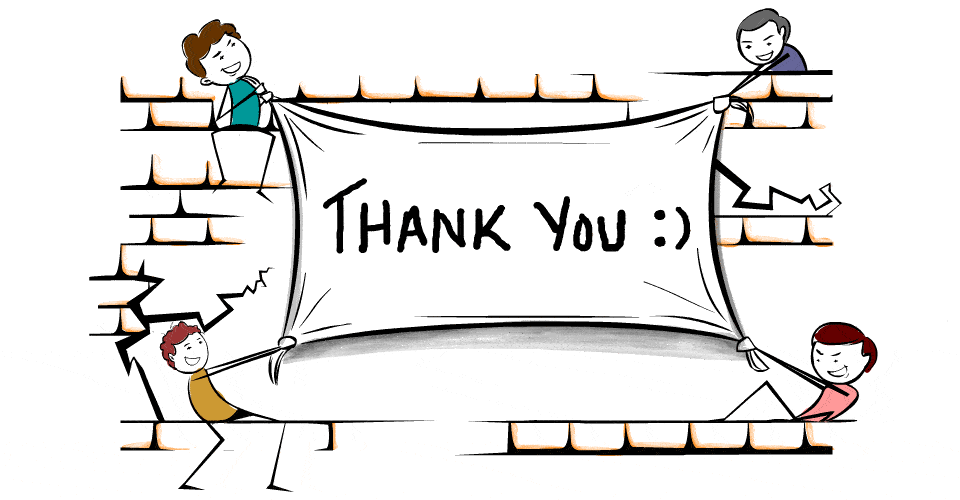
1. **Explain in short Dalvic vertual machine, running and debugging?**

* **Dalvik Virtual Machine (DVM):**
* **Purpose:** The Dalvik Virtual Machine was the original runtime environment used by Android to execute Android applications. It played a crucial role in running Android apps efficiently on mobile devices.
* **Bytecode Execution:** Android developers write their applications in Java or Kotlin. When these applications are compiled, they generate bytecode files (DEX files) that are executed by the DVM. Each Android application had its dedicated instance of the DVM.
* **Optimized for Mobile Devices:**
* **Memory Efficiency:** DVM was designed to be memory-efficient. It used a register-based architecture that required fewer memory resources compared to the stack-based architecture of the Java Virtual Machine (JVM).
* **Battery Life:** By conserving memory and CPU resources, DVM contributed to improved battery life on mobile devices.
* **Just-In-Time (JIT) Compilation:** DVM used Just-In-Time (JIT) compilation for performance optimization. Instead of interpreting bytecode directly, DVM compiled it into native machine code at runtime, which sped up execution.
* **Garbage Collection:** DVM included a garbage collector to manage memory and automatically reclaim memory occupied by objects that were no longer in use. This helped prevent memory leaks and ensure efficient memory usage.
* **Running Android Applications:**
* **Application Packaging:** Android applications are packaged as APK (Android Package) files. These files contain compiled code, resources, and manifest information.
* **APK Installation:** Users or developers install APK files on Android devices. During installation, the APK is unpacked, and its components are stored on the device.
* **Launching an App:** When a user launches an Android app, the following steps typically occur:
* The Android OS starts the app's main activity, which is declared in the app's manifest file.
* The DVM (or ART in newer versions of Android) loads the DEX files containing the app's bytecode.
* The app's code is executed by the DVM, creating the user interface and responding to user interactions.
* **Multi-Process Model:** Android apps can run in separate processes, which are isolated from each other. This helps in maintaining stability and security.
* **Debugging Android Applications:**
* **Android Debug Bridge (ADB):** ADB is a command-line tool that facilitates communication between a developer's computer and an Android device or emulator. It's a crucial tool for debugging Android apps.
* **Android Studio:** Android Studio, the official integrated development environment (IDE) for Android development, provides robust debugging tools. Developers can set breakpoints, inspect variables, step through code, and analyze app performance.
* **Logcat:** Logcat is a logging system in Android that allows developers to view real-time logs generated by the Android system and their own applications. It's invaluable for diagnosing issues and understanding app behavior.
* **Emulators:** Developers can use Android emulators provided by Android Studio for testing and debugging apps on virtual devices. Emulators offer debugging features similar to physical devices.
* **Third-Party Debuggers:** Some third-party debugging tools and IDEs also support Android app debugging. For example, JetBrains' IntelliJ IDEA and Microsoft's Visual Studio can be configured for Android development and debugging.

1. **What is OHA (Open Handset alliance Alliance)?**

* **The Open Handset Alliance (OHA)** is a consortium of technology companies and organizations that joined forces to develop and promote the Android operating system for mobile devices. The primary goal of the OHA was to create an open and standardized platform for smartphones that would be free from the restrictions imposed by proprietary operating systems.
* **Key points about the Open Handset Alliance (OHA) in Android:**
* **Formation:** The OHA was officially announced on November 5, 2007. It was initiated by Google, which unveiled the Android platform as the cornerstone of the alliance.
* **Members:** The OHA consisted of a diverse group of companies, including mobile device manufacturers, software companies, semiconductor manufacturers, and mobile network operators. Notable members included Google, HTC, Samsung, LG, Motorola, Intel, Qualcomm, and T-Mobile, among others.
* **Android Open Source Project (AOSP):** Under the OHA, Google launched the Android Open Source Project (AOSP), which made the Android source code freely available to the public. This open-source approach allowed device manufacturers and developers to customize and build upon the Android platform.
* **Shared Vision:** The OHA members shared a vision of creating a platform that would promote innovation, provide a consistent user experience across devices, and increase competition in the mobile industry.
* **Android Ecosystem:** The OHA played a pivotal role in shaping the Android ecosystem. It led to the development of Android as a widely used and versatile operating system, powering a wide range of devices beyond smartphones, including tablets, smart TVs, and automotive infotainment systems.
* **Google's Role:** While the OHA members collaborated on the development of Android, Google had a central role in its direction and management. Google's services, such as Google Search and Google Maps, were integrated into the Android platform.
* **Android's Success:** Android's open and adaptable nature, along with the support of OHA members, contributed to its rapid growth and dominance in the mobile device market.
* It's important to note that while the OHA played a significant role in the early development and promotion of Android, the alliance's activities have evolved over time. Google has continued to oversee the Android platform's development and direction, and Android has become an integral part of the Google ecosystem.

**(PPT Give you more understanding than PDF)** The material for the PDF has been compiled from various sources such as books, tutorials (offline and online), lecture notes, several resources available on Internet. The information contained in this PDF is for general information and education purpose only. While we endeavor to keep the information up to date and correct, **we make no representation of any kind about the completeness and accuracy of the material.** The information shared through this PDF material should be used for educational purpose only.

****