



Unit 2: Installation and configuration of android.

a. Explain Android SDK in detail.

The Android Software Development Kit (SDK) is a set of tools and resources provided by Google for developers to create applications for Android devices. Android SDK is a comprehensive package that equips developers with everything they need to create, test, and deploy Android applications, making the development process smoother and more efficient.

It includes-

1. **Tools for Development:** The SDK includes tools like the Android Studio IDE (Integrated Development Environment), which is a special software where developers write and build their Android apps.
2. **Libraries and APIs:** It provides libraries and APIs (Application Programming Interfaces) that developers use to access various functionalities of the Android operating system, such as handling user interfaces, managing device resources, and interacting with hardware features.
3. **Emulator for Testing:** The SDK includes an emulator, which is a virtual Android device that allows developers to test their apps on different versions of Android without needing a physical device.
4. **Documentation:** It comes with documentation that explains how to use different features and functionalities, helping developers understand how to build effective and efficient Android applications.
5. **Sample Code:** The SDK provides sample code that developers can use as a reference or starting point for their own projects. This helps speed up development by offering examples of common coding patterns.
6. **Debugging and Profiling Tools:** Debugging tools are included to identify and fix issues in the code, and profiling tools help developers optimize the performance of their applications.
7. **Build and Deployment Tools:** The SDK includes tools for building the final version of an app and deploying it to a device or uploading it to the Google Play Store for distribution.
8. **Compatibility Packages:** Android SDK often includes compatibility packages to ensure that apps work across different Android versions and devices.

b. List and explain Android Developer Tools (ADT).

Android developer tools refer to a set of software and resources that aid developers in creating, testing, and debugging Android applications. These tools collectively provide a comprehensive environment for Android developers, offering everything from code development and testing to debugging and performance optimization.

Here are key Android developer tools:

1. **Android Studio:** Android Studio is the official Integrated Development Environment (IDE) designed for Android app development. It provides a user-friendly interface for



writing, debugging, and testing Android applications. Android Studio includes advanced features such as a visual layout editor, emulator support, and powerful debugging tools.

2. **Android Software Development Kit (SDK):** The Android SDK is a comprehensive set of tools, libraries, and APIs that developers use to build Android applications. It includes everything needed to compile code, access device features, and manage the entire development lifecycle.

3. **ADB (Android Debug Bridge):** ADB is a versatile command-line tool that facilitates communication between a computer and an Android device or emulator. Developers use ADB for various tasks, including installing and debugging apps, copying files between devices, and accessing the device shell for advanced troubleshooting.

4. **Android Emulator:** The Android Emulator is a virtual device that emulates the behaviour of a real Android device. It allows developers to test their apps on different Android versions and screen sizes without needing physical devices. The emulator is essential for ensuring app compatibility across a range of configurations.

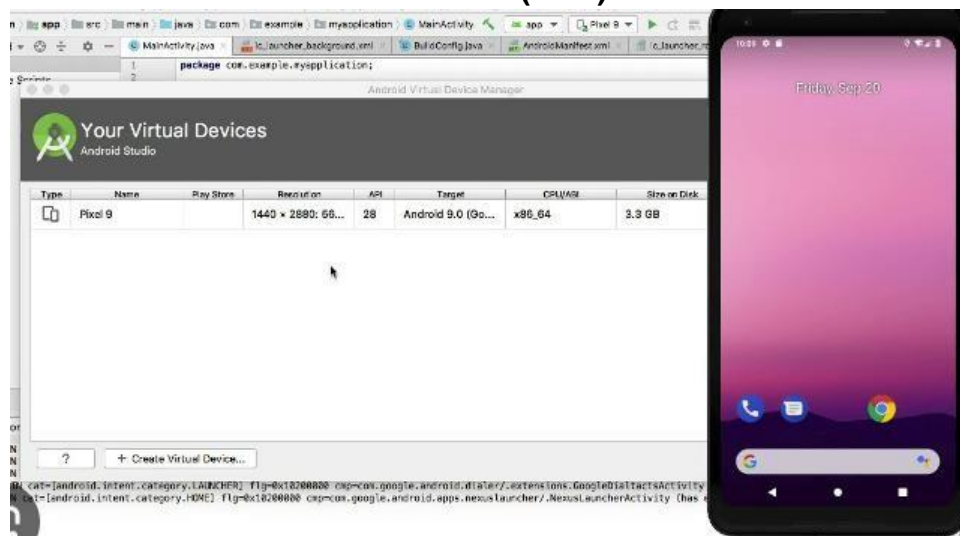
5. **DDMS (Dalvik Debug Monitor Server):** DDMS is a set of debugging tools integrated into Android Studio. It includes features for profiling system resource usage, capturing screenshots, and simulating phone calls and messages. These tools are valuable for analysing and optimizing app performance during development.

Explain ADT plugins in detail.

Its plugins include:

- **Project Creation and Management:** ADT plugins offer wizards and tools to create and manage Android projects in Eclipse. Developers can easily set up new projects, specify project properties, and manage dependencies.
- **Layout Editor:** ADT plugins include a graphical layout editor that allows developers to design app layouts visually. With drag-and-drop functionality, developers can easily arrange UI components, set their attributes, and preview the layout in different configurations.
- **XML Editors:** ADT plugins provide XML editors for various Android resource files, such as layouts, strings, styles, and manifests. These editors offer syntax highlighting, code completion, and validation to assist developers in editing XML files effectively.
- **Build and Debug Tools:** ADT plugins seamlessly integrate with the Android SDK (Software Development Kit) and provide tools for building, packaging, and deploying Android applications. They allow developers to compile their code, create APK (Android Package) files, and install the app on devices or emulators directly from Eclipse.

c. Define Android Virtual Devices (AVD).



An Android Virtual Device (AVD) is an emulator that simulates the configuration and behaviour of an Android device. It allows developers to test their applications on a variety of Android device configurations without requiring physical devices.

Purpose:

- **Testing and Debugging:** AVDs enable developers to test their apps on different Android versions, screen sizes, resolutions, and hardware configurations, ensuring compatibility across a diverse range of devices.
- **Emulating Device Features:** Developers can use AVDs to simulate various device features such as cameras, sensors, and network conditions during testing.
- **Efficient Development:** AVDs contribute to a more efficient development process by providing a quick and cost-effective way to identify and address issues without the need for multiple physical devices.

d. Explain Android emulator.

Android emulator is responsible for running, debugging and testing the android application.

- An android sdk comes with a virtual mobile device emulator that runs on the computer.
- The Android emulator mimics all the hardware and software features of a typical mobile devices except that it cannot place annual phone calls
- It provides selection of navigation and control keys which we can press using out mouse keyboard to generate events for our applications.
- We can select the version of android system we want to run in our emulator.

e. Explain Dalvik Virtual Machine (DVM) in detail.

The DVM (Dalvik Virtual Machine) is the virtual machine that was used by the Android operating system prior to Android 5.0 (Lollipop). It was specifically designed to run applications written in the Java programming language on Android devices.



Here are some key details about the DVM:

- **Purpose:** The DVM was developed as a part of the Android platform to execute Android applications efficiently. It is optimized for mobile devices with limited resources, such as memory and processing power.
- **Bytecode Format:** The DVM executes bytecode, which is the compiled form of Java source code. However, the DVM uses a specialized bytecode format called Dalvik Executable (DEX) format. The DEX format is optimized for smaller file sizes and efficient memory usage.
- **Register-based Architecture:** The DVM is a register-based virtual machine, meaning it uses a set of registers to store and manipulate data during program execution. It differs from stack-based virtual machines like the Java Virtual Machine (JVM), which operate on a stack structure.
- **Memory Management:** The DVM employs automatic memory management through garbage collection. It automatically frees memory that is no longer needed by the application, allowing developers to focus on writing code without explicitly managing memory allocation and deallocation.
- **Just-In-Time (JIT) Compilation:** The DVM initially used a technique called "Just-In-Time" (JIT) compilation, where certain parts of the application's bytecode are dynamically compiled into machine code at runtime for improved performance. This allows frequently executed code to be optimized for the specific device architecture.
- **Application Isolation:** Each Android application runs in its own instance of the DVM, providing a level of isolation and security between different applications. This means that applications cannot directly access each other's memory or resources.
- **Optimization Techniques:** The DVM employs various optimization techniques to improve performance, including instruction packing, constant pool reduction, and efficient garbage collection algorithms.

f. Differentiate between DVM and JVM.

Dalvik Virtual Machine (DVM)	Java Virtual Machine (JVM)
It is Register based which is designed to run on low memory.	It is Stack based.
DVM uses its own byte code and runs ".Dex" file. From Android 2.2 SDK Dalvik has got a Just in Time compiler.	JVM uses java byte code and runs ".class" file having JIT (Just In Time).
DVM has been designed so that a device can run multiple instances of the VM efficiently. Applications are given their own instance.	Single instance of JVM is shared with multiple applications.
DVM supports Android operating system only.	JVM supports multiple operating systems.
For DVM very few Re-tools are available.	For JVM many Re-tools are available.
There is constant pool for every application.	It has constant pool for every class.



Here the executable is APK.	Here the executable is JAR.
-----------------------------	-----------------------------

g. Explain the steps to install and configure Android studio .

Step 1: Go to <https://developer.android.com/android-studio/download> to get the Android Studio executable or zip file.

Step 2:

- Click on the Download Android Studio Button.
- Click on the “I have read and agree with the above terms and conditions” checkbox followed by the download button
- Click on the Save file button in the appeared prompt box and the file will start downloading.

Step 3: After the downloading has finished, open the file from downloads and will prompt the following dialog box. Click on next. In the next prompt, it'll ask for a path for installation. Choose a path and hit next.

Step 4: It will start the installation, and once it is completed, it will be like the image shown below.

Step 5: Once “Finish” is clicked, it will ask whether the previous settings need to be imported [if the android studio had been installed earlier], or not. It is better to choose the ‘Don’t import Settings option’. Click the OK button.

Step 6: This will start the Android Studio. Meanwhile, it will be finding the available SDK components.

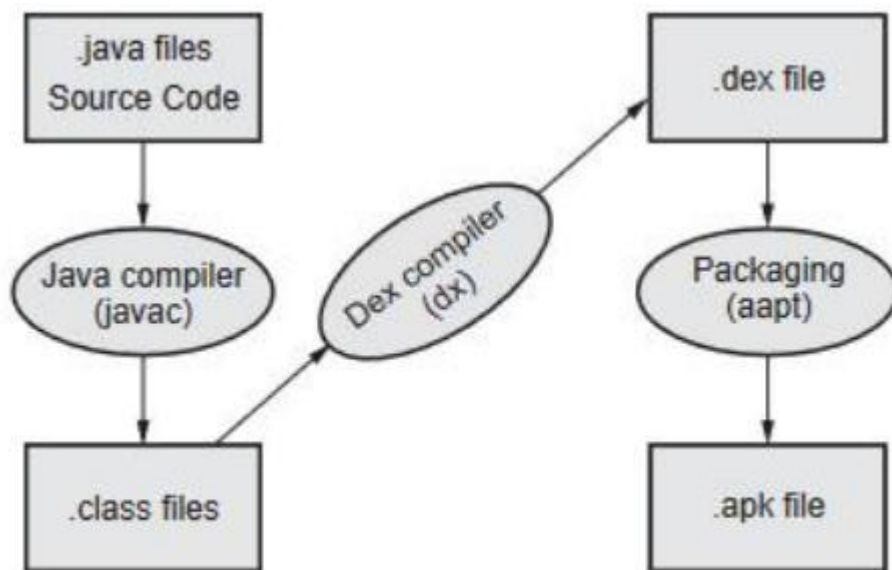
Step 7: After it has found the SDK components, it will redirect to the Welcome dialog box. Choose Standard and click on Next. Now choose the theme, whether the Light theme or the Dark one. The light one is called the IntelliJ theme whereas the dark theme is called Darcula. Choose as required. Click on the Next button.

Step 8: Now it is time to download the SDK components. Click on Finish. Components begin to download let it complete. The Android Studio has been successfully configured. Now it's time to launch and build apps. Click on the Finish button to launch it.

Step 9: Click on Start a new Android Studio project to build a new app.

i. State how the APK files are built?

- The javac tool compiles the java source file into the class file.
- The dx tool takes all the class files of your application and generates a single .dex file. It is a platform-specific tool.
- The Android Assets Packaging Tool (aapt) handles the packaging process and finally creates an executable file with extension .apk.
- An APK file contains all of a program's code (such as .dex files), resources, assets, certificates, and manifest file.



Compiling and packaging process

j. Write the directory path where images are stored while developing Android application.

app/src/main/res/drawable