APPLICATION DEVELOPMENT AND EMERGING TECHNOLOGIES

ANDROID STUDIO FUNDAMENTALS

PREPARED BY:

SUSAN S. CALUYA, MSC

Course Outcomes

At the end of the lesson students should be able to:

- Know about Android Studio
- List features of Android Studio
- Utilize and Integrated Development Environment (IDE) for efficient coding and debugging

Topics

- Android Studio (A.S.)
- How to Use A. S.
- Components in A.S.

Prerequisite

- Java Programming Language
- Object-oriented programming
- XML - properties / attributes
- Using an IDE for development and debugging

The Definition

What is Android Studio?

Android Studio is the official Integrated Development Environment (IDE) for Android App development. It is a powerful tool that allows developers to build high-quality applications for the Android platform. It has complete tools for the process of Android App development. From writing code to testing and deployment, Android studio has all the functionalities for developers to develop an Android App.

What is Android Studio?

- Android integrated development environment (IDE)
- Project and Activity templates
- Layout editor
- Testing tools
- Gradle-based build
- Log console and debugger
- Emulators

How to Use Android Studio

How to Use Android Studio

Step 1: Download Android Studio

To set up Android Studio, you need to first download the IDE from the official Android Studio download page. Choose the version that is compatible with your operating system and download the installer. Android Studio is available for Windows, macOS, and Linux. Once the download is complete, run the installer and follow the instructions to install Android Studio on your computer.

How to Use Android Studio

Step 2: Install the Required Components

During the installation process, Android Studio will prompt you to install the required components. These include the Android SDK, Android Virtual Device (AVD) Manager, and the Android Emulator. The Android SDK is a collection of libraries and tools that developers use to build Android applications. The AVD Manager is used to create and manage virtual devices for testing applications. The Android Emulator is a virtual device that allows developers to test their applications without having to use a physical device.

How to Use Android Studio

Step 3: Configure Android Studio

After installing Android Studio, you need to configure it before you can start using it. When you launch Android Studio for the first time, you will be prompted to configure the IDE. Choose the "Standard" configuration and click on "Next". In the next screen, you can choose the theme of the IDE and click on "Next" again. You can also customize the settings based on your preferences.

How to Use Android Studio

Step 4: Create a New Project

Once Android Studio is configured, you can start creating your first Android application. To create a new project, click on "Start a new Android Studio project" on the welcome screen, or select "New Project" from the File" menu. You will be prompted to choose the project name, package name, and other project details. You can also choose the minimum SDK version, which determines the minimum version of Android that the application can run on.

How to Use Android Studio

Step 5: Built Your Application

Once your project is created, you can start building your application using the various tools and features provided by Android Studio. You can use the visual layout editor to design the user interface, write code in Java or Kotlin, and use the Android SDK to access device features such as the camera, sensors, and GPS. You can also use the built-in debugging tools to troubleshoot issues and optimize your application.

How to Use Android Studio

Step 6: Test Your Application

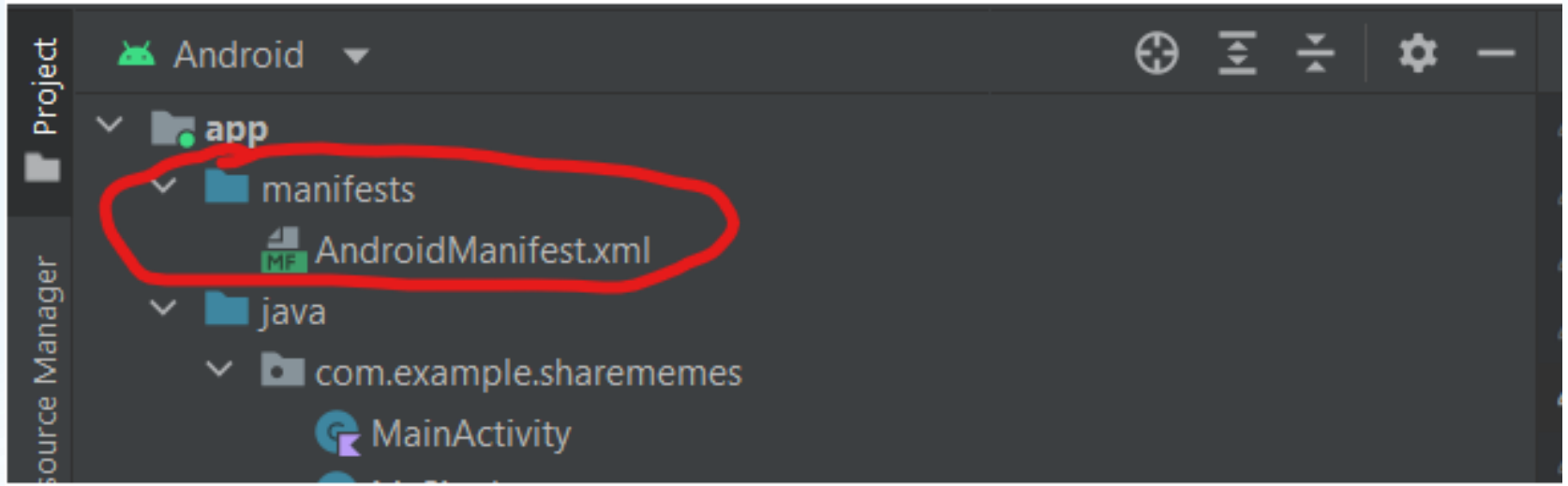
Testing your application is an important step in the development process. Android Studio comes with an emulator that allows you to test your application on different virtual devices. You can also connect your Android device to your computer and test your application directly on the device. Use the “Run” button in Android Studio to launch your application and test it on the emulator or device. You can also use the built-in profiler to analyze the performance of your application and identify any bottlenecks or performance issues.

Components in Android Studio

Components in Android Studio

1. Manifest File

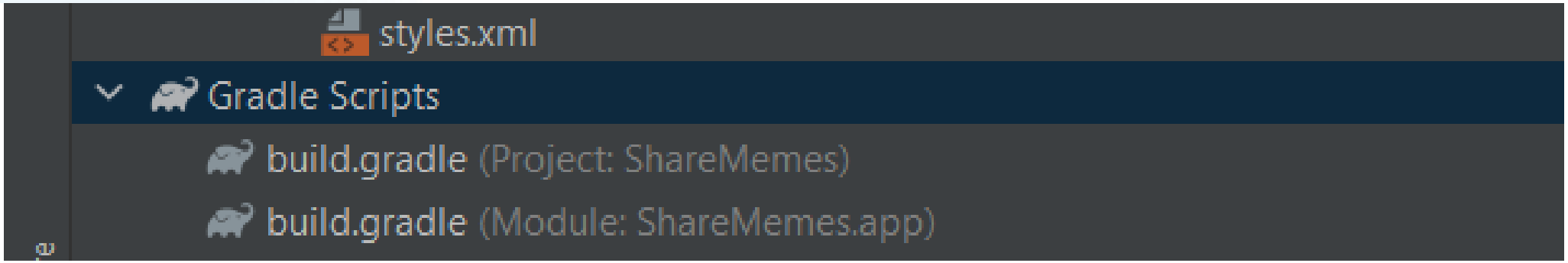
The manifest file is required for the Android system to launch the application and to determine its functionality.



Components in Android Studio

2. Build.gradle

build.gradle is a configuration file used in Android Studio to define the build settings for an Android project. It is written in the Groovy programming language and is used to configure the build process for the project.



Components in Android Studio

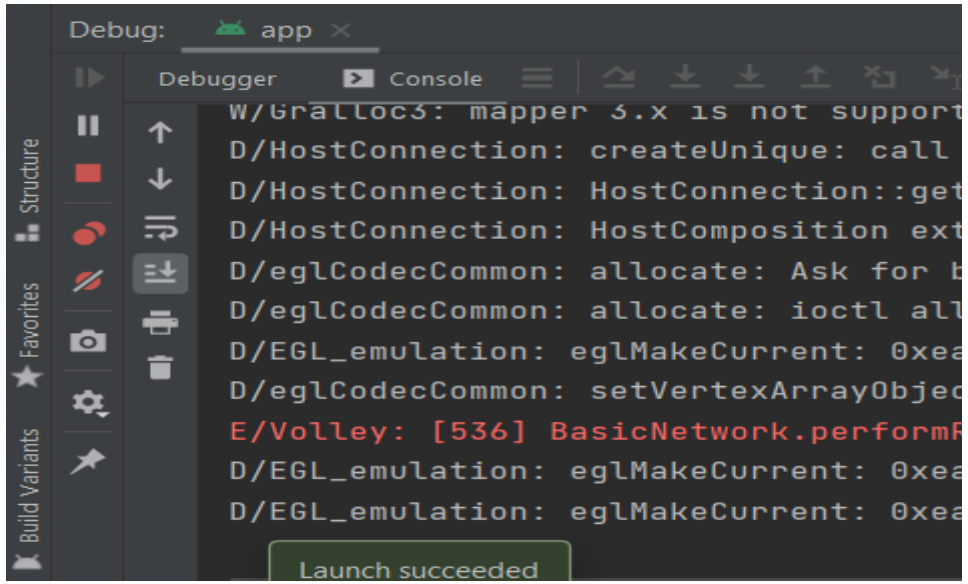
3. Git

Git is a popular version control system that allows developers to track changes to their code and collaborate with other team members. Android Studio includes built-in support for Git, making it easy to manage code changes and collaborate with others on a project.

Components in Android Studio

4. Debug

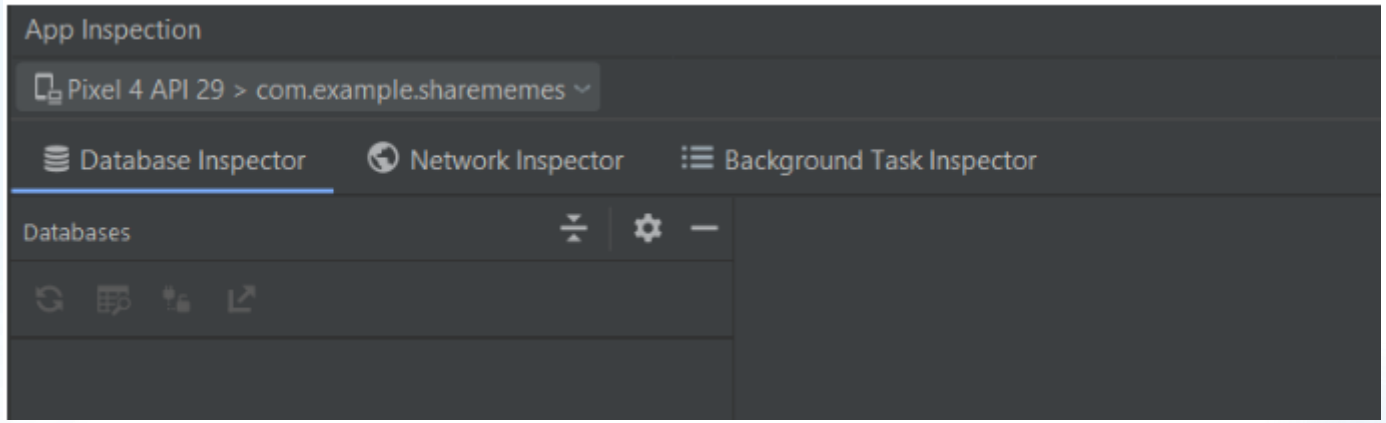
Android Studio provides a robust set of debugging tools to help developers identify and fix issues in their applications.



Components in Android Studio

5. App Inspection

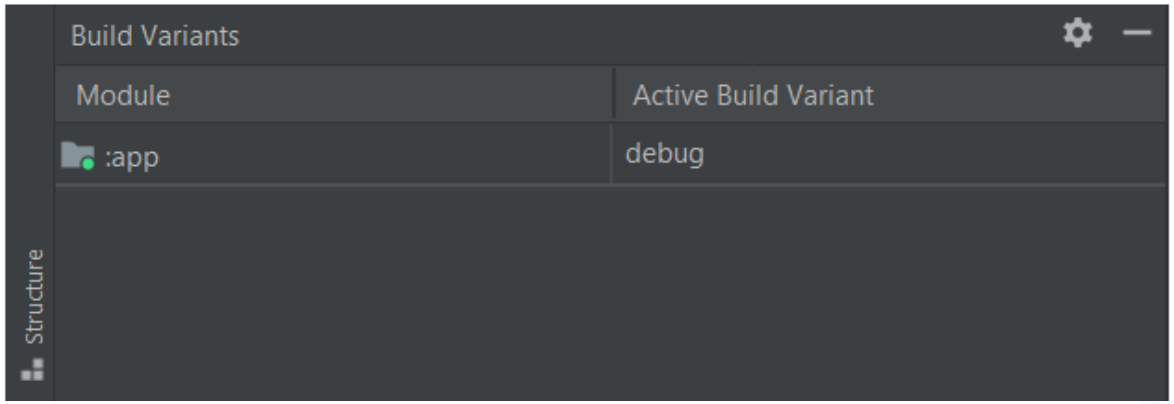
App Inspection is a feature in Android Studio that allows developers to inspect and debug their Android applications. It provides a suite of tools for analyzing the performance of the application, identifying and fixing errors, and optimizing the code.



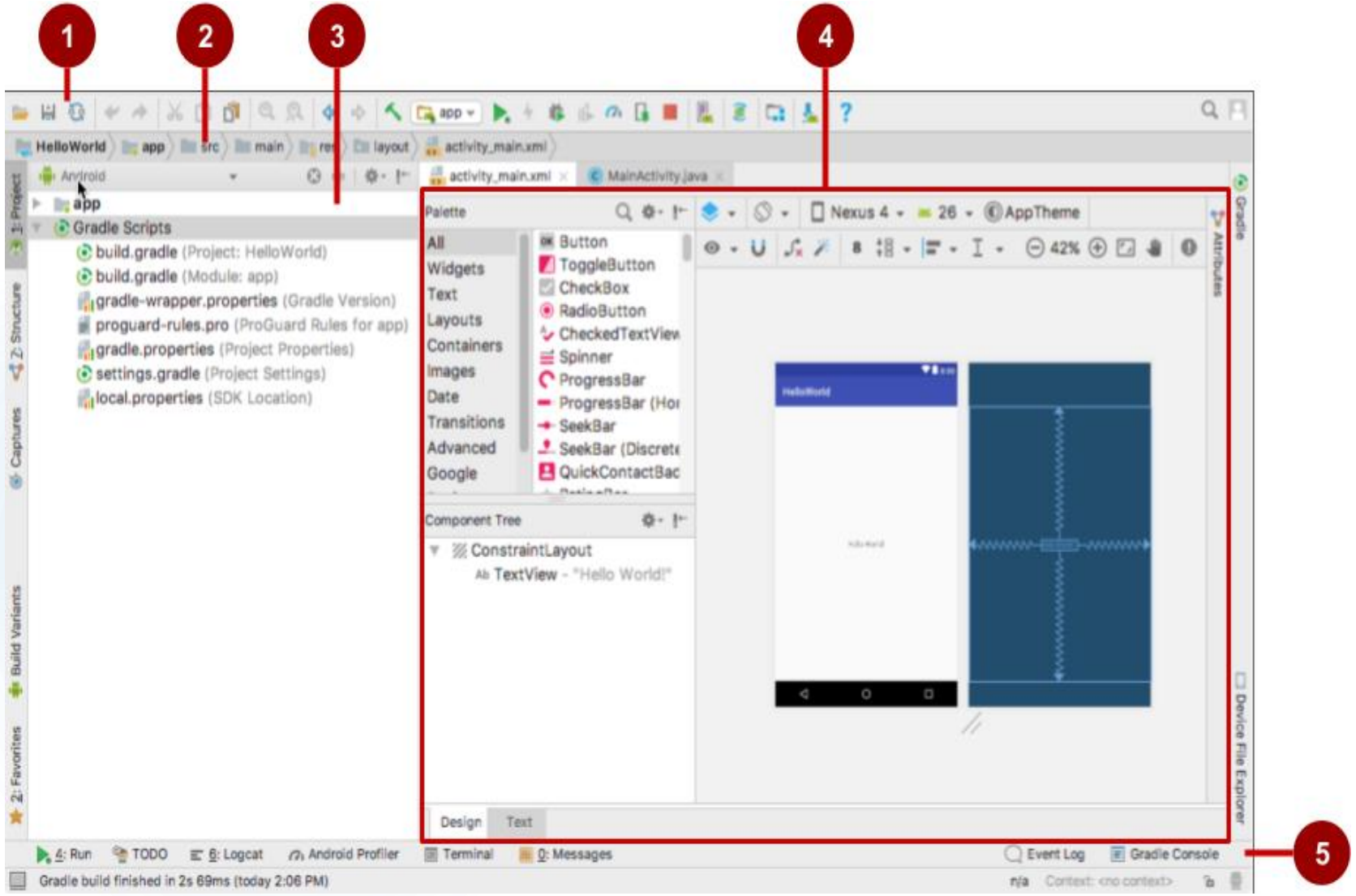
Components in Android Studio

6. Built Variance

Build variants in Android Studio are different versions of an Android app that can be built from the same source code. They are typically used to create multiple versions of an app that target different device configurations or use cases. Build variants are configured in the build.gradle file and can be built and installed separately from each other.



Android Studio Interface



1. Toolbar
2. Navigation bar
3. Project pane
4. Editor
5. Tabs for other panes

Thank You!