**✅ Section 2: Introduction to Android Studio**

**🔷 Key Concepts Taught**

1. **What is Android Studio?**
   * **Official IDE for Android development**
   * Built on **IntelliJ IDEA**
   * Supports **designing, coding, testing, debugging** Android apps
2. **How to Install Android Studio**
   * Download from: [developer.android.com/studio](https://developer.android.com/studio)
   * Installer available for:
     + Windows (64-bit)
     + macOS
     + Linux
     + ChromeOS
   * Latest version mentioned: **Ladybug**
3. **Core Features of Android Studio**
   * **Layout Editor** – drag-and-drop UI design
   * **Code Editor** – smart code completion, suggestions, linting
   * **Emulator** – test apps without a physical device
   * **Device Manager** – manage virtual and physical devices
   * **Gradle** – build system
   * **Git Integration**
   * **Jetpack Compose**, **AI tools like Gemini**
4. **Installing SDK & Tools**
   * **SDK (Software Development Kit)**: core development kit from Google
   * Components:
     + Platform Tools: command-line tools (adb, fastboot)
     + SDK Build Tools: compilers, APK packaging
     + System Images: needed for emulators
     + Libraries/APIs (camera, sensors, etc.)

**🧪 Steps to Implement Each Concept**

**🔹 Installing Android Studio**

1. Visit official site → Download latest version (Ladybug)
2. Run installer → Next → Select **Android Virtual Device** → Next → Install
3. Android SDK missing? → Installer will prompt to download:
   * Choose **API Level 35** (recommended)
   * Accept licenses → Finish

**🔹 Setting Up Android Emulator (AVD)**

1. Open Android Studio → **More Actions → Virtual Device Manager**
2. Click **+ Create Device**
3. Select device (e.g., Pixel 6 Pro)
4. Download system image (API 35) → Next
5. Configure device (name, orientation) → Finish
6. Run the emulator using the **green play button**

**🔹 Starting a New Project**

1. Open Android Studio → **New Project**
2. Select **Empty Views Activity**
   * Jetpack Compose: use Empty Compose Activity (not in this section)
3. Configure:
   * **App name:** Hello World
   * **Package name:** com.mastercoding.helloworld
   * **Save location:** Choose folder
   * **Language:** Java or Kotlin (Google recommends Kotlin)
   * **Minimum SDK:** Choose e.g. API 24
     + Lower SDK → More devices supported, but fewer modern features
     + Higher SDK → Modern features, but less device coverage
   * **Build Language:** Kotlin DSL (recommended) or Groovy DSL
4. Android Studio builds the project
5. Click green triangle ▶️ to run the app on emulator
6. App launches → Emulator shows **Hello World**

**⚙️ Tools, Libraries, APIs Used**

| **Tool/API** | **Description** |
| --- | --- |
| **Android Studio** | Official IDE |
| **AVD (Android Virtual Device)** | Simulates real Android phones, tablets, wearables |
| **Emulator** | Runs Android apps without physical device |
| **SDK Manager** | Manages SDK versions and system images |
| **Gradle** | Project build system |
| **Kotlin DSL** | Modern build configuration language |
| **System Images** | OS images for emulator testing (e.g. API 35) |

**✅ Best Practices & Instructor Advice**

* 💡 **Use emulator** for efficient, quick testing.
* 🔄 Create **multiple emulators** for different screen sizes and Android versions.
* 🎨 Use **Dark Theme (Dracula)** in Android Studio for better readability.
* 🧩 Add useful plugins (e.g., Flutter, Genymotion, Markdown support).
* 📄 Use **Kotlin DSL** for clean, type-safe Gradle scripting.
* 📁 Keep **project folder organized** – name, package, and path clearly.
* 🌐 Prefer **API Level 35** for full feature access during development.
* 📱 Always test on **both emulator and real devices** if possible.

**📘 Part B: Related Concepts Not Covered in This Section**

**🔹 Android Studio UI (Deep Dive Preview)**

Although promised for later, here's what to expect in UI overview:

* **Project View:** Project structure (manifest, java, res, gradle files)
* **Logcat:** Debug logs for app behavior and crash tracing
* **Build Console:** See real-time build progress and errors
* **Toolbar:** Run, Debug, AVD Manager, SDK Manager
* **Palette & Component Tree (for Views):** UI element management

**🔹 Physical Device Setup (if skipping emulator)**

* Enable **Developer Options** on phone → Enable **USB Debugging**
* Connect via USB → Trust computer → App can run directly on the device

**🔹 Recommended Plugins to Install Early**

* Kotlin
* Flutter
* Firebase
* Material Design Icons
* Markdown Support
* JSON Viewer

**🔹 Build Configuration File: build.gradle.kts (Kotlin DSL Example)**

plugins {

id("com.android.application")

kotlin("android")

}

android {

compileSdk = 35

defaultConfig {

applicationId = "com.mastercoding.helloworld"

minSdk = 24

targetSdk = 35

versionCode = 1

versionName = "1.0"

}

buildTypes {

release {

isMinifyEnabled = false

}

}

}

**🔹 Emulator Limitations**

* Heavier on system RAM/CPU
* Some features (e.g., Bluetooth, fingerprint scanner) may not be fully testable
* **Use physical device** for testing battery performance or hardware sensors

**📎 Summary Table**

| **Topic** | **Summary** |
| --- | --- |
| Android Studio Setup | Install from official site, latest version (Ladybug) |
| Emulator Creation | AVD Manager → API 35 → Configure device → Run |
| SDK & Tools | Platform tools, build tools, system images, libraries |
| New Project Setup | Empty Views Activity, Kotlin preferred, Kotlin DSL |
| Best IDE Practices | Dark theme, plugin use, shortcut familiarization |
| Emulator Features | Device simulation, APK install, debugging, screen record |
| Build System | Gradle with Kotlin DSL preferred |