

# Android Development Course Content

## 1. Introduction to Android

- What is Android?
- History and evolution of Android
- Overview of Android architecture
- Android development tools and setup (Android Studio, SDK, emulator)

## 2. Java/Kotlin Basics

- Introduction to Java/Kotlin (choose one as the primary language)
- Basic syntax, variables, and data types
- Control structures (if, else, switch, loops)
- Functions and methods
- Object-oriented programming concepts (classes, objects, inheritance, polymorphism)

## 3. Android Studio Overview

- Navigating Android Studio
- Project structure and files
- Using the Android Emulator
- Debugging tools

## 4. User Interface (UI) Development

- Overview of Android UI components (Views, ViewGroups)
- Layouts (LinearLayout, RelativeLayout, ConstraintLayout)
- Working with UI elements (TextView, EditText, Button, ImageView)
- Styling and themes
- Creating responsive layouts for different screen sizes

## 5. Activities and Intents

- Understanding Activities and their lifecycle
- Creating multiple Activities
- Intents (explicit and implicit)
- Passing data between Activities

## 6. Fragments

- Introduction to Fragments
- Fragment lifecycle
- Adding and replacing Fragments in Activities

- Communication between Fragments and Activities

## **7. Data Storage**

- Shared Preferences for simple data
- SQLite database basics
- Room Persistence Library for database management
- Storing files in internal and external storage

## **8. Networking**

- Overview of networking in Android
- Using REST APIs with Retrofit or Volley
- Parsing JSON data
- Displaying data in the app (RecyclerView)

## **9. Working with Media**

- Handling images and video
- Using the camera and accessing the gallery
- Playing audio and video files

## **10. Background Tasks**

- Introduction to background processing
- Using AsyncTask (deprecated in newer versions)
- Introduction to WorkManager for scheduling background tasks

## **11. Notifications**

- Creating and managing notifications
- Using Notification Channels (Android Oreo and above)

## **12. User Location and Maps**

- Using Google Maps API
- Getting user location
- Displaying maps and adding markers

## **13. Best Practices**

- Application architecture (MVC, MVVM)
- Performance optimization techniques
- Security best practices
- Version control with Git