

Enterprise AOSP Platform Development

Duration: 6 Days (48 Hours)

Day 1: Android Platform Architecture, Linux & AOSP Foundations

Module 1: Android Architecture & AOSP Fundamentals

- Android Operating System Overview and Evolution
- Android Platform Architecture
- Hardware, HAL, Native Libraries, Android Runtime and Framework
- Application Framework and System Services
- Android Runtime (ART) Fundamentals
- Android Process and Application Lifecycle
- Android Boot Flow: Bootloader, Kernel and User Space
- AOSP Overview and Ecosystem
- AOSP vs Android Platform and Google Mobile Services
- AOSP Directory Structure: frameworks, packages, system, hardware, device and vendor

Hands-on Lab

- Explore the AOSP source tree and identify major components
- Map Android architecture on a running emulator
- Trace a sample application request through framework components

Module 2: Linux Kernel & Android Platform Interfaces

- Linux Kernel Fundamentals
- Android Kernel Architecture
- Kernel Configuration Concepts
- /proc and /sys Interfaces
- Processes, Threads and Signals
- Linux Users, Groups and Permissions
- Kernel Logs and dmesg
- Native and Cross-Compilation Concepts
- Kernel and User-Space Interaction

Hands-on Lab

- Inspect /proc, /sys and kernel information using adb shell
- Analyze processes, threads and kernel logs
- Explore Android kernel configuration and build outputs

Module 3: AOSP Environment, Repo & First Build

- AOSP Development Environment
- Linux Build Dependencies
- JDK and Required Tools
- Repo Tool
- AOSP Manifest and Repository Synchronization
- Build Targets
- Build Variants: user, userdebug and eng
- Soong Build System Introduction
- Incremental vs Clean Builds
- Build Artifacts and Output Directories

Hands-on Lab

- Prepare the RDP Linux VM for AOSP development
- Initialize and synchronize a manageable AOSP source tree
- Configure a build target and perform the first emulator build
- Troubleshoot common AOSP build issues

Day 2: AOSP Build System, Device Configuration & System Customization

Module 1: Soong, Android.bp & Build Customization

- Soong Build System Architecture
- Android.bp Syntax
- Module Types and Properties
- Module Dependencies
- Visibility and Namespace
- Android.mk Compatibility
- Product Packages
- Build Variants and Conditional Configuration
- Build Graph
- Build Dependency Troubleshooting

Hands-on Lab

- Create a custom Android.bp module
- Add Java or native code to the AOSP build
- Introduce and fix a build dependency error
- Verify the generated artifact

Module 2: Product, Device & Partition Configuration

- Product Configuration
- Board Configuration Concepts
- Device Configuration Structure
- Device Tree Concepts
- system, vendor, product and userdata Partitions
- Build Properties
- Default Applications and Packages
- Vendor Components
- Device-Specific Configuration
- Image Generation

Hands-on Lab

- Modify product configuration
- Add a default package to the build
- Change selected device/build properties
- Generate and inspect customized system images

Module 3: Android Emulator, ADB & System Customization

- Android Emulator Architecture
- ADB Architecture
- adb shell
- push, pull and install
- logcat and Log Filtering
- getprop and dumpsys
- Package, Activity and Process Inspection

- System Applications vs User Applications
- Framework Resources
- Build ID, Default Settings and Boot Animation

Hands-on Lab

- Boot a locally built AOSP image
- Install and debug an APK using adb
- Modify a visible system setting and build identification
- Customize selected framework resources
- Rebuild and validate the changes in the emulator
- Collect logs and before/after validation evidence

Day 3: Android Framework, Binder/AIDL & Native Platform Development

Module 1: Android Framework & System Services

- Android Framework Internals
- Framework APIs
- System Server
- System Service Architecture and Lifecycle
- Service Registration
- Framework Resources
- Permission Checks
- Application-to-Framework Communication
- Framework Logging
- Debugging Java Framework Components

Hands-on Lab

- Trace an API call into a framework/system service
- Modify a small framework-level behavior
- Build and validate the framework change
- Debug the change using logcat and service inspection

Module 2: Binder IPC & AIDL

- Binder IPC Architecture
- Binder Transactions
- AIDL Concepts
- Defining AIDL Interfaces
- Client-Server Communication
- Binder Threads
- Service Registration
- Binder Permissions
- IPC Failure Handling
- Binder Diagnostics and Shared Memory Concepts

Hands-on Lab

- Create a simple AIDL service and client
- Build and run the service on the AOSP emulator
- Add permission validation
- Inspect Binder-related logs and analyze transaction behavior

Module 3: Native AOSP Development & Debugging

- Android Native Layer

- Platform Native Development
- Native Libraries and Executables
- JNI Concepts
- Shared and Static Libraries
- Android.bp for Native Components
- Native Logging
- Native Runtime Errors
- Tombstones and Native Crash Reports
- Symbols and Native Stack Resolution

Hands-on Lab

- Create a C/C++ native library
- Integrate it into an AOSP component
- Call native functionality from Java
- Trigger a controlled native crash
- Analyze logcat and tombstone output
- Resolve the native stack using available symbols

Day 4: HAL, SELinux, Services & Physical Device Integration

Module 1: HAL, AIDL HAL & Hardware Abstraction

- HAL Responsibilities and Boundaries
- Framework-to-HAL Data Flow
- HIDL Concepts
- AIDL-Based HAL Concepts
- Vendor Interface
- Vendor Partition
- Hardware-Dependent vs Emulator-Friendly Development
- HAL Service Registration
- Mock and Simulated Hardware Interfaces
- HAL Logging and Debugging

Hands-on Lab

- Trace an existing framework-to-HAL interaction
- Create a simulated hardware-facing component
- Expose controlled data through a mock/simulated HAL
- Validate framework-to-HAL communication using logs

Module 2: SELinux, Android Init & System Services

- SELinux in Android
- Domains and Types
- Enforcing and Permissive Concepts
- sepolicy Structure
- File Contexts
- Domain Transitions
- AVC Denials
- Android init
- init.rc and Service Definitions
- Service Startup and Restart Behavior
- Native Daemons and System Services

Hands-on Lab

- Generate and analyze a controlled SELinux denial
- Create a minimal policy change for a lab component
- Inspect init configuration and startup services
- Trace a service from boot to runtime
- Analyze controlled service failure and restart behavior

Module 3: Physical Android / Set-Top Box Integration

- ADB Connection to Physical Android Target
- USB Debugging and Authorization
- Build and Hardware Information
- adb shell on Physical Target
- /system, /vendor and /data Inspection
- Package Installation and Testing
- System Properties
- Boot and Runtime Log Collection
- Network Diagnostics
- Safe Device Validation and Recovery Workflow

Hands-on Lab

- Connect the available set-top box where supported
- Verify adb connectivity and collect device/build information
- Install and test a sample application
- Capture boot and runtime logs
- Inspect system/vendor/data partitions
- Compare physical-target behavior with the AOSP emulator
- Perform safe troubleshooting without assuming bootloader unlock or flashing access

Day 5: Advanced Debugging, Performance & Platform Validation

Module 1: Advanced AOSP Debugging & Troubleshooting

- Platform Debugging Methodology
- Application and Framework Failures
- Native Crash Investigation
- Boot Failure Analysis
- System Service Failures
- SELinux Denial Troubleshooting
- ADB-Based Remote Diagnostics
- Log Collection and Filtering
- Correlation of Logs Across Components
- Root Cause Analysis and Fix Validation

Hands-on Lab

- Diagnose a controlled platform issue
- Collect and correlate logs from multiple Android components
- Identify root cause and apply a corrective change
- Validate the fix through repeat testing

Module 2: Android Performance, Tracing & Optimization

- Android Performance Fundamentals
- CPU and Memory Bottlenecks
- Process and Thread Analysis

- I/O and Network Bottlenecks
- Memory Usage and Leak Investigation
- adb shell dumphsys
- System Tracing Concepts
- Perfetto Trace Analysis
- perf-Based Analysis
- Boot-Time Performance
- Performance Baselines and Regression Analysis

Hands-on Lab

- Collect CPU, memory and process statistics
- Use dumphsys for system/service analysis
- Capture and analyze a system trace
- Identify a platform bottleneck
- Apply an optimization and compare before/after results

Module 3: CTS, VTS & Compatibility Validation

- Android Compatibility Concepts
- CTS Overview and Test Structure
- VTS Overview
- Test Environment Preparation
- Running Suitable Compatibility Tests
- Reading Test Reports
- Failure Classification
- Source-Level Failure Investigation
- Regression Validation
- Platform Release Readiness

Hands-on Lab

- Explore CTS/VTS test structure
- Run suitable compatibility checks in the available environment
- Analyze a failed test result
- Trace the failure to the affected platform component
- Validate the corrective change and document the result

Day 6: Advanced Platform Engineering, Boot Analysis & Device Validation

Module 1: Advanced Framework, System Services & Platform Integration

- Deep Dive into System Server
- Advanced System Service Architecture
- Service Dependencies and Lifecycle
- Framework API to System Service Flow
- Permission Enforcement in System Services
- Framework Resource Integration
- System Service Configuration
- Service-to-Native Communication
- Framework Logging and Diagnostics
- End-to-End Platform Data Flow

Hands-on Lab

- Trace a complete request from application API to system service
- Modify a controlled system-service configuration/behavior

- Integrate the change with a native/platform component
- Build and validate the complete flow using adb and logs

Module 2: Advanced Boot, Init & Low-Level Platform Analysis

- Android Boot Sequence in Detail
- Bootloader to Kernel Handoff
- Kernel to Android Init
- init.rc and Import Mechanism
- Triggers and Properties
- Early-Boot Services
- Service Startup Order
- Boot-Time Log Analysis
- Boot Loop Diagnosis
- Startup Failure Recovery
- Kernel/User-Space Boundary Analysis

Hands-on Lab

- Trace the complete boot sequence using logs
- Inspect init configuration and service triggers
- Create a controlled startup configuration change
- Diagnose a simulated service/startup failure
- Recover the target using safe emulator/device procedures

Module 3: Advanced Device Validation & Real-World Troubleshooting

- Emulator vs Physical Target Validation
- Device Build and Hardware Information
- System and Vendor Partition Validation
- ADB-Based Device Diagnostics
- Application and System Package Validation
- Network and Connectivity Validation
- Service and Permission Validation
- Stability and Regression Checks
- Build Artifact Verification
- Deployment and Recovery Considerations
- Technical Troubleshooting Documentation

Hands-on Lab

- Run a structured validation checklist on the emulator
- Repeat selected checks on the available physical set-top box where supported
- Collect build, system, service and network diagnostics
- Validate platform components and configuration
- Perform a controlled regression check
- Prepare a deployment-readiness and troubleshooting report