

# Android Debugging

## Course 204 – 24 Hours

### Overview

Android Debugging is an advanced course that targets competent Android programmers (application and system) who wish to learn more on common failure modes of Android applications and services, and how to debug them with various tools and programming methods. The course covers application debugging with and without native code as well as debugging techniques for system (AOSP) code.

### Course Objectives

Understand common Android application and system failure modes and efficiently debug them using advanced methods.

### Who Should Attend

Competent Android programmers who wish to achieve a higher degree of understanding of common Android component failure modes and advanced debugging techniques used to handle them.

### Prerequisites

Participants must have completed the Android Programming course or have equivalent knowledge.

### Course Contents

#### **Android Application Debugging**

- Overview
- Debugging java code
- Debugging java and native code
- Common pitfalls
- Using ADB
- Using Logs and Logcat
- Lint
- Debug class
- Memory analyzer tool
- Using DDMS
- Simulating events
- Hierarchy viewer
- Monkeyrunner
- Unit test framework

### **Android System Debugging**

- Debugging information
- ELF tools
- Symbols, name mangling and map files, problems regarding symbols
- Include paths, Lib paths
- Dynamic loading
- strace
- ltrace
- GDB overview
- Useful commands
- Remote debugging
- Debugging native shared libraries
- Automation
- Preparing your code for efficient debug and trace
- Proper exception trapping and handling
- Core dumps, crash data and tombstones

### **Android System Tools**

- atrace
- perf
- dumpstate and dumpsys
- System properties – setprop/getprop/watchprop
- schedtop/schedtest
- librank
- procrank/procmem
- showmap
- timeinfo
- input, ioctl
- Controlling init
- notify
- dalvikvm
- dexdump
- service
- am
- pm
- svc
- monkey

### **Kernel Debugging**

- Kernel configuration for debug
- Debug with printk
- Debug with proc and sysfs
- debugfs
- kgdb
- Ftrace
- Tracepoints