SYNTAX
שפה מייצרת חשיבה

Android Internals

We are pleased to provide you with a comprehensive overview of our upcoming 45-hour Android Internals course, which will be expertly conducted by Mr. Eran Katsav, the CEO and founder of Syntax College.

About Syntax College (and its CEO, Eran Katsav)

Syntax College is renowned for delivering a range of courses focused on the Android operating system, catering to private corporations, government organizations, and offering online modules to the general public. Noteworthy clients include organizations such as General Motors™ and Carlson Wagonlit Travel™, who have graciously provided letters of recommendation, which are enclosed for your reference.

It is imperative to highlight that all our courses are personally conducted by Mr. Eran Katsav himself, a distinguished professional holding a B.Sc. degree in Computer Science from the Hebrew University in Jerusalem and a master's degree with honors from Tel Aviv University.

Eran's teaching career spans over 22 years. He currently holds positions at Reichman University (IDC, International School) and the Holon Institute of Technology (HIT). At these institutions, he serves as a lecturer in the Department of Computer Science, specializing in Native Android courses covering Kotlin and Java. More than that, in HIT Eran serves as the head and founder of the mobile lab within the Department of Digital Medical Technologies.
Concurrently, Eran has established himself as an independent developer in the mobile field over the past decade. Among his notable accomplishments is the creation of the first Hebrew voice assistant app, "Bip it," which has been adapted into various white-label versions, including integration into all Suzuki™ vehicles distributed in Israel.

SYNTAX
שפה מייצרת חשיבה

Course Overview

Our Android Internals course offers a comprehensive exploration of every facet of the Android Operating System. It commences with an in-depth examination of its historical evolution and extends to a profound analysis of its core architecture. The curriculum encompasses a thorough study of critical components, including the Android Open Source Project, Google Mobile Services, and the intricate layers of the platform, such as the Kernel, HAL, ART, NDK, and Java/Kotlin API Framework. Students will develop a profound understanding of the Android boot process, system services, communication with system services, application framework, memory management, file system, and security, which is achieved through both by Linux process isolations (Sandbox) and SELinux.

Furthermore, the course covers Android Studio for app development, command-line tools like ADB, app components, intents, permissions, and various aspects of app development, ranging from GUI fundamentals to multithreading. It provides hands-on exercises for ADB usage and delves into Android Native Development Kit (NDK), JNI, and Android in-kernel changes, with a particular focus on Binder and inter-process communication. Whether you are a newcomer to Android or seeking to enhance your expertise, this course equips you with the knowledge and skills to master the Android Operating System.

Meeting Details

The Android Internals course is structured across five sessions. Each session spans 9 hours of in-depth study and is divided into two parts. The first part will be held from 9:00 AM to 12:45 PM, followed by the second part from 1:30 PM to 4:30 PM. Meeting schedules will be thoughtfully coordinated with your organization's needs. For your convenience, we have attached a detailed syllabus and a session-by-session breakdown for your reference.

SYNTAX
שפה מייצרת חשיבה

Android Internals Syllabus

Android OS Layer

- Android historical overview
- The Android Open Source Project
- Google mobile services
- Platform Architecture (Kernel, HAL, ART, NDK, Java and Kotlin API Framework, System Apps)
- Android Core - Virtual Machines: JVM, DVM & ART
- Android OS boot process explained with focus on Zygote
- Android System Server
- Android System Services (exploring, activating and creating)
- Communication with System Services through managers
- Exploring application framework
- Processes & Threads
- Android Memory Management
- Android File System Overview
- Security: Device, Kernel and App
- Android System permissions (signature|privileged)
- Android Platforms (Wear, TV, Android for Cars)
- Android releases primary key features
- AndroidX support library and Android Jetpack

Android Studio

- Android Studio Overview
- Projects overview (App Modules, App files and directory structure)
- Build and run your app and Build Configurations
- App Gradle
- App Profiling and performance
- Analyze your build with app analyzer

Command line tools

- Android SDK tools
- **Android Debug Bridge(ADB) - In details**
- Record the phone screen and capturing screen shots
- Copying files (from device to computer and vice versa)

- ADB shell (am - Activity Manager, keyeevnt, etc)
- Capturing an image from the live camera and retrieving the photo with ADB.
- **Exercise** - Sending an SMS, Navigating and more with ADB

Android Application from inside

- App Components (Activity, Service, Broadcast Receiver & Provider)
- App states (Foreground, Background, idle)
- Intents and Intent Filter - communication between processes
- **App Manifest File**
- GUI basics and resource localizations
- App Security & Permissions
- User Location & Google play services
- Foreground & Bounded Services
- App Storage (internal & external)
- Using File Provider as a demonstration for Content Provider
- Multithreading and Responsiveness
- Summery - Security, Processes & IPC (Inter Process Communication)

Do it yourself (ADB)

- Android asset packaging tool (aapt)
- Location App
    - Granting and revoking location permissions
    - activate foreground location report
    - Activating background location tracker
- The storage app
    - App internal and external storage overview
    - Sending broadcasts to activate the camera and save the data in different locations
    - Granting external storage permission
    - Accessing app internal storage with ADB and retrieving files
- MultiThreading app
    - Downloading imagine s from web in background thread vs UI thread
    - Responsive vs Jammed UI
    - Changing android threading policy
- Querying heap status
    - Checking processes memory consumptions
    - Making changes in app States and components lifecycle states

SYNTAX
שפה מייצרת חשיבה

Android Native Development Kit (NDK) and JNI

- What is in NDK? And Why?
- What is JNI and how it works
- Shared code libraries
- Using NDK and JNI full Android System undue example
- Hands on native calculator Android app
- Using NDK Manually (without Android studio)
- NDK advantages and disadvantages
- More on JNIEnv
- NDK's Stable APIs

Android in-kernel changes with focus on Binder

- Linux Kernel Android API changes (Binder, Ashmem, Pmem/ION, Wakelock, Early Suspend, Alarm, Low Memory Killer, Logger, Real — time requirements)
- Inter Process Communication (IPC) in Android OS
- Binder as the base for IPC in android
- Security with Binder Tokens (Incl Examples from Framework)
- Binder in details (Proxy, Stub, Parcel, binder objects and transactions with native code)
- Binding to Framework service using adb
- What is AIDL?
- Building a AIDL Binder based Service and Client by Example
- ashmem (Anonymous Shared Memory) vs Linux Shared memory
- ashmem in details on (MemoryFile, FileDescripor, read/write, pin/unpin)
- ashmem and pmem
- Memory pressure and Legacy LMK driver
- Android low memory killer daemon
- Wakelocks kernel add-on and it framework PowerManager implementation
- Logger and Android logging system

SYNTAX
שפה מייצרת חשיבה

Course Schedule:

Day 1
9:00 - 16:00 - Android OS Overview - Theory

Day 2
9:00 - 12:30 - Android Studio + Android Debug Bridge - Theory
13:30 - 16:00 - Command Line Tools - Activating Phone services with adb - Practical

Day 3
9:00 - 12:30 - Android Application from inside - Theory
13:30 - 16:00 - Do it yourself (ADB) - Using ADB tools to activate in-depth App Components - Practical

Day 4
9:00 - 12:30 - Closing Gaps and user preferred bonus topics
13:30 - 16:00 - Android Native Development Kit (NDK) and JNI- Theory + Practical

Day 5
9:00 - 16:00 - Android in kernel changes with focus on Binder - Theory + Practical

Pre-Requirements

All Participants Must have Basic Linux OS knowledge (File System structure, using command line tools, basic memory management knowledge etc.)