

(/)

FREE  
RESOURCES (/S)

PAID  
TRAINING



[CATEGORIES \(/TRAINING/CATEGORIES.HTML\)](/TRAINING/CATEGORIES.HTML) |  
[PUBLIC SCHEDULE \(/TRAINING/SCHEDULE.HTML\)](/TRAINING/SCHEDULE.HTML) |

[ALL COURSES \(/TRAINING/COURSES.HTML\)](/TRAINING/COURSES.HTML) |  
[ONSITE \(/TRAINING/ONSITE.HTML\)](/TRAINING/ONSITE.HTML) |

 e.g. *Tab Navigation with JQuery, HTML5 Developers*



[Android Internals](#)

# Android Internals Training Course

## Course Summary

The Android Internals training course is designed for those who are already familiar with basics of the Android SDK and are looking to customize and/or extend the functionality of the platform.

Android Internals focuses on the Android NDK, and Android IDL APIs, to give you a clean access to underlying hardware and services, with future compatibility in mind. You will learn how to build custom images and hack the platform.



[Duration](#) | [Objectives](#) | [Audience](#) | [Prerequisites](#) | [Outline](#) | [Additional Notes](#) |  
[Testimonials](#) |

## Duration

[\[top\]](#)

5 days.

Android Internals is composed of the following:

- [Android Overview Module \(/training/android/419.html\)](/training/android/419.html) (1 hour)
- [Android Stack Module \(/training/android/440.html\)](/training/android/440.html) (1 day)
  - [Android Kernel Layer \(/training/android/421.html\)](/training/android/421.html) (3 hours)
  - [Android Native Layer \(/training/android/422.html\)](/training/android/422.html) (2 hours)
  - [Android Application Framework Layer \(/training/android/423.html\)](/training/android/423.html) (1 hour)
  - [Android Applications Layer \(/training/android/430.html\)](/training/android/430.html) (1 hour)
- [Java Native Interface \(JNI\) and the Android Native Development Kit \(NDK\) \(/training/android/431.html\)](/training/android/431.html) (4 hours)
- [Android Inter-Process Communication \(IPC\) with Binder and AIDL \(/training/android/432.html\)](/training/android/432.html) (4 hours)
- [Android Security Essentials \(/training/android/433.html\)](/training/android/433.html) (3 hours)
- [Building Android from Source \(/training/android/434.html\)](/training/android/434.html) (2 hours)
- [Android Startup \(/training/android/435.html\)](/training/android/435.html) (2 hours)
- [Android Subsystems \(/training/android/436.html\)](/training/android/436.html) (3 hours)
- [Creating a Customized Android System Image \(/training/android/437.html\)](/training/android/437.html) (1 day)
- [Android Tools and Debugging \(/training/android/438.html\)](/training/android/438.html) (2 hours)

## Objectives

[\[top\]](#)

By completing Android Internals training course you will be able to:

- Explain the anatomy of the Android platform and get is physiology (layer interactions)
- Build native applications in Android using JNI and NDK
- Take advantage of Android AIDL to build IPC-enabled bound services
- Build the entire Android platform from source and get what's what
- Customize and extend the Android platform to build custom ROMs
- Modify and extend Android frameworks and services
- Take advantage of custom hardware with Android
- Understand where Android departs from standard Linux

## Audience

[\[top\]](#)

This Android Internals course is for developers who want to dig deeper than the standard Android SDK. It is for those who want to hack the system a bit in order to add system services and hardware support for non-standard components or port Android to completely new boards.

## Prerequisites

[\[top\]](#)

### Java

To take this course, you must know Java. You should be able to answer most of the following questions:

- What is a difference between a class and an object?
- What is the difference between static and non-static field?
- What is the difference between extends and implements keywords?
- What is an anonymous inner class?
- What is the purpose of @Override?

To refresh your Java skills you can review our [NewCircle Java Tutorial](http://thenewcircle.com/bookshelf/java_fundamentals_tutorial/index.html) ([http://thenewcircle.com/bookshelf/java\\_fundamentals\\_tutorial/index.html](http://thenewcircle.com/bookshelf/java_fundamentals_tutorial/index.html)).

Additionally, knowledge of Eclipse is required. You could watch this [30-minute Eclipse tutorial](http://thenewcircle.com/s/video_introduction_to_eclipse_driving_java_productivity.595/index.html) ([http://thenewcircle.com/s/video\\_introduction\\_to\\_eclipse\\_driving\\_java\\_productivity.595/index.html](http://thenewcircle.com/s/video_introduction_to_eclipse_driving_java_productivity.595/index.html)) to get up to speed.

### C/C++/Make

To get the most benefit out of this class you must have a basic understanding of C and C++. For example, you should be able to answer the following:

- What is a header file?
- What is gcc and how to use it?
- Basic usage of sprintf()
- What is make and how does it work?
- Be able to read and understand basic Makefiles
- Be able to read and understand shell scripts

### Linux

You should be familiar with basic Linux operating system. For example, you should be able to answer most of the following:

- How do you use the following commands: ls, ps, cp, mv, pwd, cat, chmod, chown, mount, and similar.
- What is the init process?
- What are users and groups in Linux and how do r/w/x permissions work?

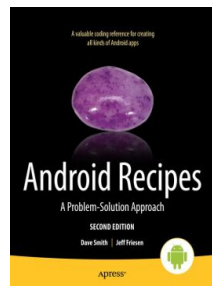
## Instructors

[\[top\]](#)

### [Dave Smith \(/expert/dave\\_smith,20690.html\)](/expert/dave_smith,20690.html)

Dave Smith is the Android Practice Lead at NewCircle, where he focuses on development and maintenance of courseware materials.

Dave has worked in developing software at all levels of the Android platform since 2009; from writing user applications using the SDK to building and customizing the Android source code for embedded devices. Prior to that, he was an embedded applications developer and hardware systems integrator for the M2M industry, working mostly with 8 and 16-bit microprocessors. His favorite mobile projects are those that integrate custom accessory hardware with consumer devices or involve building Android for embedded platforms. Today he specializes primarily in integrating custom device interfaces, such as USB and UART, with application layer services on embedded Android hardware.



Dave is also passionate about providing resources for developers that they can make use of long term. He is the author of the popular Android book [Android Recipes: A Problem Solution Approach](http://www.apress.com/9781484204764) (<http://www.apress.com/9781484204764>) published by Apress; a cookbook style text dedicated to getting Android developers up and running quickly by providing real-world useful examples of how to use the Android SDK and NDK to build applications quickly and well. Dave is a regular speaker at Android conferences, where he usually speaks on topics related to hardware integration and framework internals. He frequently shares ideas via the NewCircle Stream ([https://thenewcircle.com/s/author/20690/dave\\_smith](https://thenewcircle.com/s/author/20690/dave_smith)), his personal [development blog](http://blog.wiresareobsolete.com) (<http://blog.wiresareobsolete.com>), Twitter ([@devunwired](http://twitter.com/devunwired) (<http://twitter.com/devunwired>)), and [Google+](http://plus.google.com/+DaveSmithDev) (<http://plus.google.com/+DaveSmithDev>).

Dave received his degrees in Electrical Engineering and Computer Science from the Colorado School of Mines, and is a licensed Professional Engineer.

### [More about Dave Smith... \(/expert/dave\\_smith,20690.html\)](/expert/dave_smith,20690.html)

### [Gil Zhaiek \(/expert/gil\\_zhaiek,24638.html\)](/expert/gil_zhaiek,24638.html)

Gil Zhaiek joined the NewCircle team from Vancouver, Canada and provides his expertise in the Android framework and the Linux kernel. He has 13 years of software and hardware experience that is critical for the field of board bring-ups. He started as a C++ coder of an online financial calculator, switched to FPGA and digital board design for a couple of years, and later integrated his knowledge in designing C++ Testbenches as a consultant to various hardware teams in more than 20 companies.



Having experience in nearly all the hardware and software components, he was able to transition smoothly to Real Time OS with Embedded Linux where he wrote drivers and was able to debug the hardware independently. A few years ago, Gil and his wife moved to Vancouver where he joined Recon Instruments as the Senior Embedded Android Engineer and later the Embedded Software Manager responsible for board bring-ups and Android framework customization.

Some of Gil's notable products he worked on include the MOD Live which was the first Ski Smart Head-Mounted-Display and the Recon Jet - a light sports HMD.

Gil earned his B.Sc in Mathematics and Computer Science from the University of Arizona. He also holds an MBA from Heriot-Watt University.

[More about Gil Zhaiek... \(/expert/gil\\_zhaiek,24638.html\)](/expert/gil_zhaiek,24638.html)

[Justin Lloyd \(/expert/justin\\_lloyd,26776.html\)](/expert/justin_lloyd,26776.html)

Justin is a driven, entrepreneurial, serial CTO with 35 years in the business of video games, developing and delivering titles on 26 platforms. He is creator/co-creator of dozens of land-based casino and social casual casino games, successfully developing a number of F2P and real-money gaming related properties.

He has worked on such franchises as Civilization, Pitfall, Hotwheels, Tyco, Star Wars, Shrek and Battletanx and with Disney, eBay, Microsoft, Apple, Lucasarts, Activision, Mattel, Vivendi Universal and SONY as clients. Justin has extensive experience with consumer facing commercial software, large-scale websites and other types of digital and physical products.



[More about Justin Lloyd... \(/expert/justin\\_lloyd,26776.html\)](/expert/justin_lloyd,26776.html)

[Larry Schiefer \(/expert/larry\\_schiefer,25019.html\)](/expert/larry_schiefer,25019.html)

Larry has spent over 16 years leading teams, designing and developing software and embedded systems for multiple markets. His ability to understand complete systems and the software stack has lead to high quality and high performance designs. In addition to developing device drivers, platform software and applications, Larry has developed and presented training for internal teams,

end customers and technical conferences on highly technical content. Larry is currently the CTO and co-founder of [HiQES \(http://hiqes.com/\)](http://hiqes.com/), a engineering services company.

Five years ago, Larry was tasked with leading a new team to do Android platform software development, so he has experience diving headlong into new technology. Larry spearheaded internal ramp-up of developers, developing Android kernel drivers and middleware, and native code starting with the first public releases of Android. In fact, Larry was developing native C/C++ services and apps for Android before the first release of the Android NDK! Since that time, Larry has been the lead developer on three different new Android platforms, multiple Android platform extensions and applications.



[More about Larry Schiefer... \(/expert/larry\\_schiefer,25019.html\)](/expert/larry_schiefer,25019.html)

[Marius Mailat \(/expert/marius\\_mailat,14878.html\)](/expert/marius_mailat,14878.html)

Marius Mailat has been developing commercial software since 2000. His focus is mobile applications architecture and development for Android, QNX and iOS.

He is an experienced Java developer who spent the majority of his career building large-scale Java applications for Xerox and Sabre Travel Network in Germany.

Android projects in which he has been involved are banking interfaces and shopping tools for Android. His interest in the secrets of Android platform was geared towards building Defense Squad Mobile Security app, a free, simple to use and innovative Android security solution. Marius is the founder of the Romanian Android Community Androider.ro which has now around 5000 active members.

[More about Marius Mailat... \(/expert/marius\\_mailat,14878.html\)](/expert/marius_mailat,14878.html)

[Ron Munitz \(/expert/ron\\_munitz,28298.html\)](/expert/ron_munitz,28298.html)

Ron is an entrepreneur and software development consultant. He has a long history of developing for performance and safety critical software, leading development groups, training application and platform developers, and helping companies in the process of board bring-ups. He also has experience with the design and execution of embedded systems products, security best practices and product start-up.

He specializes in all aspects of distributed systems and Android internals. As the Founder and CTO of Nubo, the first Remote Android Workspace for the Enterprise, he is the designer and original developer of the first Android Remote Display Protocol.



Additionally, Ron is a lecturer at Afeka's college of Engineering, where he has developed and taught the first ever academic Android Internals Course. Ron holds a Master's of Computer Science from Bar-Ilan University.

## Outline

[\[top\]](#)

### **Android Overview Module (/training/android/419.html)** (1 hour)

This module introduces you to the Android operating system to ensure a basic familiarity with the Android architecture, background, and terminology. You also learn additional sources of information for modifying the Android OS, porting it to new hardware, and complying with Google's Android compatibility requirements.

- Overview of the Android platform architecture
- Introduction to the structure and lifecycle of Android applications
- Internet resources for Android platform developers
- Android compatibility requirements: the Compatibility Definition Document (CDD) and Compatibility Test Suite (CTS)

### **Android Stack Module (/training/android/440.html)** (1 day)

The Android OS is categorized into four layers. From the bottom up they are the kernel layer, the user-space native layer, the application framework layer, and the applications layer. In this module you explore each of the layers in depth to understand their role and the components they contain.

### **Android Kernel Layer (/training/android/421.html)** (3 hours)

In this module you learn how Google has extended the standard Linux kernel for Android and how the upper layers of the Android OS interact with these extensions.

- Where to find the source for Android's custom kernel
- Binder: Android's primary inter-process communication (IPC) mechanism
- Anonymous shared memory (ashmem): Android's replacement for POSIX SHM
- Android's physical memory allocator, ION
- Android power management extensions: wakelocks, early suspend, and alarms
- Android's low-memory process killer
- The Android logging system

- Android's user and group management and kernel security enforcement (the "paranoid network security" kernel option)

### **Android Native Layer (/training/android/422.html) (2 hours)**

In this module you explore the portion of Android user-space that is implemented in native C/C++ code. You learn: in what ways Android's libc implementation, Bionic, is not POSIX-compliant (and why); how Android's user-space Hardware Abstraction Layer (HAL) defines a standard API for exposing hardware to the rest of the Android OS; what significant libraries, frameworks, and daemons are incorporated in Android; and how Android's Dalvik Virtual Machine (VM) enables both application and platform developers to write Android code in Java.

- Bionic: Android's implementation of libc and how it differs from the BSD libc
- User-space Hardware Abstraction Layer (HAL): standard APIs for accessing hardware
- Overview of standard Android native daemons (e.g., adbd, rild, ueventd, etc.) and their purpose
- Overview of other Android function libraries and frameworks
- The Dalvik Virtual Machine (VM) and how it differs from a Java VM

### **Android Application Framework Layer (/training/android/423.html) (1 hour)**

Android's application framework layer not only exposes a Java API for application developers, but also implements much of the Android operating system in Java. In this module, you discover how Java-based system services manage the operating system, applications, and device hardware. You also learn the general model by which client-side manager classes interact with these system services through Binder IPC to access system features.

- System service architecture
- Significant system services and their roles (e.g., ActivityManagerService, PackageManagerService, ConnectivityManagerService, etc.)

### **Android Applications Layer (/training/android/430.html) (1 hour)**

This module describes the basic structure of an Android application and how it is distributed as an Android application package (APK). It also identifies the standard set of Android system applications and where they are implemented in the source distribution.

- Android application structure



- Standard Android system applications, wallpapers, and input method editors

### **Java Native Interface (JNI) and the Android Native Development Kit (NDK) (</training/android/431.html>) (4 hours)**

Much of the Android platform is implemented in Java, and yet that code needs to access functionality written in C/C++ and compiled to native machine code. Additionally, an application developer may want to implement portions of an app in C/C++ for performance or to incorporate existing native libraries and frameworks.

Java Native Interface (JNI) is a standard Java technology for integrating Java and native code. In this module, you learn the fundamentals of JNI and how to use it as a bridge between the native and Java-based runtimes. You also learn how to use Android's Native Development Kit (NDK) to implement portions of an application in native code and distribute it for use on multiple machine architectures.

- JNI development process overview
- Implementing Java methods in native code
- Mapping Java types to native types
- Managing object references in native code
- Managing Strings, arrays, and other Java object types in native code
- Throwing and catching exceptions in native code
- Using NDK to incorporate native code in an app
- Supporting multiple machine architectures with NDK
- NDK "stable" APIs
- Lab: NDK

### **Android Inter-Process Communication (IPC) with Binder and AIDL (</training/android/432.html>) (4 hours)**

Binder is Android's primary inter-process communication mechanism. (In fact, most standard POSIX IPC mechanisms are not available for use in Android.) Even higher-level Android IPC techniques, such as broadcast Intents and interactions between system services and client-side manager classes, use Binder as the underlying transport mechanism.

In this module, you learn: the capabilities of Binder; how to generate Binder-based interfaces in Java using Android Interface Definition Language (AIDL); how Binder is used by client-side manager classes to communicate with system services; and how applications can expose their own Binder-based interfaces to other applications.

- Overview of Binder and its capabilities
- Higher-level Android IPC mechanisms based on Binder
- Binder communication and service discovery
- Generating Binder-based service interfaces in Java using Android Interface Definition Language (AIDL)
- Creating custom Parcelable Java types for use with Binder
- Exposing a Binder-based interface from an application
- Asynchronous Binder interactions
- Binder limitations
- Binder security
- Detecting and handling Binder "death notifications"
- Lab: Binder-based application service

### **Android Security Essentials (/training/android/433.html)** (3 hours)

Android extends standard Linux security to control access to device features like network interfaces, cameras, and stored personal information. In this module you learn how Android's permission model interacts with standard Linux security and how to define and enforce custom permissions to restrict access to system extensions.

- User and group ID management
- Android permission enforcement
- Declaring custom Android permissions
- Lab: Custom permissions
- Securing application components using permissions

### **Building Android from Source (/training/android/434.html)** (2 hours)

In this module you learn how to set up an Android build system, download the Android source, build Android system images, and run them on emulators and real hardware.

- Setting up an Android build system
- Obtaining the Android source tree
- Selecting the target product and build variant

- Building Android system images from source
- Running custom Android images on emulators and real hardware

### **Android Startup (/training/android/435.html) (2 hours)**

This module describes Android system startup including bootloading the kernel, launching standard Linux daemons, and initializing a variety of Binder-based system services. It also explains the importance of the Zygote daemon for reducing Android application startup time and memory consumption. Additionally, you learn how to customize the system boot process through custom init scripts.

- Bootloading the kernel
- The init process and Android's init scripting language
- The standard boot process and how to customize it
- The purpose of the Zygote daemon
- Startup of system services

### **Android Subsystems (/training/android/436.html) (3 hours)**

This module presents an architectural overview of several of the most significant Binder-based system services in Android, explores interactions between services and client processes and between the various Android platform layers, identifies key source files in the implementation of the services, and provides references to additional resources for subject matter experts to extend and modify many services.

- Vibrator
- Power Service
- Alarm Service
- Package Service
- WiFi Service
- Location Service
- Android Media Framework
- Telephony
- Device Policy Service
- SurfaceFlinger

- Camera Service
- NFC Service

### **Creating a Customized Android System Image (/training/android/437.html) (1 day)**

This module integrates concepts from the entire course in the hands-on creation of a custom Android system image. You learn how to implement customizations at all Android stack levels, including custom kernels, HAL user-space libraries, executables, daemons, Java libraries, system applications, and Binder-based system services. Additionally, you'll learn how to support third-party developers for your Android devices by creating an SDK add-on that exposes your custom Java APIs and provides developers with a custom image they can use to create Android Virtual Devices (AVDs) on which they can run and test their code.

- Setting up a custom device directory structure
- Registering a custom device with Android's build system
- Adding the Makefile plumbing for a device
- Generating platform signing keys (Optional)
- Adding a custom kernel
- Adding a custom native library and executable
- Adding a custom daemon
- Creating a custom Java library to expose a native library (i.e., JNI in the platform)
- Consuming a custom Java/JNI library via a custom application (Optional)
- Implementing a custom Binder-based system service
- Building a custom application using a custom client-side manager class
- Creating and distributing a custom SDK add-on to support third-party developers (Optional)

### **Android Tools and Debugging (/training/android/438.html) (2 hours)**

In addition to standard utilities for monitoring and troubleshooting Linux-based systems, Android includes several custom tools of its own. This module shows you how to use these tools to monitor and troubleshoot the Android kernel, processes, system services, and applications.

- Debugging native code on Android
- Debugging Java code on Android
- Debugging applications and system services

## Additional Notes

[\[top\]](#)

### Trademark Notice

Android is a trademark of Google Inc. Use of this trademark is subject to [Google Permissions](http://www.google.com/permissions/index.html) (<http://www.google.com/permissions/index.html>).

## Student Testimonials

[\[top\]](#)

“Marko was great, he explained stuff really well, he gave personal attention to everybody in the class. He knew his stuff very well and he kept the class at the right pace. Thanks Marko!! It was great having you here..”

### Rohit - QIS Eng Test

“Slides were detailed and covered everything, we were provided solutions to each lab so we don't get stuck, the instructor was very knowledgable. Overall, this course is significantly better done than typical courses.”

### David - Intel Corporation

“This was a great class. Very thoroughly laid out and well covered.”

### Brenda - IONU Security

“Sasa is very good at explaining Android topics and gave us an excellent overview in a short time frame.”

### David - Vulnerability Research Labs

“The class provides good, reliable and complete materials. Ron, the instructor, is very good at what he's talking about and is able to share his knowledges clearly and efficiently.”

### Charles - Qualcomm, Inc.

“Very informative and up to date session, recommended for those who want to get more knowledge in Android. Looking forward for more sessions with NewCircle!!”

### Ong - Razer Pte Ltd

“Quite a lot gets covered in 2 days. Will give any one a quick start. Thank you, enjoyed it.”

**Nanda - Dreamer Corporation**

“I attended the Android Internals Training in October 2014 and found the course excellent. The structure was interesting and informative and thoroughly detailed. Ron is very experienced and very knowledgeable and has a good sense of humor. Material covered in the course is very vast and detail oriented but with a combination of lectures, slides and labs, it was organized and the flow was smooth. I took many tips and lots of information and definitely plan to apply to my work.”

**Radhika - Lab126**

“I thought the class moved at a good pace. I am glad that the instructor did not expect us to be Java programmers and made us code in Java all the time. It made the class more enjoyable.”

**Harold - Research Dept.**

“Class content: Worth every penny. Dave delivers the technical details that can accommodate beginners or savants. If you didn't double your Android knowledge after taking this class, you were sleeping.”

**Robert - Optics1 Incorporated**

“It is hard to rate everything as excellent but this really was an excellent course. Marko was very knowledgeable and knows how to present this work. I will surely benefit from having had this course. Great class and excellent instructor.”

**Prithvi - Texas Instruments**

“fast pace, compressed learning.”

**Bastiaan - Intel Corporation**

“Larry's an excellent instructor who had enormous patience and responded to my every question in a very easy to understand manner! I thoroughly enjoyed his class and understood the internals concepts very well. This should enable me to work on Android related projects at work with more confidence. I'll definitely recommend him as an instructor for any other related courses....! BTW, the course material was nicely crafted too. Thanks!”

**Kittur - Intel Corporation**

“Actually, I am now wondering to attend your 5 days training to learn more about Android application development.”

**Lucas - Qualcomm, Inc.**

“Excellent instructor. Knowledge of the subject is clearly extensive. Very good course material, I will definitely keep it for reference. Appreciated the tailoring of the material to our needs.”

**Ranjan - Intel Corporation**

“Marko was an excellent instructor. I feel very confident with the information I've learned over the past few days. Thanks.”

**Patrick - The MITRE Corporation**

“The course was very good. It covered a lot of material well. The instructor was excellent. A very good presenter, teacher, and a likable person.”

**Jeffrey - Intel**

“The class material was detailed and the instructor very informed. Answered all our questions quoting relevant examples to better our understanding of various concepts of Android internals.”

**Shalini - Intel Corporation**

“The training was extremely useful, showing Android concepts that I daily need to use in my job.”

**Lucas - LG Rochaverá, R&D Center**

“Learned a lot in this class and peaked my interest so much that I am going to pursue Android going ahead in my career.”

**Ranganath - Intel Corporation**

“The class was extremely thorough. I was impressed how well the instructor knew the material and had well thought out answers to every question we threw at him.”

**Chris - Lab126**

“This is a great course, very informative with hands on, customized examples. Gil knows the information that he is teaching and does a great job at explaining it. He also shows examples in the code when there are questions relating to the code.”

**Lynette - Intel Corporation**

“Very useful course, getting to the point of Android internals with an evident link between theoretical stuff and practice”

**Efrain - Intel Corporation**

“Larry, you did a great job with the class and you know your stuff.”

**Tom - Intel Corporation**

“Very good.”

**Dongchun - QRD**

“Dave is a great instructor. He is very knowledgeable in all aspects of the android system. Highly recommend this class to anybody who wants to understand the inner workings of the android platform.”

**Alexander - Fusion Alliance**

“I think the course content was very in-depth and allowed for a solid, well-rounded understanding of how the Android framework is set up from the bottom up. The labs really allowed us a hands-on experience to further re-enforce this, and was very beneficial. In addition, the instructor was super, super knowledgeable about almost everything Android!”

**Rhea - Amazon.com Inc.**

“I think the training was very valuable for me and it will be very helpful to my feature work.”

**Michiyuki - Qualcomm, Inc.**

“Thanks for jumping into example code to answer our questions.”

**Julie - Texas Instruments**

“I think Marko is a skilled trainer, excellent class driving with excellent background, Marko rocks!”

**Alfredo - TI**



---

[Paid Training](#)



---

[About](#)



[BACK TO TOP](#)