

# Android System Development Workshop



[www.easyarm.com](http://www.easyarm.com)

Enabling “Embedded Learning in INDIA”



## Workshop

When people want to work parallel with the growing technology, learning the new technology is primary step. To help all, EasyARM came-up with two days workshop which is focused to give the overview of Android and explore its building blocks. Primary elements to start working on Android with a hands-on experience. The targeted audience should have prior knowledge in Linux Basics and good C, C++ or Java Programming skills.

**Fee & Duration: [ Bangalore Only ]**

Rs. 3000.00 , 2 Days [ 10:00 AM to 4:00 PM ]

## Registration:

Drop a mail to [info@easyarm.com](mailto:info@easyarm.com) with your contact details.

For any query please call on 080-41307589 || 9972039671



[www.easyarm.com](http://www.easyarm.com) || [info@easyarm.com](mailto:info@easyarm.com) || +91-80-41307589 || +91-9972039671

## Day-1

### Android Anatomy

1. Introduction & history of Android
2. Native Libraries
3. Android Runtime
4. Application Framework

### Android Kernel

1. Introduction
2. Binder
3. Power
4. Ashmem
5. Low Memory Killer
6. Logger, ADB
7. Miscellaneous Patches

### Getting Started

1. Setting up development Machine
2. Packages required on development machine
3. Hands On (Demonstration)

### Repo & Git - Overview

1. Manifest file
2. Working with repo and git
3. How to Download Build & Compiling

### Android Emulator on Windows and Linux Host

1. Introduction
2. Goldfish & QEMU
3. Working with emulator
4. Connecting to emulator over ADB
5. Hands on with Eclipse SDK emulator (Demonstration)

### Android Runtime

1. Introduction to Dalvik/Zygote
2. How Android framework starts
3. app\_runtime -> zygote
4. System server, Android services
5. Instances of Dalvik
6. Hardware abstraction layer

## Day-2

### Android Toolchain

1. Introduction
2. Features of Android Toolchain
3. Dependency of libc with android build

### Android Native Layer Development

- Overview of Android C/C++ Libraries
- Modifying C/C++ Libraries
- Porting New C/C++ Libraries
- Cross Compiling a C Program and executing on Target.

### Preparing and Porting Android for ARM Board

- Download Android kernel
- Compile Android kernel
- Prepare Android filesystems
- Porting applications on ARM Board

## Registration Process

**Payment Mode: [Complete Payment should done be in advance]**

### Venue Details:

**# 9/1, 1st Floor,3rd Main, 8th Block, Opp: Police station, Kormangala, Banalore-560095.**

**Note: Candidate should carry Laptop with minimum of 1GB Ram and 10GB free space.**

## FAQs

### 1. What is Android ?

Android is a mobile Operating System.

### 2. In which area of Android Can I work ?

Android UI Programming: If you are a java UI programmer.

Android Java Framework: For Java programmers.

Android JNI Layer: For Java , C & C++ Programmers.

Android Native Development: C , C++ & Linux Programmers.

Android Kernel, BSP & Device Drivers: C, C++, Linux System & Device Driver Programmers.

### 3. Which area is covered in this workshop ?

An overview of all layers have been covered and major part concentrated on the lower layer which is C, C++ & Linux.

### 4. What should I know to attend this workshop ?

This is a introductory workshop for Android which covers all parts of Android, so you should fit in any one Layer on question 2.

### 5. What all things I know after this workshop ?

Big picture of Android.

Layers & Modules of Android Framework.

Setup the work environment for Android.

Executing Sample Java Applications on real device.

Android Runtime flow.

Writing & Executing Native C/ C++ Programs on real device.

Host and Device communication using Android ADB.

Compiling Android for a Hardware and Bring-up different targets.

### 6. What next to this workshop ?

You are ready to experiment on different layers of Android as per your domain.

EasyARM conducts advance workshops on Android.

1. " Android Application Development " ( Java Programmers )

2. " Android Advance Programming" ( Java Programmers )

3. " Android Porting, BSP, Device Drivers & Native Framework Development" ( C \ C++ & Linux programmers )