

SMARTPHONES

Revised 2/29/2012

/training/etc

The Art of Knowledge.

This Page Intentionally Left Blank

Table of Contents

Android™

Introduction to Android™	1
Advanced Android™ (Outline and Instruction by Marakana).....	2
Android™ Bootcamp Training (Outline and Instruction by Marakana).....	3
Android Internals.....	4

iPhone and iPad

Objective-C and iPhone Bootcamp.....	5
--------------------------------------	---

This Page Intentionally Left Blank

Course Description:

Introduction to programming for the Android platform course is designed to quickly get you up to speed with writing apps for Android devices. You will learn the basics of the Android platform, and gain an understanding of the application lifecycle. By the end of the course, you will be able to write simple GUI applications, use built-in widgets and components, and much more. This is a perfect course to get started with Android programming.

Who Should Attend:

This course is for students who wish to get up to speed with writing apps for Android devices.

Prerequisites:

Students should have basic Java programming skills, or equivalent OOP language experience.

Benefits of Attendance:

Upon completion of this course, students will be able to:

- Understand how to set up the Android SDK Environment.
- Use Eclipse for Java to develop on emulators and handsets.
- Understand the structure of an Android application.
- Write GUI applications.
- Use built-in widgets and controls.

Course Outline:**Introduction to Android**

Overview
The Android Architecture
Fundamentals of an Android Application
Lifecycle of a Component
The User Interface
User Interface Events

The Android Development Environment

System Requirements
Downloading Java
Downloading the Java Documentation
Installing Java
Installing the Java Documentation
Configuring the PATH Environment for Java
Downloading and Installing Eclipse
Downloading the Android SDK
Configuring the Android SDK
The Android Development Tools (ADT) Plug-in
Installing the ADT Plug-in
Configuring the ADT Plug-in
Creating an Android Virtual Device (AVD)

Your First Android Application

Creating a New Android Project
Running an Android Project in the Emulator
The Generated Project Files and Resources
Modifying the Android Project
Java Packages vs. Android Packages

Android Activities

Applications and Their Activities
A Sample Activity
Running the Sample Activity
The Activity Stack
Lifecycle Callback Methods
Adding Logging Support
A Lifecycle and Logging Example
Viewing the Log Files (The DDMS Perspective)
Saving State on Configuration Changes
The Bundle Class

Intents

What is an Intent?
Launching an Activity by Class Name
Implicit Intents
Returning Results from an Activity

The User Interface

Overview of Views
Java vs. XML Resources
Referencing Layout Resources Programmatically
Text Controls
Button Controls
Button Interaction
Date and Time Controls
Layouts and ViewGroups
Weight and Gravity

LinearLayout ViewGroups
RelativeLayout ViewGroups

Menus and Dialogs

Menus
Dialogs
Alert Dialogs
Progress Dialogs

Data Storage and Retrieval

Overview
Shared Preferences
Internal Storage

Course Description:

Advanced Android™ training is taking mobile application development to the next level. You will learn how to create custom widgets, create animations, work with cameras, use sensors, create and use advanced content providers, and much more. The course goes into testing and deployment of Android applications, as well. This course is also part of the Android Bootcamp.

Who Should Attend:

This course is for students who are familiar with basic Android development practices and want to learn more advanced concepts.

Prerequisites:

Students should have Java experience and familiarity with basic Android development practices. They should have taken Intro to Android™ training or have equivalent experience.

Benefits of Attendance:

Upon completion of this course, students will be able to:

- Create custom widgets.
- Create animations.
- Work with cameras.
- Use sensors.
- Create and use advanced content providers.

Course Outline:**Lists and Adapters**

TimelineActivity
Basic TimelineActivity Layout
About Adapters
Timeline Adapter
ViewBinder: A Better Alternative
Updating Manifest File
Base Activity

Broadcast Receivers

About Broadcast Receivers
BootReceiver
The TimelineReceiver
Broadcasting Intents
The Network Receiver
Adding Custom Permissions

Content Providers

Creating Content Provider
Using Content Providers Through Widgets

System Services

Compass Demo
Location Service
Updating Yamba to User the Location Service
Intent Service
Sending Notifications

Course Description:

Android™ Bootcamp Training Course is a hands-on training for designing and building mobile applications using Android™ open-source platform. Android™ Bootcamp course explains the philosophy of developing for Android™ through its main application development building blocks and their interaction with one another.

This complete hands-on course encourages students to learn by building increasingly more sophisticated and meaningful mobile applications for Android™.

By the end of the course, each participant will build their own complete Android application incorporating most of the key aspects of the platform. Typically, we build a Twitter app for Android, but there are other choices depending on participants' interests.

This course has been updated for Ice Cream Sandwich.

Who Should Attend:

This course is designed for software developers interested in designing, creating, deploying, and testing applications for the Android™ mobile phone platform. It is valuable to both novices and gurus, who already have experience in developing mobile applications for other platforms.

Prerequisites:

Java experience is required to get the most benefit from this training.

Benefits of Attendance:

Upon completion of this course, students will be able to:

- Build your own Android apps.
- Understand how Android™ applications work, their life cycle, manifest, intents, and using external resources.
- Design and develop useful Android™ applications with compelling user interfaces by using, extending, and creating your own layouts and Views and using Menus.
- Take advantage of Android's Application Framework API to build complex applications.
- Utilize the power of background services, threads, and notifications.
- Use Android's communication APIs for SMS, telephony, network management, and internet resources (HTTP).
- Secure, tune, package, and deploy Android™ applications.

Course Outline:**Android Overview**

Android Overview
History
Android Versions

The Stack

Stack Overview
Linux
Native Libraries
Dalvik
App Framework
Applications

Quick Start

Installing SDK
Hello, World!
The Emulator

Main Building Blocks

Main Building Blocks Overview
A Real World Example
Activities
Intents
Services
Content Providers
Broadcast Receivers
Application Context

Yamba Project Overview

The Yamba Application
Project Design
Part 1: Android User Interface
Part 2: Preferences, File System, Menus
Part 3: Android Services
Part 4: Working with Databases
Part 5: Lists and Adapters
Part 6: Broadcast Receivers
Part 7: Content Providers
Part 8: System Services

Android User Interface**Two Ways to Create User Interface**

Views and Layouts
Starting Yamba Project
StatusActivity Layout
StatusActivity Java Class
Logging in Android
Threading in Android
Other UI Events
Adding Color and Graphics
Alternative Resources
Optimizing User Interface

Preferences, File System, Options Menu

Preferences
Options Menu
Shared Preferences
File System, Explained

Services

Yamba Application Object
UpdaterService
Looping In The Service
Pulling Data From Twitter

Databases

About SQLite
DbHelper
First Example
Update UpdaterService
Refactoring Status Data

Lists and Adapters

TimelineActivity
Basic TimelineActivity Layout
About Adapters
Timeline Adapter
ViewBinder: A Better Alternative
Updating Manifest File
Base Activity

Broadcast Receivers

About Broadcast Receivers
BootReceiver
The TimelineReceiver
Broadcasting Intents
The Network Receiver
Adding Custom Permissions

Content Providers

Creating Content Provider
Using Content Providers Through Widgets

System Services

Compass Demo
Location Service
Updating Yamba to Use the Location Service
Intent Service
Sending Notifications

Course Description:

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

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

Who Should Attend:

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.

Prerequisites:

To get the most benefit from this class you must have a basic understanding of C and Java. Stronger proficiency with C, C++, Java, and Make is recommended.

Benefits of Attendance:

Upon completion of this course, students will be able to:

- Explain the anatomy of the Android platform and get its 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

Course Outline:**Android Stack**

Android Linux kernel Layer
Android User-space Native Layer
Application Framework Layer
Application Layer

Java Native Interface (JNI) on Android using Native Development Kit (NDK)

Overview of NDK
Motivation for NDK
Java Native Interface (JNI)
Using NDK
Example of developing with NDK on Android
Threading on Android: loopers and handlers, UI thread-safety, AsyncTask
Overview of NDK make files: Android.mk, Application.mk
Compilation options: controlling CPU Application Binary Interface (ABI)
NDK Stable APIs: Logger, ZLib, OpenGL, jnigraphics, OpenSL, native application APIs
Debugging and best-practices

Android Binder Interprocess Communication (IPC) with AIDL

Overview of IPC
Motivation for IPC on Android
Overview of Binder workflow and its responsibilities
Overview of Android Interface Definition Language (AIDL)
Example of developing with Binder IPC and AIDL
Best practices

Android Security

Android Security Architecture and Philosophy
Application signing
User IDs
File Access
Using Permissions
Permission Enforcement
Custom Permissions
Requiring permissions in our code
Enforcing permissions dynamically
ContentProvider URI permissions
Securing our apps (public vs private components, protecting BroadcastReceivers, PendingIntents)
Overview of Android Exploits
Defending against buffer-overruns with memory protection
Tap-jacking on Android
Other security concerns: encryption, device management, anti-malware, firewall, SE-Linux, etc.
Security Lab: Securing an Android Application

Building Android From Source

Motivation for building from source
Initializing the build environment: Linux, MacOS X
Downloading the source: overview of git and repo, manifest, branches, etc.
Android source-code structure
Building Android image
Understanding built images: ramdisk.img, boot.img, system.img, userdata.img, recovery.img, etc.
Running custom Android images with the emulator
Flashing custom Android images on real hardware
Building and using a custom Linux Kernel

Android Startup

Bootloading the kernel
Android's init startup
Zygote startup
System Server startup
Automatic startup of applications

Customizing Android

Overview of Android Compatibility Test Suite (CTS)
Overview of Android build system (make files)
Creating a custom daemon
Creating a custom native shared library (wrapping a Linux device driver)
Creating a custom Java library (wrapping custom native shared library with JNI)
Creating a custom System Service
Creating a custom Android application (to be installed on the ROM under /system/app)
Customizing Android Lab

Utilizing USB Devices from Android - Appendix

Overview of USB
Android as a USB Host
Android as a USB Accessory

Power Management on Android - Appendix

Overview of Linux PM
Android vs. Linux PM (wake-locks revisited)
Power management stack
Power management effect on drivers (notification of power-events and usage of wakelocks)

Media Services on Android - Appendix

Overview of Media Framework on Android
Understanding the role of the mediaserver
Exploring Stageright and building HW codecs based on OpenMAX IL
Stepping through audio (capture and playback) on Android -

AudioFlinger, Audio Policy Services, libaudio, ALSA, etc.
Using OpenSL ES from NDK-written applications
Understanding camera capture on Android (including preview and control)
Overview of DRM on Android

Connectivity Services on Android - Appendix

Overview of connectivity services on Android
Basic connectivity state management and failover
Wifi connectivity stack
Telephony connectivity stack (including RIL)
Bluetooth stack
GPS stack

Graphics on Android - Appendix

Display drivers (frame buffers)
Surfaces and Surface Flinger
OpenGL

Other Devices on Android - Appendix

Input devices
Sensors
Lights
Vibrator

Automated Testing on Android - Appendix

Overview of unit/functional testing
Overview of jUnit
Overview of Android Unit Testing framework
Overview of Monkey and Monkey Runner
Best-practices for testing

Course Description:

Understanding the Objective-C programming language is critical to becoming a successful iPhone developer. This class is designed to teach you a solid foundation in Objective-C 2.0 from the ground up. This course does not require or assume any background in Objective-C, C, C++ or Object Oriented Programming (OOP) in general. You will learn how to enter, develop, and debug your programs under Mac OS X using the Xcode development tool.

In the second part of the course, students will learn how to write iPhone / iPad Applications with the SDK!

Who Should Attend:

This course is for programmers who want to write applications for Mac OS X, the iPhone, the iPod Touch, or the iPad. Some prior programming experience, preferably with an object-oriented programming language.

Prerequisites:

Students should have some prior programming experience, preferably with an object-oriented programming language.

Benefits of Attendance:

Upon completion of this course, students will be able to:

- Define new classes and write programs in Objective-C
- Compile, edit, and run Objective-C programs under Mac OS X
- Write subclasses, categories, and understand the principles of inheritance
- Understand how dynamic typing works and the principle of polymorphism
- Use the Foundation Framework classes to work with numbers, strings, collections, and files
- Archive and copy objects and to thoroughly understand the principles of memory management
- Get started writing applications for the iPhone, iPod Touch, and iPad
- Write iPhone and iPad applications
- Use XCode to manage the application development project, and to compile, edit, and debug your applications
- Run your application on a device
- Use Interface Builder to design a user interface
- Integrate iPhone interface elements like buttons, switches, and sliders into your application
- Work with views and view controllers
- Work with navigation controllers, tables, and pickers

Course Outline:**Getting Started in Objective-C**

What is Objective-C?
Your First Program
Steps for Using Xcode
What is a Class, Object, or Method?
Defining a Class to Work with Fractions
Accessing Instance Variables

Data Types and Expressions

Basic Data Types
Arithmetic Expressions
Defining a Calculator Class

Loops and Making Decisions

The for statement
The while statement
The do statement
The if Statement
The Switch Statement
The Conditional Operator and Boolean Variables

More On Classes

Properties, Synthesized Accessor Methods, and the dot Operator
Multiple Method Arguments
Passing Objects as Arguments
Local Variables; The self Keyword
Allocating and Returning Objects in Methods

Inheritance, Categories, and Protocols

Inheriting Methods and Instance Variables
Subclasses: Extension through Inheritance
Classes Owning Their Own Objects; Overriding Methods
Categories
Protocols

Polymorphism, Dynamic Typing and Dynamic Binding

Polymorphism; Static Typing
Dynamic Typing and Dynamic Binding

The Preprocessor and Underlying C Language Features

Preprocessor
Arrays
Functions
Structures
Pointer
Unions
Miscellaneous Language Features

Numbers and Strings

Number Objects
String Objects

Collections

Array Objects
Fast Enumeration
Sorting Arrays
Dictionary Objects
Set Objects

Working with Files

Managing Files and Directories: NSFileManager
Working with Paths: NSPathUtilities.h

Memory Management

The autorelease Pool
Reference Counting
Summary of Memory Management Rules
Garbage Collection

Copying Objects

The copy and mutablecopy Methods
Shallow Versus Deep Copying
Implementing the NSCopying Protocol
Copying Objects in Setter and Getter Methods

Archiving Objects

Archiving with XML Property Lists
Archiving with NSKeyedArchiver
Writing Encoding and Decoding Methods
Using NSData to Create Custom Archives
Using the Archiver to Copy Objects

Writing Your First iPhone Application

Introduction to Cocoa Touch
The iPhone SDK
Your First iPhone Application

Web-based vs. Native Applications

The Difference Between the Two

Cocoa, Cocoa Touch and the iPhone SDK

Definitions

iPhone OS Frameworks

What is a Framework?
Overview of iPhone SDK Frameworks

iPhone Application Development Overview

Overview of the Tools and Process

Some Quick Reviews

Objective-C
Protocols
Memory Management

Writing Your First iPhone Application

Outlets and Actions
Introduction to Interface Builder

Using the iPhone Simulator
Extending your Application: Adding an Application Icon

Simple Debugging Techniques

Using the Preprocessor
NSLog and the Description Method
Using Breakpoints

Running Your App on a Device

The iPhone Provisioning Portal
Developer Certificates, App IDs, Device IDs, Provisioning Profiles
XCode's Organizer Window

Application Execution and MVC

Applications and Events
The Model-View-Controller Design Pattern

Project Lab: Calculator

Create a Working Calculator App

Screen Geometry

The iPhone and iPad Coordinate System
The frame, bounds, center, and origin
Supporting Structures and Functions
Applications and Events

Understanding Views and View Controllers

Properties of Views
Subviews and Superviews
Creating a View Controller
Multiple View Controllers
Creating Views Programmatically
Understanding a View-Based Application
Manipulating Views: Moving and Hiding
UIImageView – Working with Images
Adding Simple Animation
Handling Device Rotation and Control Resizing and Repositioning

Project Lab: State Capitals

Understanding Delegates
Using the UIPickerView class
Reading Data from the Internet

Project Lab: The Birds

Creating a Navigation-based Application
Working with a Table View Controller
Populating Data in a Table
Handling Row Selection
Working with UITextView's
Playing Audio and Video Files
Saving an Image to the Photo Album
Displaying Alerts
Creating a Default.png screen

Project Lab: The Gear

Transforming Images
Using Timers
Simple Animation
Saving and Restoring the Application State
Handling Touch Events
Responding to Shake Gestures
Working with Sliders and Switches
Adding Sound Effects

Programming for the iPad

Application Design Decision: Universal vs. Separate Apps
Examples

And More....

Additional Topics, Time Permitting