Born on 23rd September 2008

hola android!

Born on 9 January 2007

my first love

You don’t need to own an android phone to do the class!

some factoids
What's in a smartphone - the hardware inside?
And, why is it such a revolutionary device?
What software runs on phones?
How do I code apps and distribute them to the masses?
Oh, and how do I get rich... quickly.
Where does the vision lead?

what’s under the hood?

The naked iPhone 3GS

Things you can do
what’s your favorite app?

Things you can do

what makes things different is the embedded sensors

Nokia workshop on large-scale sensor networks, February 2005, Kuusamo, Finland
gyroscope, barometer, compass

two microphones

how do social "conversation networks" evolve?

SoundSense
Hong Lu, Wei Pan, Nicholas Di Lame, Tanveer Chowdhury, Andrew T. Campbell

SoundSense

location (GPS, wifi, cellular)

front and back cameras
collocation nets (BlueTooth)

near-field communications (NFC)

making sense of data
where does the vision lead?

phones are open and programmable

“massive markets driving innovation - some 15 year old will release the equivalent of facebook for phones in the next 3 years”, Andrew T. Campbell, January 5, 2011

density leads to big data

learning, big data, mining, apps
we want to push intelligence to the phone

- sensing
- feature extraction
- inference
- learning/adaptation
- app specific
- privacy
smartphones are getting smarter; at some point they’ll:
- understand our behavioral patterns
- mimics human perception
- anticipate our every move
- help us navigate our day
- become integrated into the fabric of our lives

ultimately leading to the “cognitive phones”.

In your lifetime ...

things will look different
we'll interact with smartphones in new ways

smartphone sensing group

factoids on androids ;-)

Android applications are written in the Java

The Android SDK tools compile the code—along with any data and resource files—into an Android package (file.apk) which is considered to be one application and is the file that Android devices use to install the application.
This creates a very secure environment in which an application cannot access parts of the system for which it is not given permission.

There are ways for an application to share data with other applications and for an application to access system services; for example, an application can request permission to access device data such as the GPS, user’s contacts, SMS messages, the mountable storage (SD card), camera, Bluetooth, and more.

All application permissions must be granted by the user at install time.

Android 4.0, Ice Cream, API 14 Sandwich

**API numbers and OS targets**

Each version of the Android OS is identified by an API level number:

Two separate targets are applicable:

- Android SDK Platform contains all the Android APIs
- Google APIs by Google Inc contains all the Android APIs and Google Maps APIs

Android 1.0, the first commercial version of the software, was released on 23 September 2008

**getting started: installing the platform**


Step 1: You might need to install the JDK
Step 3: Install Android SDK
Step 4: Install Android Development Tools (ADT) custom plugin for the Eclipse IDE

step 5: adding platforms and other components
application components

activities
An activity represents a single screen with a user interface. For example, an email application might have one activity for checking a list of new emails, another activity to compose an email, and another activity for reading emails.

services
A service is a component that runs in the background to perform long-running operations or to perform work for remote processes. A service does not provide a user interface. For example, a service might play music in the background.

content providers
A content provider manages a shared set of application data. You can store the data in the file system, an SQLite database, on the web, or any other persistent storage location your application can access. Through the content provider, other applications can query or even modify the data (if the content provider allows it).

broadcast receivers
A broadcast receiver is a component that responds to system-wide broadcast announcements. Many broadcasts originate from the system—for example, a broadcast announcing that the screen has turned off, the battery is low, or a picture was captured. Applications can also initiate broadcasts—for example, to let other applications know that some data has been downloaded to the device and is available for them to use.

the manifest file
Before the Android system can start an application component, the system must know that the component exists by reading the application’s AndroidManifest.xml.

Your application must declare all its components in this file.
The manifest does a number of things in addition to declaring the application’s components, such as:
- Identify any user permissions the application requires, such as Internet access or read-access to the user’s contacts.
- Declare the minimum API Level required by the application, based on which APIs the application uses.
- Declare hardware and software features used or required by the application, such as a camera, Bluetooth services, or a multitouch screen.
- APIs the application needs to be linked against (other than the Android framework APIs), such as the Google Maps library.

create android virtual devices (AVD)
AVD is an emulator instance that enables you to model an actual device.
- Consists of a hardware profile
- Mapping to a phone limitations (e.g., screen size, cellular wireless speeds)
- Emulated storage, such as Secure Digital (SD) card, etc.

You can create as many AVDs as you like Use different AVDs to test your applications under different scenarios

application resources
An Android application is composed of more than just code—it requires resources that are separate from the source code, such as images, audio files, and anything relating to the visual presentation of the application.

You should define animations, menus, styles, colors, and the layout of activity user interfaces with XML files.
Using application resources makes it easy to update various characteristics of your application without modifying code and—by providing sets of alternative resources—enables you to optimize your application for a variety of device configurations (such as different languages and screen sizes).

For every resource that you include in your Android project, the SDK build tools define a unique integer ID, which you can use to reference the resource from your application code or from other resources defined in XML. For example, if your application contains an image file named logo.png (saved in the res/drawable/ directory), the SDK tools generate a resource ID named R.drawable.logo, which you can use to reference the image and insert it in your user interface.
course

papers + coding

MyRuns app
Lab 0 - Set up environment do Hello World, ++
Lab 1 - Construct the UI
Lab 2 - Database (SQLite) and adapters
Lab 3 - Maps, location (GPS) and services
Lab 4 - Motion sensors and exercising stats
Lab 5 - Cloud side using App Engine

group projects
collaborative working
no quizzes, just code
cool apps, jokes .. yes.

and most importantly

HAVE FUN!