Market Background
Mobile Phone Market

1.18 Billion handsets sold in 2008

90% of world population has cell coverage
The players

All Handsets sold
Q3 2008

<table>
<thead>
<tr>
<th>Company</th>
<th>Percent of market</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nokia</td>
<td>39.4%</td>
</tr>
<tr>
<td>Samsung</td>
<td>17.3%</td>
</tr>
<tr>
<td>Sony Ericsson</td>
<td>8.6%</td>
</tr>
<tr>
<td>Motorola</td>
<td>8.5%</td>
</tr>
<tr>
<td>LG Electronics</td>
<td>7.7%</td>
</tr>
<tr>
<td>All others</td>
<td>18.5%</td>
</tr>
</tbody>
</table>

What does market share mean?

Number of handsets sold in a quarter

Not number of handsets in use
Smart Phones

"mobile phone offering advanced capabilities beyond a typical mobile phone, often with PC-like functionality"

Wikipedia

<table>
<thead>
<tr>
<th>Year</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>1992</td>
<td>Simon (IBM)</td>
</tr>
</tbody>
</table>
| 1996 | Nokia 9000  
Nokia Communicator series |
| 2001 | BlackBerry (RIM)  
BREW (Qualcomm) |
| 2002 | Windows Mobile |
| 2007 | iPhone |
| 2008 | Android |
| 2009 | Palm Pre, Nokia N900  
Windows Mobile 6.5 |
| 2010 | Windows Mobile 7  
Zune Phone???
## Smart Phone Global Market share

<table>
<thead>
<tr>
<th>Vendor/OS</th>
<th>Q2'08</th>
<th>Q2'09</th>
<th>Growth</th>
</tr>
</thead>
<tbody>
<tr>
<td>Symbian</td>
<td>58.2%</td>
<td>50.3%</td>
<td>-2.1%</td>
</tr>
<tr>
<td>RIM</td>
<td>16.7%</td>
<td>20.9%</td>
<td>41.6%</td>
</tr>
<tr>
<td>Apple</td>
<td>2.1%</td>
<td>13.7%</td>
<td>629.9%</td>
</tr>
<tr>
<td>Microsoft</td>
<td>14.3%</td>
<td>9.0%</td>
<td>-28.7%</td>
</tr>
<tr>
<td>Android</td>
<td>-</td>
<td>2.8%</td>
<td>NA</td>
</tr>
<tr>
<td>Others</td>
<td>8.6%</td>
<td>3.3%</td>
<td>-56.8%</td>
</tr>
</tbody>
</table>

## Estimated Operating Profits
### First Half 2009

<table>
<thead>
<tr>
<th>Company</th>
<th>Revenue</th>
<th>Op, Profit</th>
<th>Op. Margin</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nokia</td>
<td>17,014</td>
<td>1,926</td>
<td>11.3%</td>
</tr>
<tr>
<td>Samsung</td>
<td>12,223</td>
<td>1,283</td>
<td>10.5%</td>
</tr>
<tr>
<td>RIM</td>
<td>6,887</td>
<td>1,423</td>
<td>20.7%</td>
</tr>
<tr>
<td>LG</td>
<td>6,514</td>
<td>593</td>
<td>9.1%</td>
</tr>
<tr>
<td>Apple</td>
<td>5,094</td>
<td>2,038</td>
<td>40%</td>
</tr>
<tr>
<td>Sony Ericsson</td>
<td>4,561</td>
<td>-841</td>
<td>-18.5%</td>
</tr>
<tr>
<td>Motorola</td>
<td>3,630</td>
<td>-762</td>
<td>-21.0%</td>
</tr>
</tbody>
</table>

---

HTML 5 & Mobile Phones
Google IO Demo

http://www.youtube.com/watch?v=S5aJAAgZIvk
Some HTML 5 Features

Canvas - 2D drawing

Audio/Video playback

Offline storage (client-side database)

Document editing

Drag & Drop
Browsers supporting HTML 5

Chrome
Firefox 3.5
Safari 4
Opera
Why is HTML 5 important?

Web applications that can work offline
Palm Pre & Web OS

Applications are developed using

Html 5
Javascript
CSS
PhoneGap

http://phonegap.com/

Develop application using
  HTML
  CSS
  Javascript

Native Application run on
  iPhone
  Android
  Blackberry
JavaGround

http://www.javaground.com/

Application (Games) developed in Java

Applications run on
  J2ME phones
  Brew
  iPhone
  Android
  Windows Mobile
Web Apps

Build Web apps that look native using
  Html 4/5
  Javascript
  CSS

Can be stored locally

Can store data locally

Cheaper to build than applications

Can be modified for multiple phones
Ways to develop for Smart Mobile Phones

Native SDK for the phone

Cross platform systems

Web Applications
Android
Android

Googles mobile phone OS and SDK

Java only
   Special VM
   Nonstandard byte code

Eclipse is development IDE

Linux

Application framework
2D & 3D graphics
Audio, video and still image support
SQLite database
Embeddable web browser

Hardware dependent

GSM
Bluetooth, EDGE, 3G, WIFI
Camera, GPS, compass
accelerometer
Android SDK


See Getting Started at Android Docs

Current version 1.5r3

Supported OS
  Windows XP, Vista
  Mac OS X 10.4.8 or later (intel processor only)
  Linux (Tested on Ubuntu Dapper Drake)

IDE
  Eclipse 3.3 or 3.4
  Java JDK 5 or JDK 6
Design Issues for Mobile Phone Apps

Screen Size
User input
Memory constraints
Limited CPU
Battery life
Security
Why Android

Why did Google create Android
  Google search, maps, talk part of Android

Why study Android in this course
  New generation of mobile app development
  Google App store http://www.android.com/market/

Archos App Store
  http://appslib.com/developers/index.html
**Emulators**

Very useful in developing applications

Not the same as running on real device
   Emulator has bugs
   Device has different bugs
   Device has restriction and limitations
   Device as resources not on your development machine

Eclipse starts emulator when run Android app
   Can recompile and run app without exiting and restarting emulator
Hello World Example

Download and install Android
http://developer.android.com/sdk/1.5_r3/index.html

Follow Hello World Tutorial

Hello World
Following "Hello Android" section of "Getting Started"

Auto generated parts of application

HelloAndroid.java
Source code

R.java
Provides access to resources

Resources
icon.png (Application icon)
main.xml (Optional Layout of application view)
strings.xml (Allows separation of source code and display text)

AndroidManifest.xml
Describes application contents
package sdsu.cs696;

import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;

public class HelloAndroid extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        TextView tv = new TextView(this);
        tv.setText("Hello, Android");
        setContentView(tv);
    }
}

Bold text indicates text added or modified from auto-generated code
Println does not work

```java
package sdsu.cs696;

import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;

public class HelloAndroid extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        TextView tv = new TextView(this);
        tv.setText("Hello, Android");
        setContentView(tv);
        System.out.println("Debug here");
    }
}
```
Basic Android Application Parts

Activities
- UI building block
- Views & Activity subclasses

Content Providers
- Shares data between applications

Intents
- System messages

Services
- Long-running nonGUI code
Things your program can use

Data Storage
  SQL database

Network Access
  Raw sockets
  Embeddable Web browser

Multimedia
  Sound
  Video

GPS
  Location

Phone services
Views

View
Displays content in rectangular area of screen
Handles
  Layout, focus, scrolling
  Keyboard events
  Gestures

ViewGroups
Manages set of views and view groups
Composite pattern
Some Views

AutoCompleteTextView MultiAutoCompleteTextView
Button RadioButton
CheckBox RatingBar
CheckedTextView ScrollView
Chronometer SeekBar
DatePicker Spinner
DigitalClock TabHost
EditText TabWidget
ExpandableListView TableRow
Gallery TimePicker
GridView ToggleButton
ImageButton TwoLineListItem
ListView VideoView
MapView ViewAnimator
MapView WebView
MapView ZoomButton
MapView ZoomControls
Activity

Single, focused thing that a user can do

Usually each screen has its own activity

An application may have multiple screens, hence multiple activities

An application runs in its own Linux process
Activity Lifecycle

Active
    Running activity in foreground of screen

Paused
    Lost focus, but still visible
    Retains all state information
    In extreme memory situations may be killed

Stopped
    Not visible
    Retains all state information
    Often will be killed

Killed
Activity Example

package edu.sdsu.cs683;

import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;

public class CountStates extends Activity {
    int paused = 0;
    int killed = 0;
    int stopped = 0;
    TextView text;
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    if (savedInstanceState != null) {
        paused = savedInstanceState.getInt("paused");
        killed = savedInstanceState.getInt("killed");
        stopped = savedInstanceState.getInt("stopped");
    }
    text = new TextView(this);
    text.setText("Paused: " + paused + " stopped: " + stopped + " killed "
                 + killed);
    setContentView(text);
}
protected void onResume() {
    super.onResume();
    text.setText("Paused: " + paused + " stopped: " + stopped + " killed " + killed);
}

protected void onStart() {
    super.onStart();
    text.setText("Paused: " + paused + " stopped: " + stopped + " killed " + killed);
}

protected void onStop() {
    stopped++;
    super.onStop();
}
protected void onPause() {
    paused++;
    super.onPause();
}

protected void onDestroy() {
    killed++;
    super.onDestroy();
}

protected void onSaveInstanceState(Bundle outState) {
    outState.putInt("paused", paused);
    outState.putInt("killed", killed);
    outState.putInt("stopped", stopped);
}