

CS 696 Mobile Phone Application Development
Fall Semester, 2009
Doc 1 Mobile & Android Introduction
Sept 3, 2009

Copyright ©, All rights reserved. 2009 SDSU & Roger Whitney, 5500 Campanile Drive, San Diego, CA 92182-7700 USA. OpenContent (<http://www.opencontent.org/opl.shtml>) license defines the copyright on this document.

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

Company	Percent of market
Nokia	39.4%
Samsung	17.3%
Sony Ericsson	8.6%
Motorola	8.5%
LG Electronics	7.7%
All others	18.5%

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

1992	Simon (IBM)
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

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

Estimated Operating Profits

First Half 2009

	Revenue	Op, Profit	Op. Margin
Nokia	17,014	1,926	11.3%
Samsung	12,223	1,283	10.5%
RIM	6,887	1423	20.7%
LG	6,514	593	9.1%
Apple	5,094	2,038	40%
Sony Ericsson	4,561	-841	-18.5%
Motorola	3,630	-762	-21.0%

HTML 5 & Mobile Phones

Google IO Demo

<http://www.youtube.com/watch?v=S5aJAaGZlvk>

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

<http://developer.android.com/guide/index.html>

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

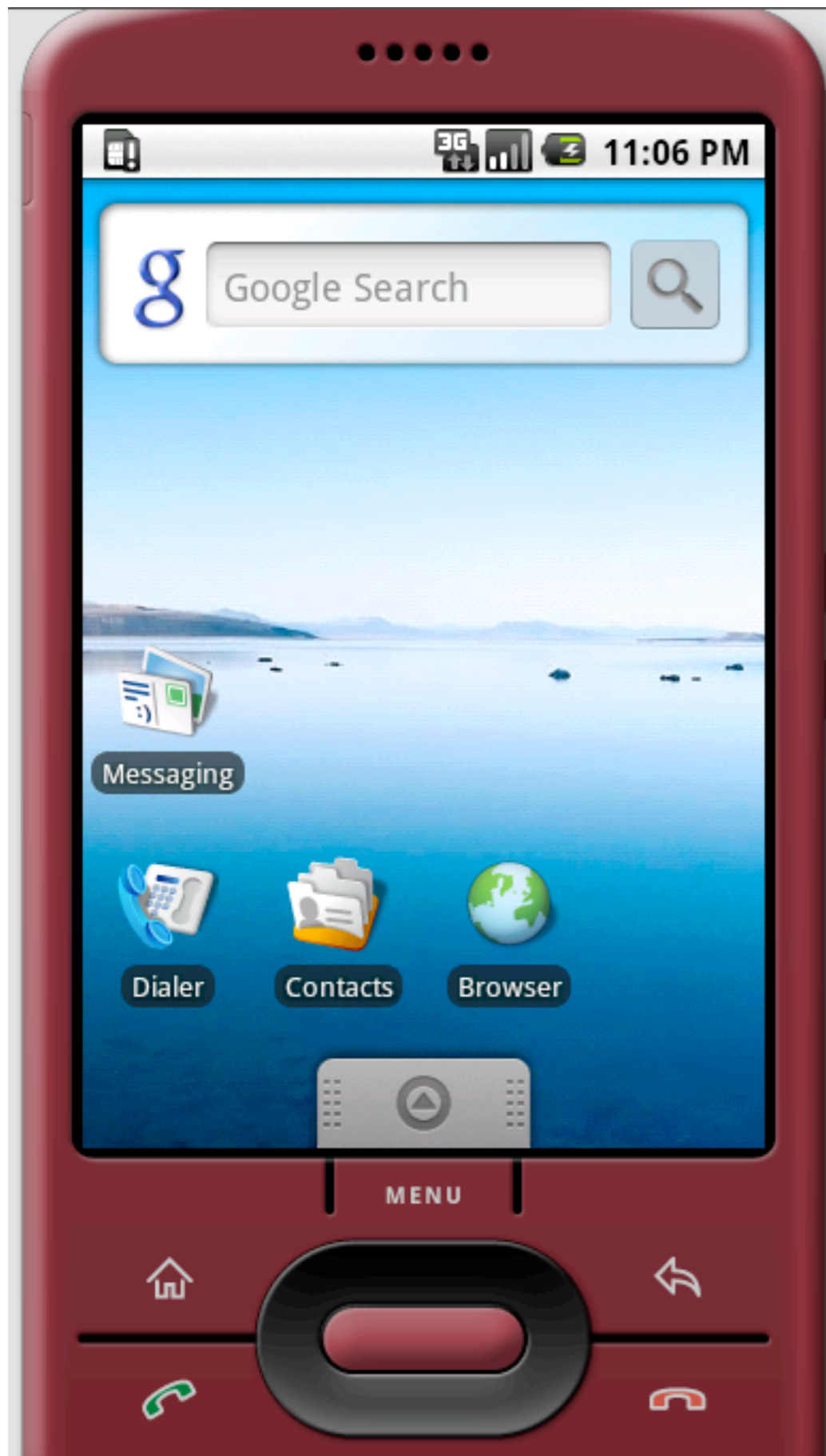
Why study Android in this course

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

<http://developer.android.com/guide/tutorials/hello-world.html>

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

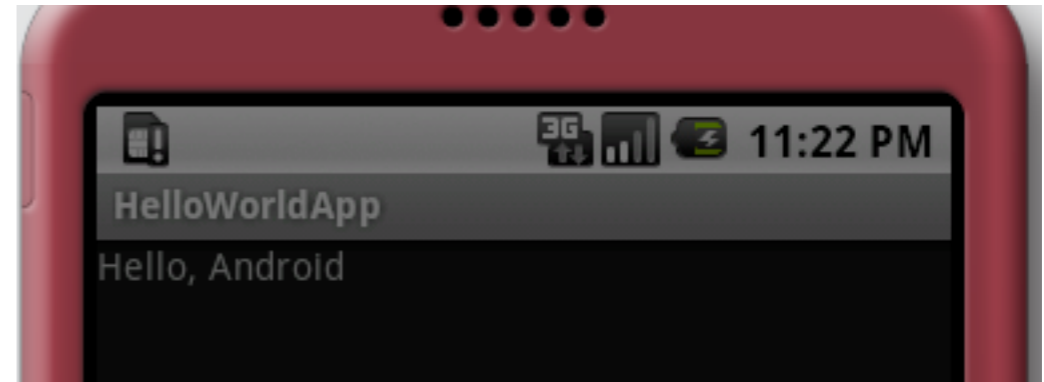
Describes application contents

Hello.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);
    }
}
```



Println does not work

```
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

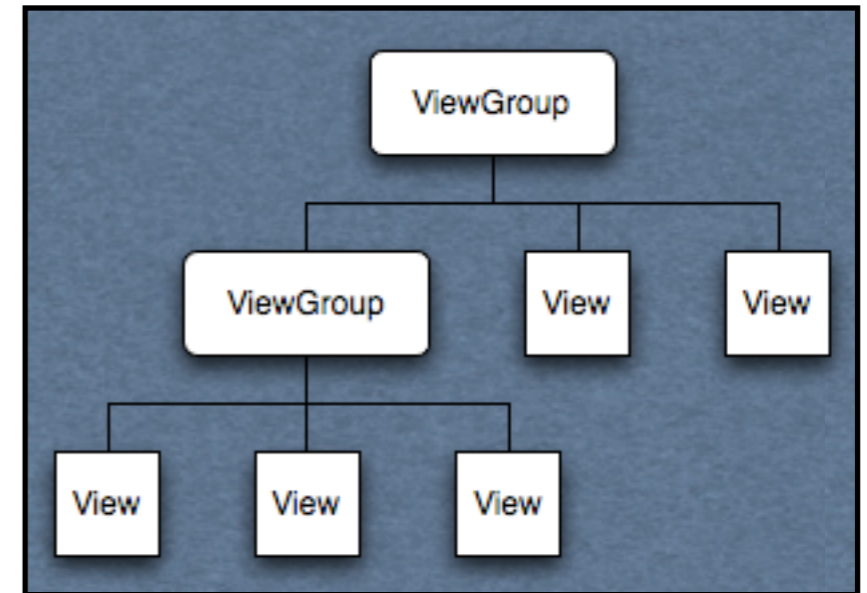
Keyboard events

Gestures

ViewGroups

Manages set of views and view groups

Composite pattern



Some Views

AutoCompleteTextView

Button

CheckBox

CheckedTextView

Chronometer

DatePicker

DigitalClock

EditText

ExpandableListView

Gallery

GridView

ImageButton

ListView

MapView,

MultiAutoCompleteTextView

RadioButton

RatingBar

ScrollView

SeekBar

Spinner

TabHost

TabWidget

TableRow

TimePicker

ToggleButton

TwoLineListItem

VideoView

ViewAnimator

WebView

ZoomButton

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

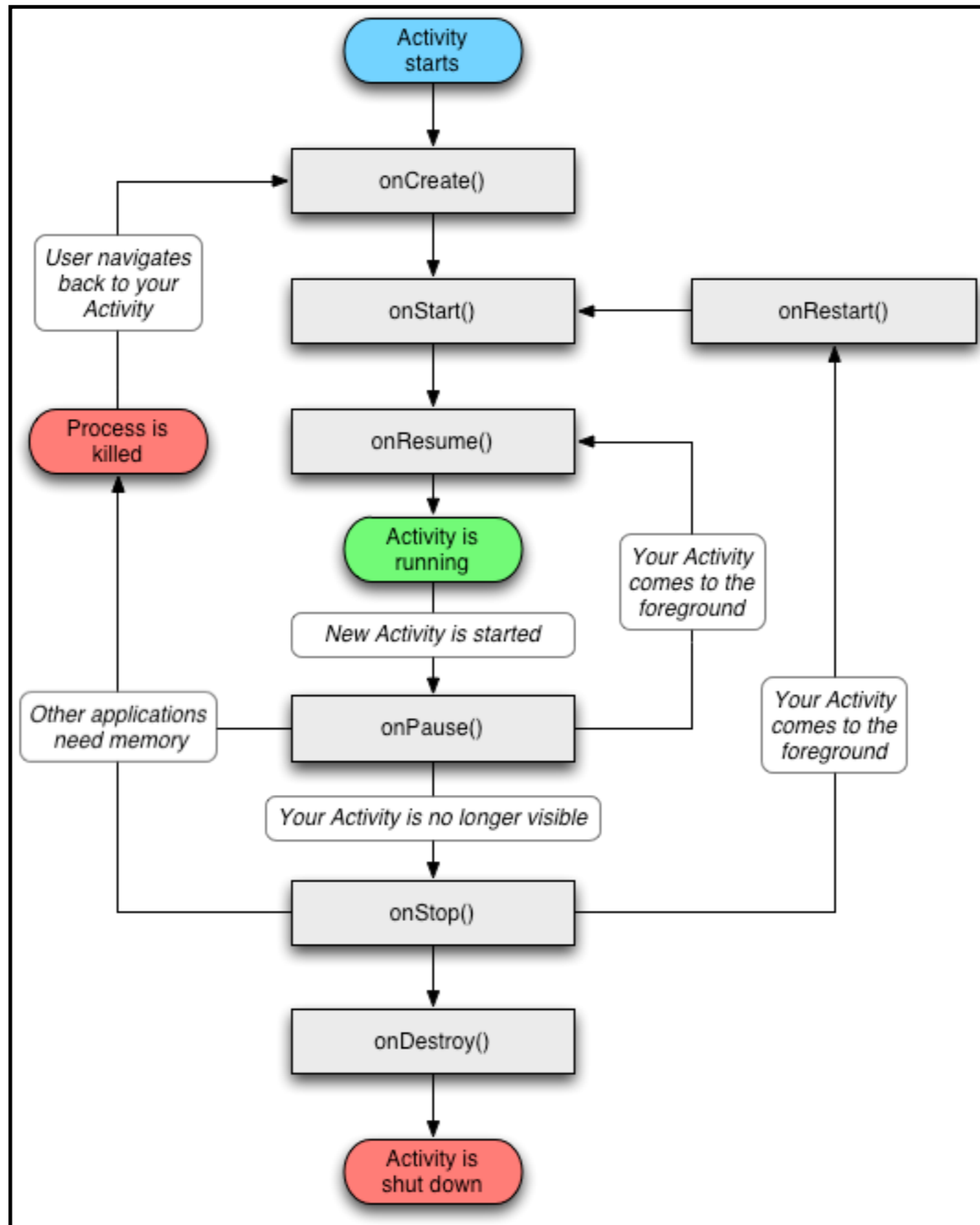
Stopped

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;
```

Activity Example

```
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);  
}
```

Activity Example

```
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();  
}
```

Activity Example

```
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);  
}
```

```
}
```