

Introduction: MOOC Organization & Topics

Douglas C. Schmidt

d.schmidt@vanderbilt.edu

www.dre.vanderbilt.edu/~schmidt



Professor of Computer Science

Institute for Software
Integrated Systems

Vanderbilt University
Nashville, Tennessee, USA

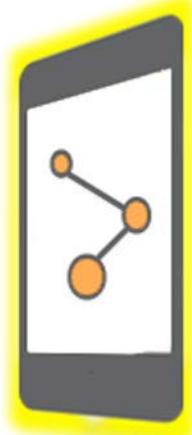


Learning Objectives in this Part of the Module

- Understand the MOOC's structure & contents



mobile
apps



mobile
services



cloud
services



Learning Objectives in this Part of the Module

- Understand the MOOC's structure & contents

Section 0: MOOC Introduction

- Part 1: MOOC Organization & Topics
- Part 2: MOOC Prereqs, Workload, & Learning Strategies
- Part 3&4: Concurrency Motivations & Challenges
- Part 5&6: Overview of Patterns and Frameworks

Section 1: Overview of Android & Java Concurrency

- Part 1: Android Layers
- Part 2: Java Threading Mechanisms
- Part 3: Java Built-in Synchronization Mechanisms

Section 2: Android Concurrency Frameworks

- Part 1&2: Concurrency Framework Classes & Patterns
- Part 3: The Threaded Downloads Application
- Part 4: Android Looper
- Part 5: Android Handler & HaMeR Framework
- Part 6: Posting/Processing Runnables with the HaMeR Framework
- Part 7: Sending/Handling Messages with the HaMeR Framework
- Part 8: The AsyncTask Framework
- Part 9: Blackbox & Whitebox Frameworks with AsyncTask
- Part 10: Evaluating Android Concurrency Frameworks

Section 3: Concurrency Patterns in Android

- Part 1: The Thread-Specific Storage Pattern
- Part 2: The Command Processor Pattern
- Part 3: The Active Object Pattern
- Part 4: The Half-Sync/Half-Async Pattern



Learning Objectives in this Part of the Module

- Understand the MOOC's structure & contents

Section 0: MOOC Introduction

Part 1: MOOC Organization & Topics

Part 2: MOOC Prereqs, Workload, & Learning Strategies

Part 3&4: Concurrency Motivations & Challenges

Part 5&6: Overview of Patterns and Frameworks

Section 1: Overview of Android & Java Concurrency

Part 1: Android Layers

Part 2: Java Threading Mechanisms

Part 3: Java Built-in Synchronization Mechanisms

Section 2: Android Concurrency Frameworks

Part 1&2: Concurrency Framework Classes & Patterns

Part 3: The Threaded Downloads Application

Part 4: Android Looper

Part 5: Android Handler & HaMeR Framework

Part 6: Posting/Processing Runnables with the HaMeR Framework

Part 7: Sending/Handling Messages with the HaMeR Framework

Part 8: The AsyncTask Framework

Part 9: Blackbox & Whitebox Frameworks with AsyncTask

Part 10: Evaluating Android Concurrency Frameworks

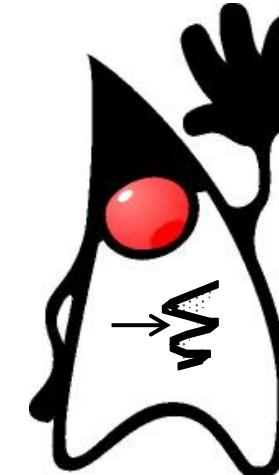
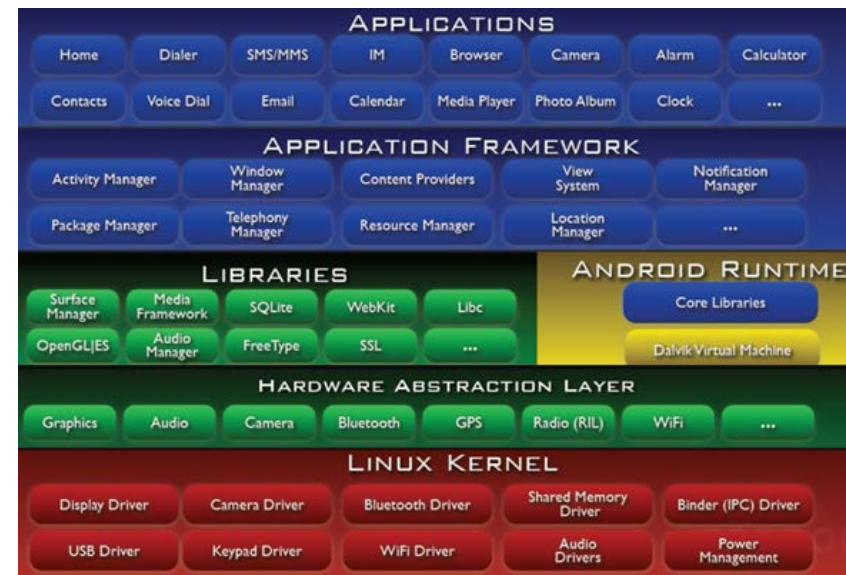
Section 3: Concurrency Patterns in Android

Part 1: The Thread-Specific Storage Pattern

Part 2: The Command Processor Pattern

Part 3: The Active Object Pattern

Part 4: The Half-Sync/Half-Async Pattern



Learning Objectives in this Part of the Module

- Understand the MOOC's structure & contents

Section 0: MOOC Introduction

Part 1: MOOC Organization & Topics

Part 2: MOOC Prereqs, Workload, & Learning Strategies

Part 3&4: Concurrency Motivations & Challenges

Part 5&6: Overview of Patterns and Frameworks

Section 1: Overview of Android & Java Concurrency

Part 1: Android Layers

Part 2: Java Threading Mechanisms

Part 3: Java Built-in Synchronization Mechanisms

Section 2: Android Concurrency Frameworks

Part 1&2: Concurrency Framework Classes & Patterns

Part 3: The Threaded Downloads Application

Part 4: Android Looper

Part 5: Android Handler & HaMeR Framework

Part 6: Posting/Processing Runnables with the HaMeR Framework

Part 7: Sending/Handling Messages with the HaMeR Framework

Part 8: The AsyncTask Framework

Part 9: Blackbox & Whitebox Frameworks with AsyncTask

Part 10: Evaluating Android Concurrency Frameworks

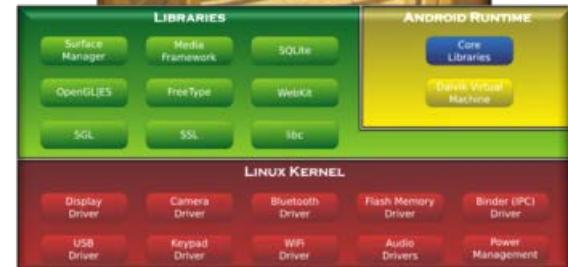
Section 3: Concurrency Patterns in Android

Part 1: The Thread-Specific Storage Pattern

Part 2: The Command Processor Pattern

Part 3: The Active Object Pattern

Part 4: The Half-Sync/Half-Async Pattern



Learning Objectives in this Part of the Module

- Understand the MOOC's structure & contents

Section 0: MOOC Introduction

Part 1: MOOC Organization & Topics

Part 2: MOOC Prereqs, Workload, & Learning Strategies

Part 3&4: Concurrency Motivations & Challenges

Part 5&6: Overview of Patterns and Frameworks

Section 1: Overview of Android & Java Concurrency

Part 1: Android Layers

Part 2: Java Threading Mechanisms

Part 3: Java Built-in Synchronization Mechanisms

Section 2: Android Concurrency Frameworks

Part 1&2: Concurrency Framework Classes & Patterns

Part 3: The Threaded Downloads Application

Part 4: Android Looper

Part 5: Android Handler & HaMeR Framework

Part 6: Posting/Processing Runnables with the HaMeR Framework

Part 7: Sending/Handling Messages with the HaMeR Framework

Part 8: The AsyncTask Framework

Part 9: Blackbox & Whitebox Frameworks with AsyncTask

Part 10: Evaluating Android Concurrency Frameworks

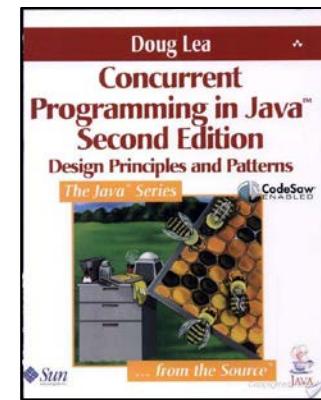
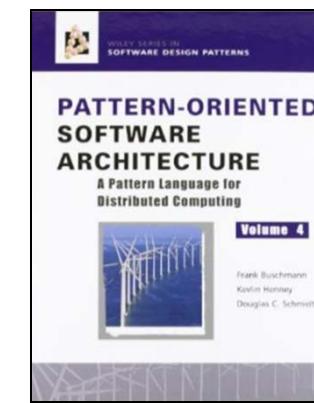
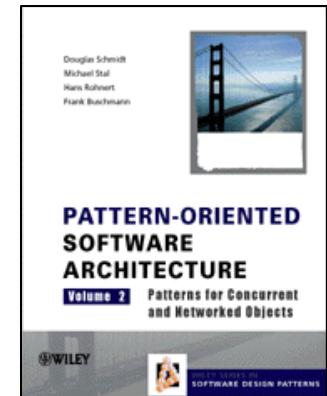
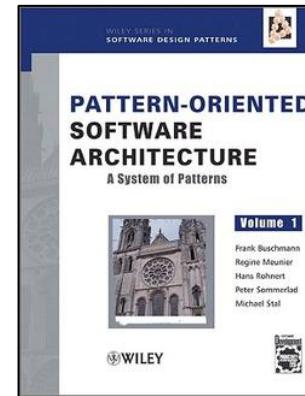
Section 3: Concurrency Patterns in Android

Part 1: The Thread-Specific Storage Pattern

Part 2: The Command Processor Pattern

Part 3: The Active Object Pattern

Part 4: The Half-Sync/Half-Async Pattern



Learning Objectives in this Part of the Module

- Understand the MOOC's structure & contents

Section 0: MOOC Introduction

Part 1: MOOC Organization & Topics

Part 2: MOOC Prereqs, Workload, & Learning Strategies

Part 3&4: Concurrency Motivations & Challenges

Part 5&6: Overview of Patterns and Frameworks

Section 1: Overview of Android & Java Concurrency

Part 1: Android Layers

Part 2: Java Threading Mechanisms

Part 3: Java Built-in Synchronization Mechanisms

Section 2: Android Concurrency Frameworks

Part 1&2: Concurrency Framework Classes & Patterns

Part 3: The Threaded Downloads Application

Part 4: Android Looper

Part 5: Android Handler & HaMeR Framework

Part 6: Posting/Processing Runnables with the HaMeR Framework

Part 7: Sending/Handling Messages with the HaMeR Framework

Part 8: The AsyncTask Framework

Part 9: Blackbox & Whitebox Frameworks with AsyncTask

Part 10: Evaluating Android Concurrency Frameworks

Section 3: Concurrency Patterns in Android

Part 1: The Thread-Specific Storage Pattern

Part 2: The Command Processor Pattern

Part 3: The Active Object Pattern

Part 4: The Half-Sync/Half-Async Pattern



See [github.com/douglasraigschmidt/
POSA-15/wiki/POSA-15-FAQ](https://github.com/douglasraigschmidt/POSA-15/wiki/POSA-15-FAQ) item #26

Overview of the MOOC Topics in Section 0

Overview of Topics Covered in Section 0

Section 0: MOOC Introduction

- Part 1: MOOC Organization & Topics
- Part 2: MOOC Prereqs, Workload, & Learning Strategies
- Part 3&4: Concurrency Motivations & Challenges
- Part 5&6: Overview of Patterns and Frameworks

Section 1: Overview of Android & Java Concurrency

- Part 1: Android Layers
- Part 2: Java Threading Mechanisms
- Part 3: Java Built-in Synchronization Mechanisms

Section 2: Android Concurrency Frameworks

- Part 1&2: Concurrency Framework Classes & Patterns
- Part 3: The Threaded Downloads Application
- Part 4: Android Looper
- Part 5: Android Handler & HaMeR Framework
- Part 6: Posting/Processing Runnables with the HaMeR Framework
- Part 7: Sending/Handling Messages with the HaMeR Framework
- Part 8: The AsyncTask Framework
- Part 9: Blackbox & Whitebox Frameworks with AsyncTask
- Part 10: Evaluating Android Concurrency Frameworks

Section 3: Concurrency Patterns in Android

- Part 1: The Thread-Specific Storage Pattern
- Part 2: The Command Processor Pattern
- Part 3: The Active Object Pattern
- Part 4: The Half-Sync/Half-Async Pattern



Overview of Topics Covered in Section 0

Section 0: MOOC Introduction

- Part 1: MOOC Organization & Topics
- Part 2: MOOC Prereqs, Workload, & Learning Strategies
- Part 3&4: Concurrency Motivations & Challenges
- Part 5&6: Overview of Patterns and Frameworks

Section 1: Overview of Android & Java Concurrency

- Part 1: Android Layers
- Part 2: Java Threading Mechanisms
- Part 3: Java Built-in Synchronization Mechanisms

Section 2: Android Concurrency Frameworks

- Part 1&2: Concurrency Framework Classes & Patterns
- Part 3: The Threaded Downloads Application
- Part 4: Android Looper
- Part 5: Android Handler & HaMeR Framework
- Part 6: Posting/Processing Runnables with the HaMeR Framework
- Part 7: Sending/Handling Messages with the HaMeR Framework
- Part 8: The AsyncTask Framework
- Part 9: Blackbox & Whitebox Frameworks with AsyncTask
- Part 10: Evaluating Android Concurrency Frameworks

Section 3: Concurrency Patterns in Android

- Part 1: The Thread-Specific Storage Pattern
- Part 2: The Command Processor Pattern
- Part 3: The Active Object Pattern
- Part 4: The Half-Sync/Half-Async Pattern



Overview of Topics Covered in Section 0

- **Section 0: MOOC Introduction**
 - MOOC Organization & Topics



"It's a dangerous business going out your door. You step onto the road & if you don't keep your feet there's no knowing where you might be swept off to."

Overview of Topics Covered in Section 0

- **Section 0: MOOC Introduction**

- MOOC Organization & Topics
- MOOC Prerequisites & Learning Strategies



Overview of Topics Covered in Section 0

- **Section 0: MOOC Introduction**

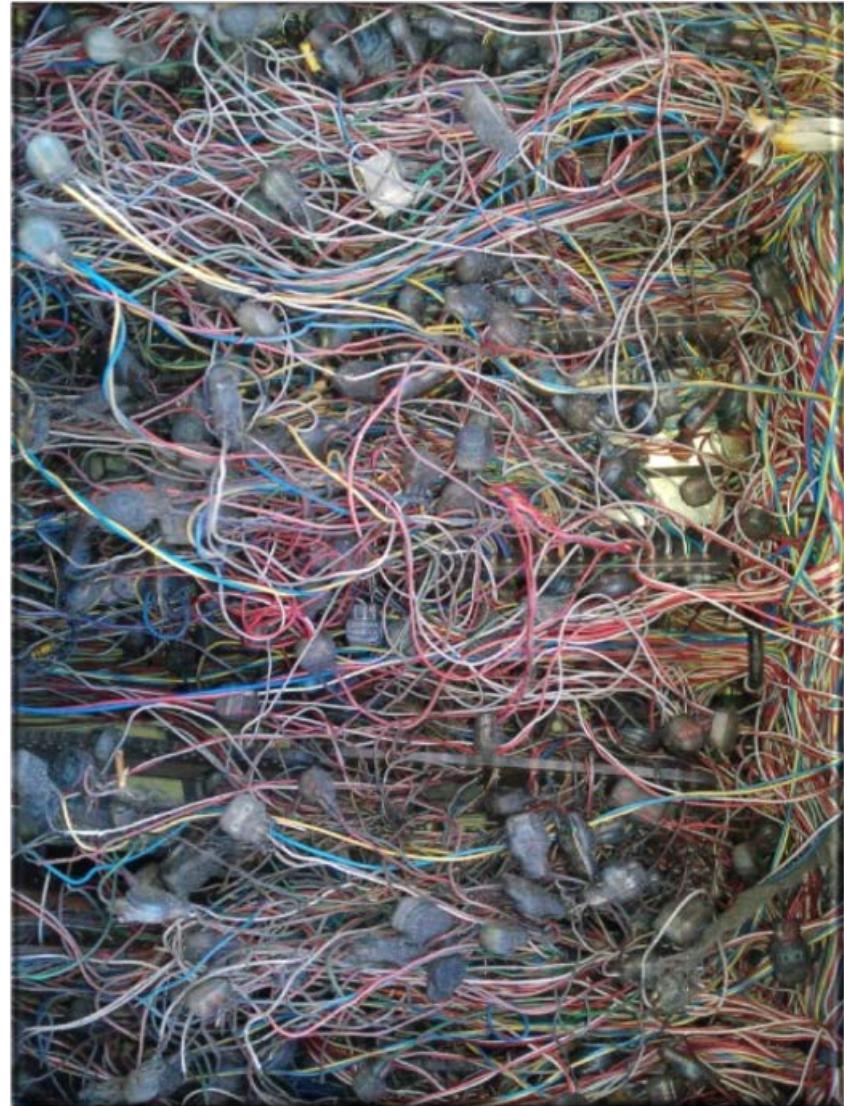
- MOOC Organization & Topics
- MOOC Prerequisites & Learning Strategies
- Concurrency Motivations & Challenges



Overview of Topics Covered in Section 0

- **Section 0: MOOC Introduction**

- MOOC Organization & Topics
- MOOC Prerequisites & Learning Strategies
- Concurrency Motivations & Challenges



Overview of Topics Covered in Section 0

- **Section 0: MOOC Introduction**

- MOOC Organization & Topics
- MOOC Prerequisites & Learning Strategies
- Concurrency Motivations & Challenges



Overview of Topics Covered in Section 0

- **Section 0: MOOC Introduction**

- MOOC Organization & Topics
- MOOC Prerequisites & Learning Strategies
- Concurrency Motivations & Challenges



Overview of Topics Covered in Section 0

- **Section 0: MOOC Introduction**

- MOOC Organization & Topics
- MOOC Prerequisites & Learning Strategies
- Concurrency Motivations & Challenges



Overview of Topics Covered in Section 0

• Section 0: MOOC Introduction

- MOOC Organization & Topics
- MOOC Prerequisites & Learning Strategies
- Concurrency Motivations & Challenges

Eclipse File Edit Run Source Navigate Project Refactor Window Help

DDMS - HelloAndroid/src/course/examples/HelloAndroid.java - Eclipse - Users/aporter/Classes/AndroidCourse/eclipse

Devices

Name	ID	Tid	Status	utime	stime	Name
com.android.process.media	812	8649	280	0	0	idle
com.android.mms	866	8650	5	167	0	HeapWorker
com.android.email	388	8651	0	0	0	GC
com.android.camera	428	8652	0	0	0	Final Catcher
com.android.gallery	459	8653	0	0	0	JNLP
course.sample	334	8646	8701	0	0	Compiler
com.android.browser	209	8655	0	0	0	Binder Thread #1
com.android.development	1007	8610	0	0	0	Binder Thread #2
com.android.term	1212	8654	0	0	0	Binder Thread #3

Emulator Control

Telephony Status

Voice: home Speed: Full

Data: home Latency: None

Telephony Actions

Incoming number:

Outgoing number:

SMS

Message:

Console LogCat

Time	pid	tag	Message
05-05 17:07:55.0 2334 dalvikvm			W/rendev: still suspended after wake (acq! doze)
05-05 17:08:23.0 2334 dalvikvm			DiskIo: 100% I/O busy freed 219K free 219K external: 714K/100K, paused 670ms
05-05 17:08:23.0 75		ActivityManager	Received connection from receiver HelloAndroid ->Hello
05-05 17:08:55.0 2334 dalvikvm			recent time failed: lava.net.SocketException: Address family not supported by protocol
05-05 17:09:11.0 2334 dalvikvm			W/rendev: still suspended after wake (acq! doze)
05-05 17:09:27.0 2334 dalvikvm			TRACE STARTED: /dev/ashmem
05-05 17:09:27.0 2334 dalvikvm			W/rendev: still suspended after wake (acq! doze)
05-05 17:09:27.0 2334 dalvikvm			TRACE STOPPED: writing 309 records
05-05 17:09:27.0 2334 dalvikvm			*** active profiler count now 1
05-05 17:09:27.0 2334 dalvikvm			The active profiler count is now 1
05-05 17:09:27.0 2334 dalvikvm			*** active profiler count now 0
05-05 17:09:27.0 2334 dalvikvm			TRACE STOPPED: writing 493 records
05-05 17:09:27.0 2334 dalvikvm			*** active profiler count now 0
05-05 17:09:27.0 2334 dalvikvm			W/rendev: 2 database(s) active, out-of-reuse results will be cleared



Overview of Topics Covered in Section 0

- **Section 0: MOOC Introduction**

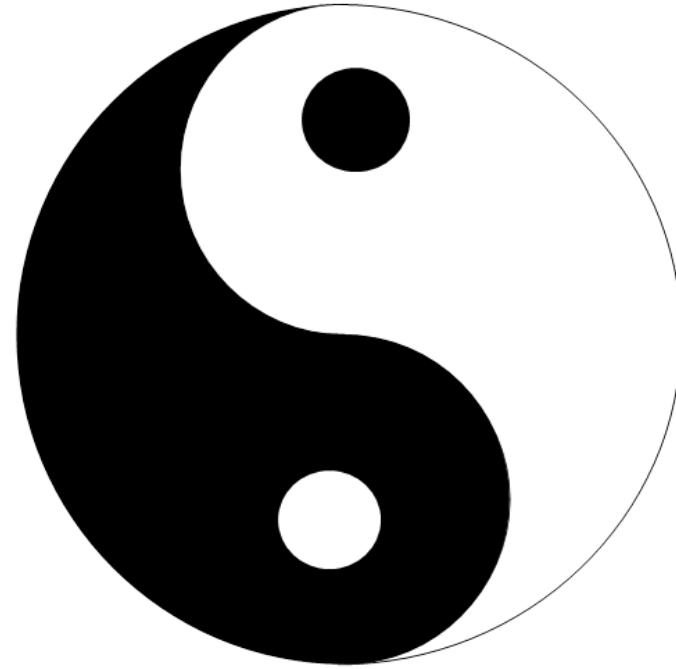
- MOOC Organization & Topics
- MOOC Prerequisites & Learning Strategies
- Concurrency Motivations & Challenges



Overview of Topics Covered in Section 0

- **Section 0: MOOC Introduction**

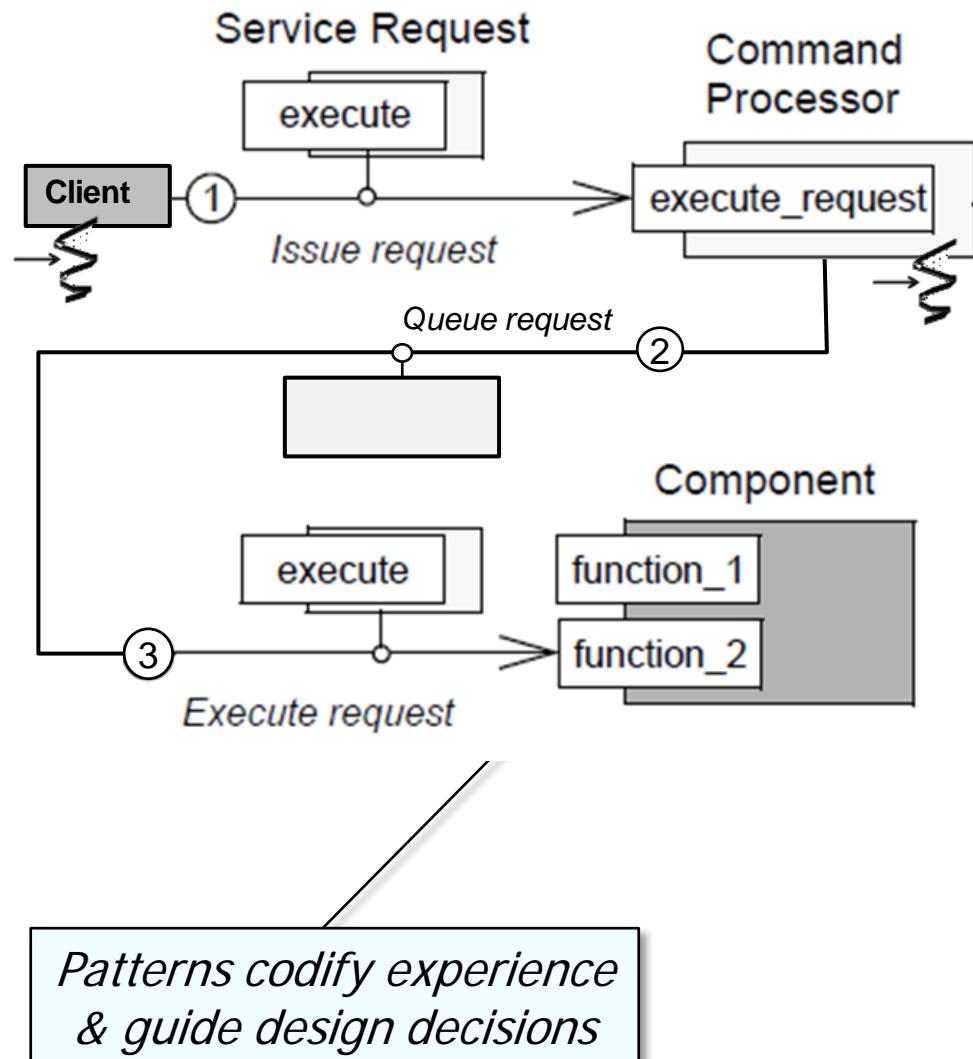
- MOOC Organization & Topics
- MOOC Prerequisites & Learning Strategies
- Concurrency Motivations & Challenges
- Overview of Patterns & Frameworks



Overview of Topics Covered in Section 0

- **Section 0: MOOC Introduction**

- MOOC Organization & Topics
- MOOC Prerequisites & Learning Strategies
- Concurrency Motivations & Challenges
- Overview of Patterns & Frameworks

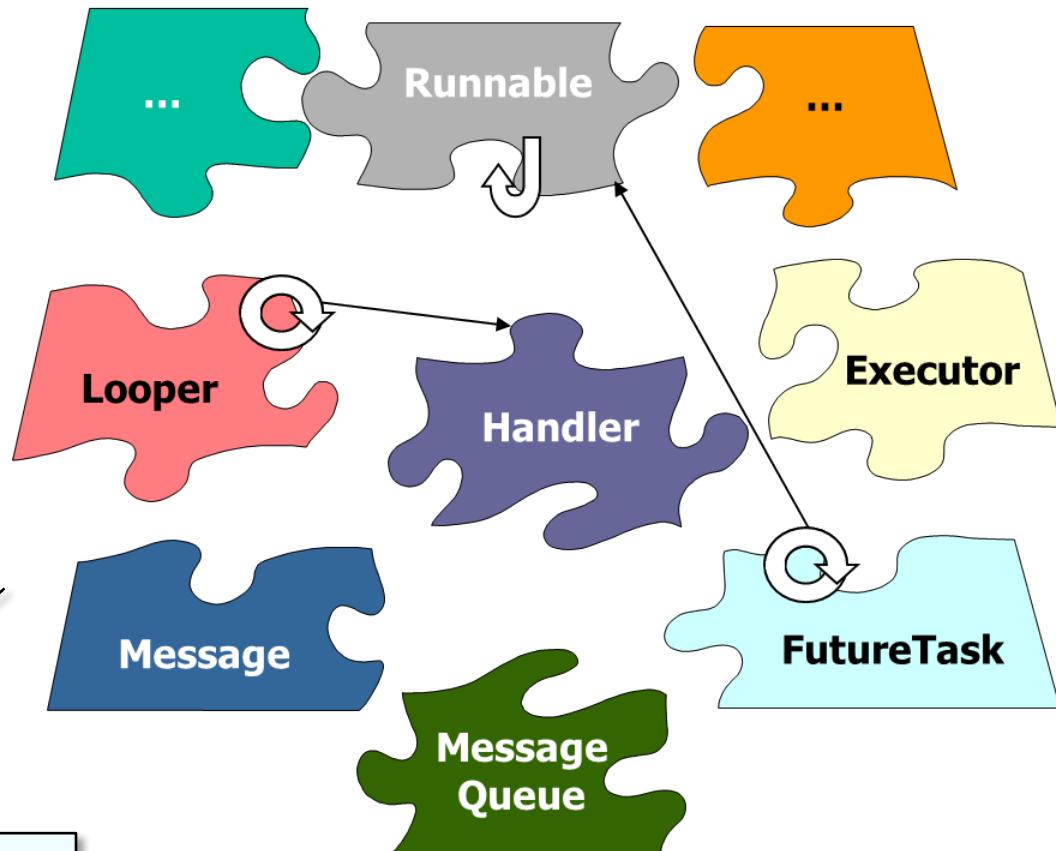


Overview of Topics Covered in Section 0

- **Section 0: MOOC Introduction**

- MOOC Organization & Topics
- MOOC Prerequisites & Learning Strategies
- Concurrency Motivations & Challenges
- Overview of Patterns & Frameworks

Application-specific functionality



Frameworks reify patterns into reusable software artifacts

Overview of Topics Covered in Section 0

- **Section 0: MOOC Introduction**

- MOOC Organization & Topics
- MOOC Prerequisites & Learning Strategies
- Concurrency Motivations & Challenges
- Overview of Patterns & Frameworks



Overview of Topics Covered in Section 0

- **Section 0: MOOC Introduction**

- MOOC Organization & Topics

- MOOC Prerequisites & Learning Strategies

- Concurrency Motivations & Challenges

- Overview of Patterns & Frameworks



Overview of the MOOC Topics in Section 1

Overview of Topics Covered in Section

Section 0: MOOC Introduction

Part 1: MOOC Organization & Topics

Part 2: MOOC Prereqs, Workload, & Learning Strategies

Part 3&4: Concurrency Motivations & Challenges

Part 5&6: Overview of Patterns and Frameworks

Section 1: Overview of Android & Java Concurrency

Part 1: Android Layers

Part 2: Java Threading Mechanisms

Part 3: Java Built-in Synchronization Mechanisms

Section 2: Android Concurrency Frameworks

Part 1&2: Concurrency Framework Classes & Patterns

Part 3: The Threaded Downloads Application

Part 4: Android Looper

Part 5: Android Handler & HaMeR Framework

Part 6: Posting/Processing Runnables with the HaMeR Framework

Part 7: Sending/Handling Messages with the HaMeR Framework

Part 8: The AsyncTask Framework

Part 9: Blackbox & Whitebox Frameworks with AsyncTask

Part 10: Evaluating Android Concurrency Frameworks

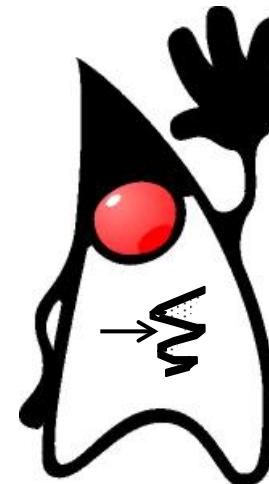
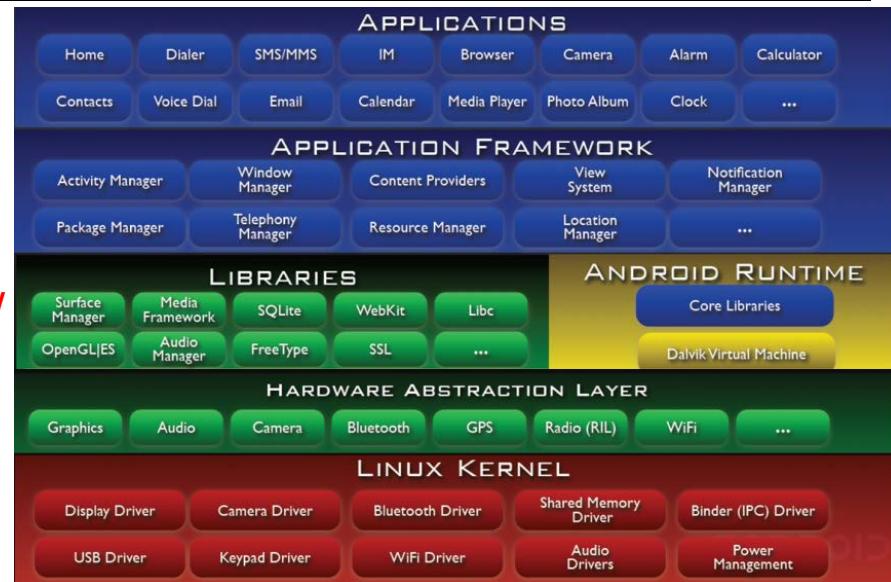
Section 3: Concurrency Patterns in Android

Part 1: The Thread-Specific Storage Pattern

Part 2: The Command Processor Pattern

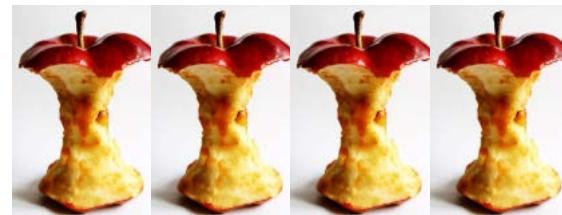
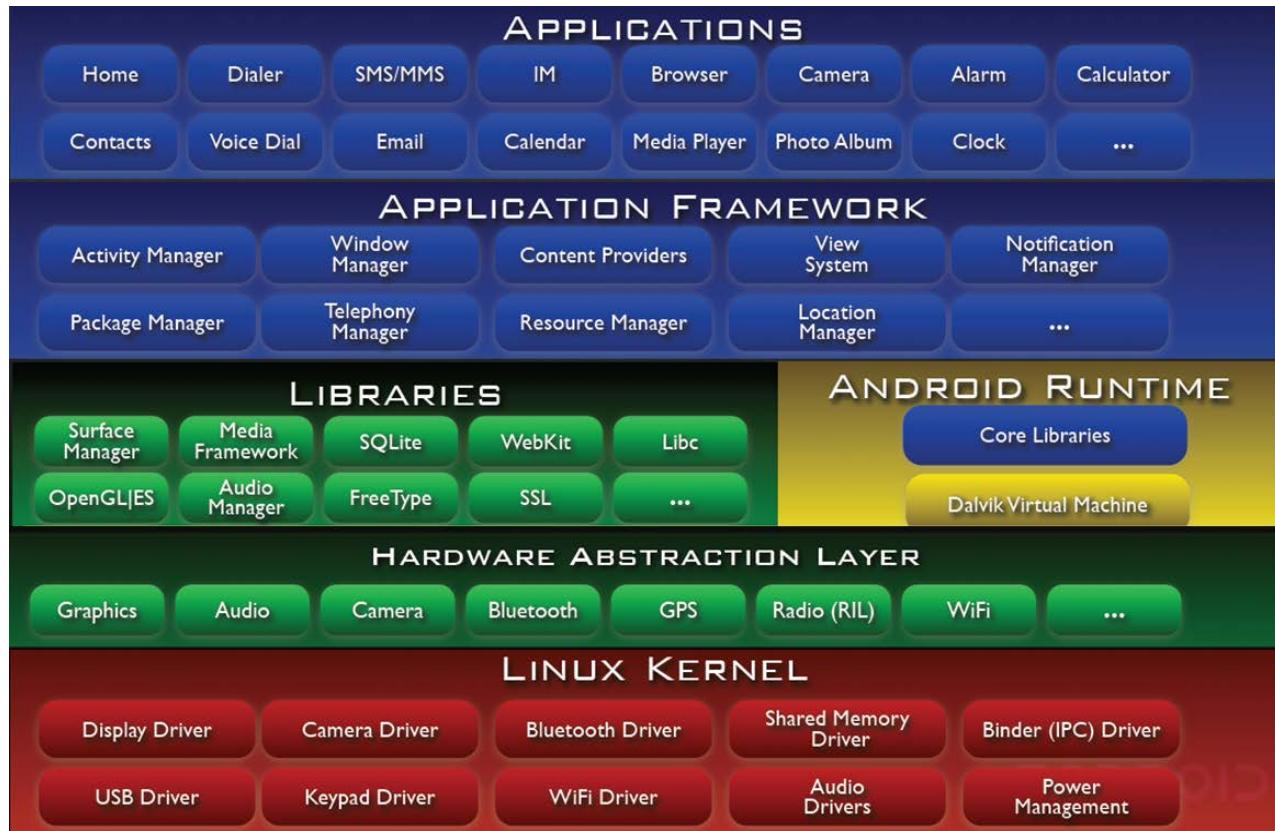
Part 3: The Active Object Pattern

Part 4: The Half-Sync/Half-Async Pattern



Overview of Topics Covered in Section 1

- **Section 1: Overview of Foundations**
 - Overview of Android Layers

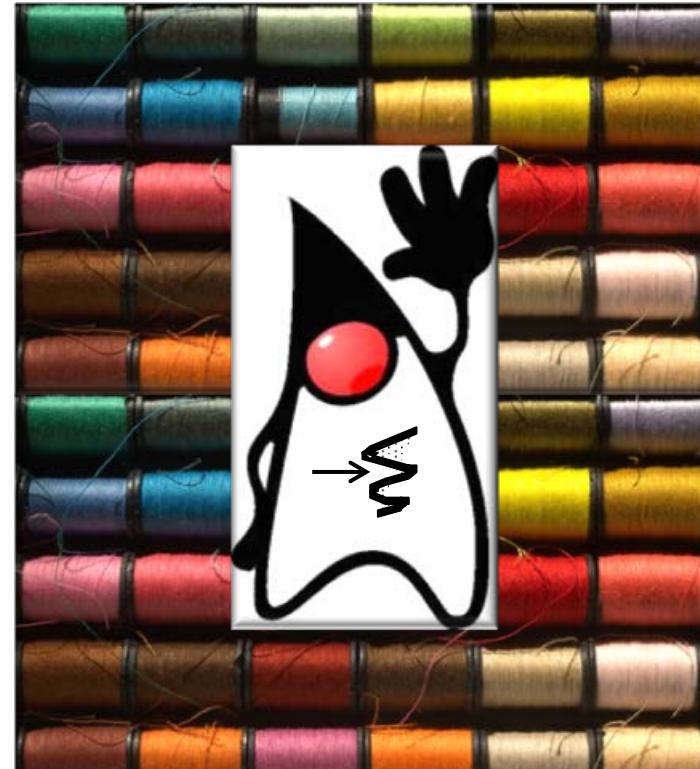


See www.laurencegellert.com/2012/08/what-is-a-full-stack-developer

Overview of Topics Covered in Section 1

- **Section 1: Overview of Foundations**

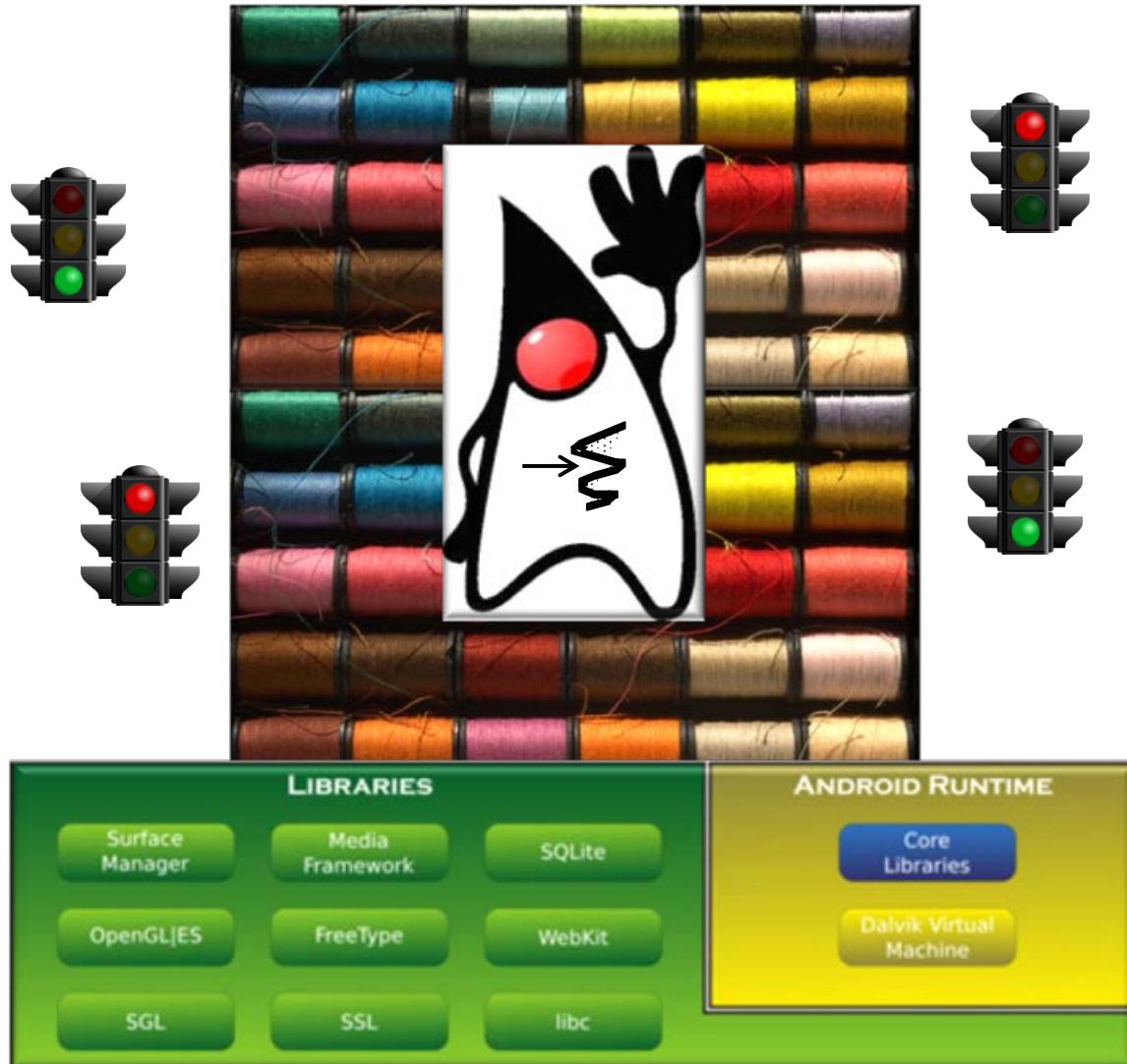
- Overview of Android Layers
- Java Threading Mechanisms



Overview of Topics Covered in Section 1

- **Section 1: Overview of Foundations**

- Overview of Android Layers
- Java Threading Mechanisms
- Java Built-in Synchronization Mechanisms



Section 1 just provides enough information on Java concurrency mechanisms to implement the programming assignments

Overview of Topics Covered in Section 1

- **Section 1: Overview of Foundations**

- Overview of Android Layers
- Java Threading Mechanisms
- Java Built-in Synchronization Mechanisms



LiveLessons: Java Concurrency

[Douglas C. Schmidt](#) (d.schmidt@vanderbilt.edu)
Associate Chair of [Computer Science and Engineering](#),
[Professor](#) of Computer Science, and Senior Researcher
in the [Institute for Software Integrated Systems](#) (ISIS)
at [Vanderbilt University](#)

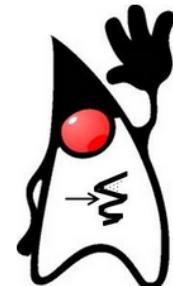


Table of Contents

- [Course Summary](#)
- [Course Topics](#)
- [Course Objectives](#)
- [Recommended Background](#)
- [Java Source Code](#)
- [Facebook Group](#)
- [Additional Resources](#)
- [About the Instructor](#)

Course Summary

There's growing demand for software developers who understand how to write concurrent programs for a range of computing platforms, including mobile devices, laptops, desktops, servers, and cloud computing environments. This demand is driven by advances in infrastructure technologies, such as high performance multi-core and distributed core processors, inexpensive mass storage, ubiquitous network connectivity, and commodity hardware and software platforms. Leveraging these advances effectively requires application developers with a robust knowledge of concurrency in order to (1) increase performance by overlapping communication and computation to run in parallel, (2) improve responsiveness by processing user interface operations in a different execution context than other background processing operations, and (3) simplify programming structure by avoiding complex and tangled purely event-driven software architectures. To help you program quality concurrent programs in Java, this course focuses on the pattern-oriented techniques, tools, and methods covered in the following five lessons:

See github.com/douglascraigschmidt/POSA-15/wiki/POSA-15-FAQ item #36

Overview of the MOOC Topics in Section 2

Overview of Topics Covered in Section 2

Section 0: MOOC Introduction

- Part 1: MOOC Organization & Topics
- Part 2: MOOC Prereqs, Workload, & Learning Strategies
- Part 3&4: Concurrency Motivations & Challenges
- Part 5&6: Overview of Patterns and Frameworks

Section 1: Overview of Android & Java Concurrency

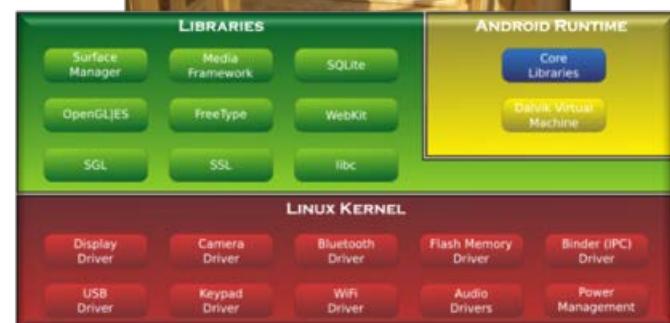
- Part 1: Android Layers
- Part 2: Java Threading Mechanisms
- Part 3: Java Built-in Synchronization Mechanisms

Section 2: Android Concurrency Frameworks

- Part 1&2: Concurrency Framework Classes & Patterns
- Part 3: The Threaded Downloads Application
- Part 4: Android Looper
- Part 5: Android Handler & HaMeR Framework
- Part 6: Posting/Processing Runnables with the HaMeR Framework
- Part 7: Sending/Handling Messages with the HaMeR Framework
- Part 8: The AsyncTask Framework
- Part 9: Blackbox & Whitebox Frameworks with AsyncTask
- Part 10: Evaluating Android Concurrency Frameworks

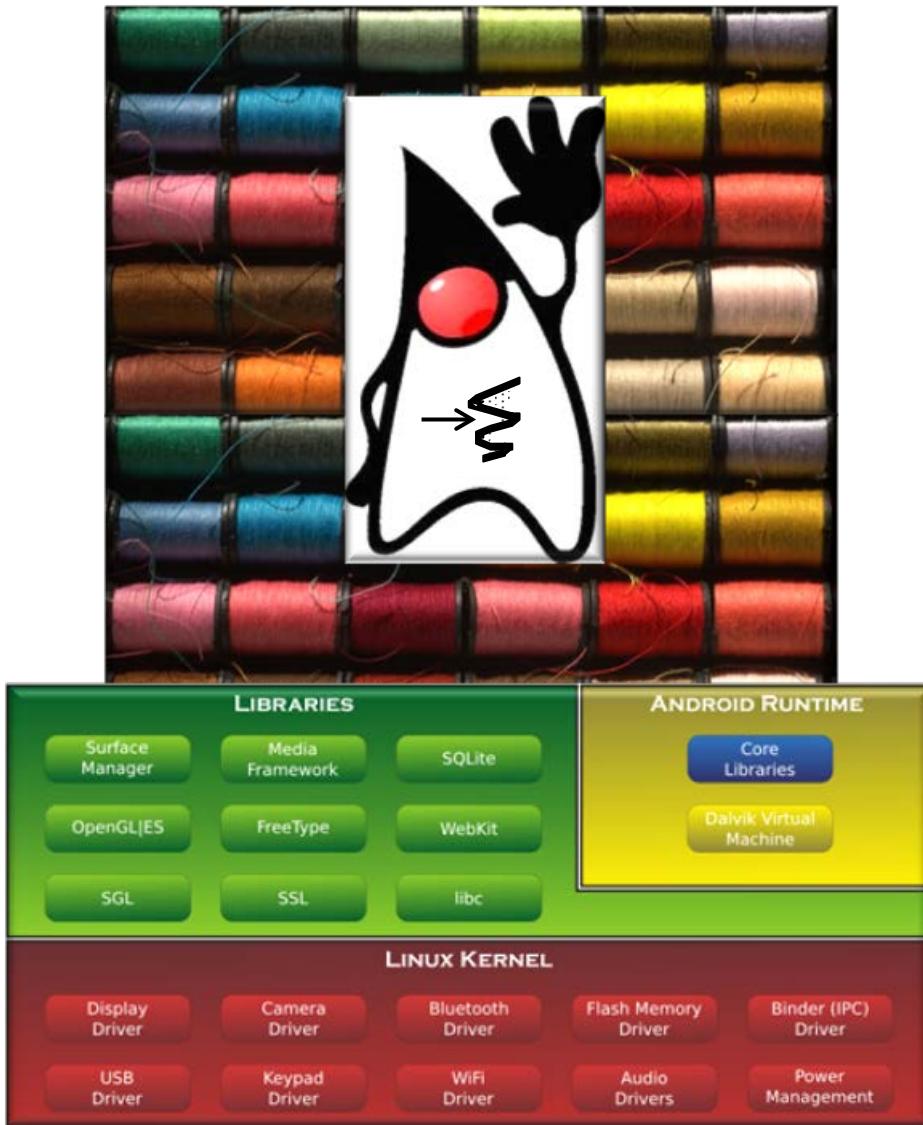
Section 3: Concurrency Patterns in Android

- Part 1: The Thread-Specific Storage Pattern
- Part 2: The Command Processor Pattern
- Part 3: The Active Object Pattern
- Part 4: The Half-Sync/Half-Async Pattern



Overview of Topics Covered in Section 2

- Section 2: Android Concurrency Frameworks
 - Android's concurrency infrastructure uses Java



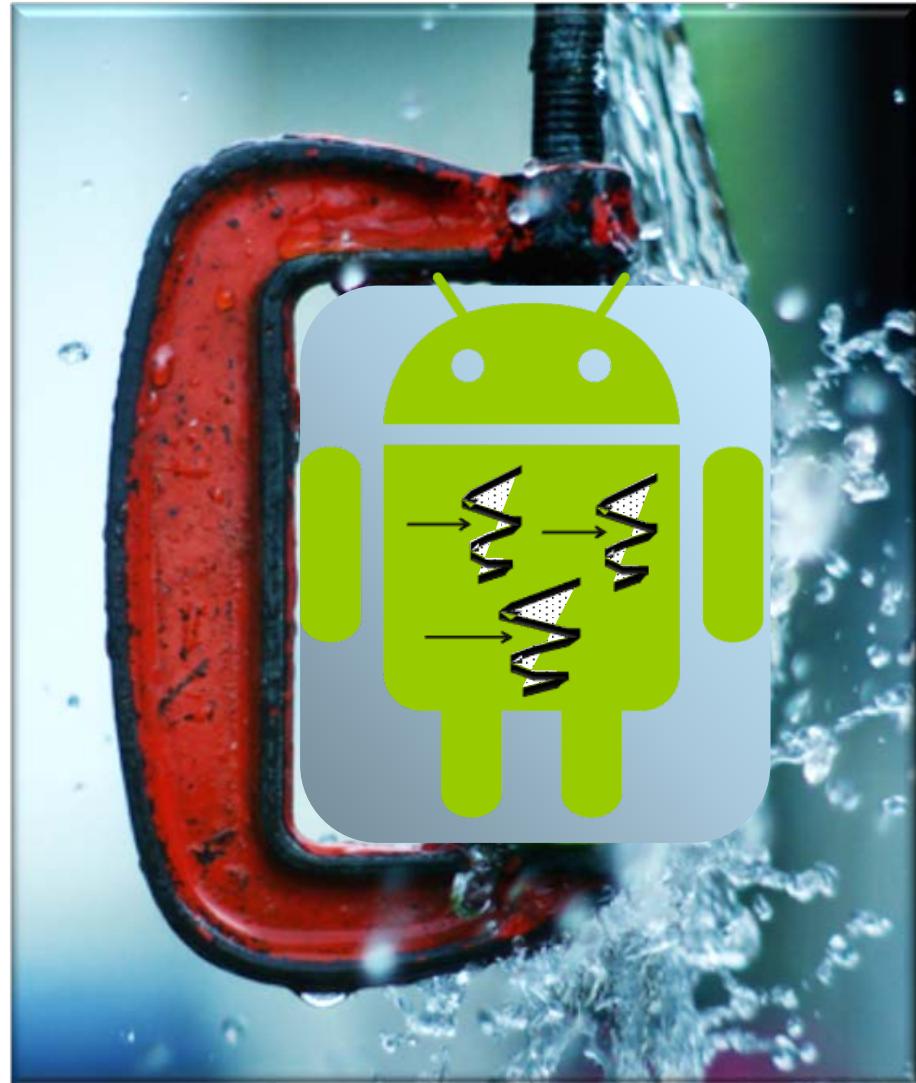
Overview of Topics Covered in Section 2

- **Section 2: Android Concurrency Frameworks**
 - Android's concurrency infrastructure uses Java
 - Java's mechanisms don't address Android design constraints



Overview of Topics Covered in Section 2

- **Section 2: Android Concurrency Frameworks**
 - Android's concurrency infrastructure uses Java
 - Java's mechanisms don't address Android design constraints



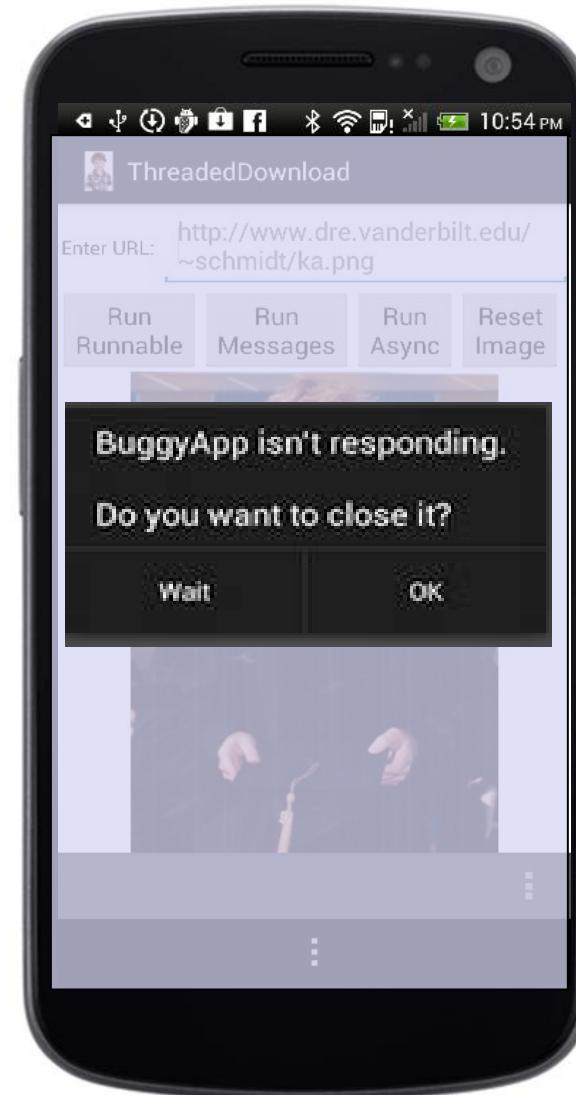
Overview of Topics Covered in Section 2

- **Section 2: Android Concurrency Frameworks**
 - Android's concurrency infrastructure uses Java
 - Java's mechanisms don't address Android design constraints, e.g.
 - GUI libraries aren't thread-safe



Overview of Topics Covered in Section 2

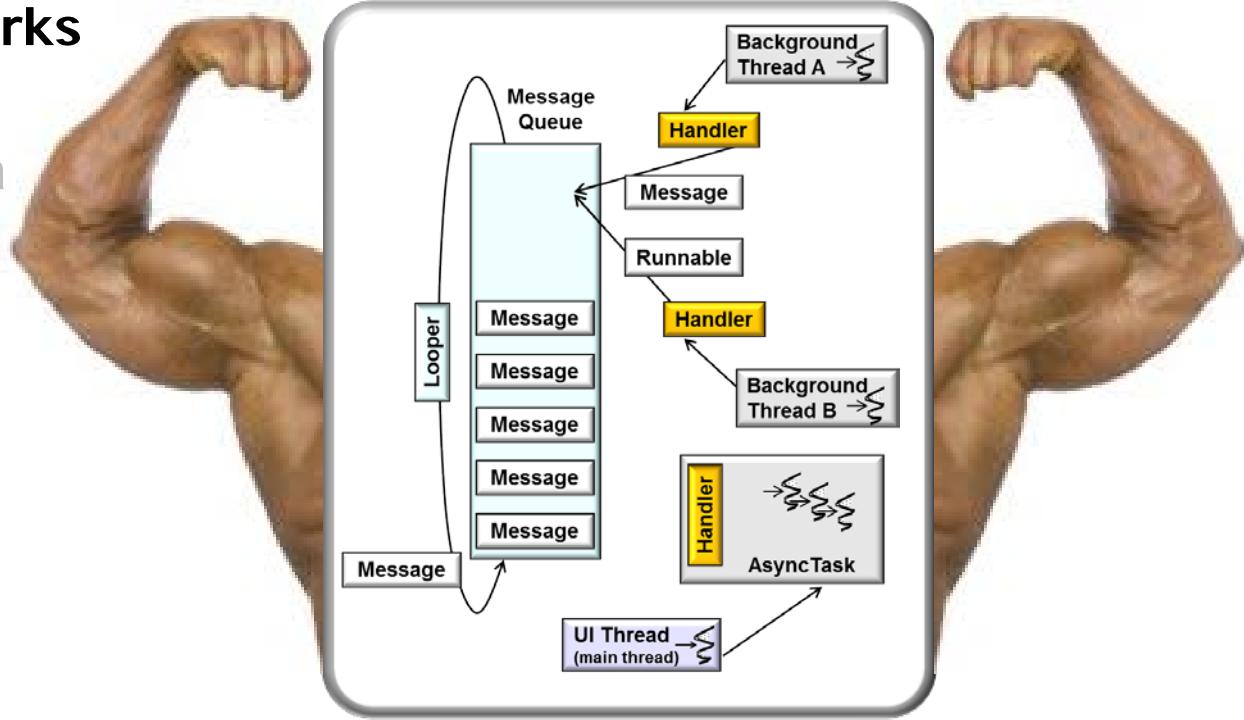
- **Section 2: Android Concurrency Frameworks**
 - Android's concurrency infrastructure uses Java
 - Java's mechanisms don't address Android design constraints, e.g.
 - GUI libraries aren't thread-safe
 - The UI Thread can't block for long



Overview of Topics Covered in Section 2

- **Section 2: Android Concurrency Frameworks**

- Android's concurrency infrastructure uses Java
- Android's concurrency frameworks overcome its design constraints



Overview of Topics Covered in Section 2

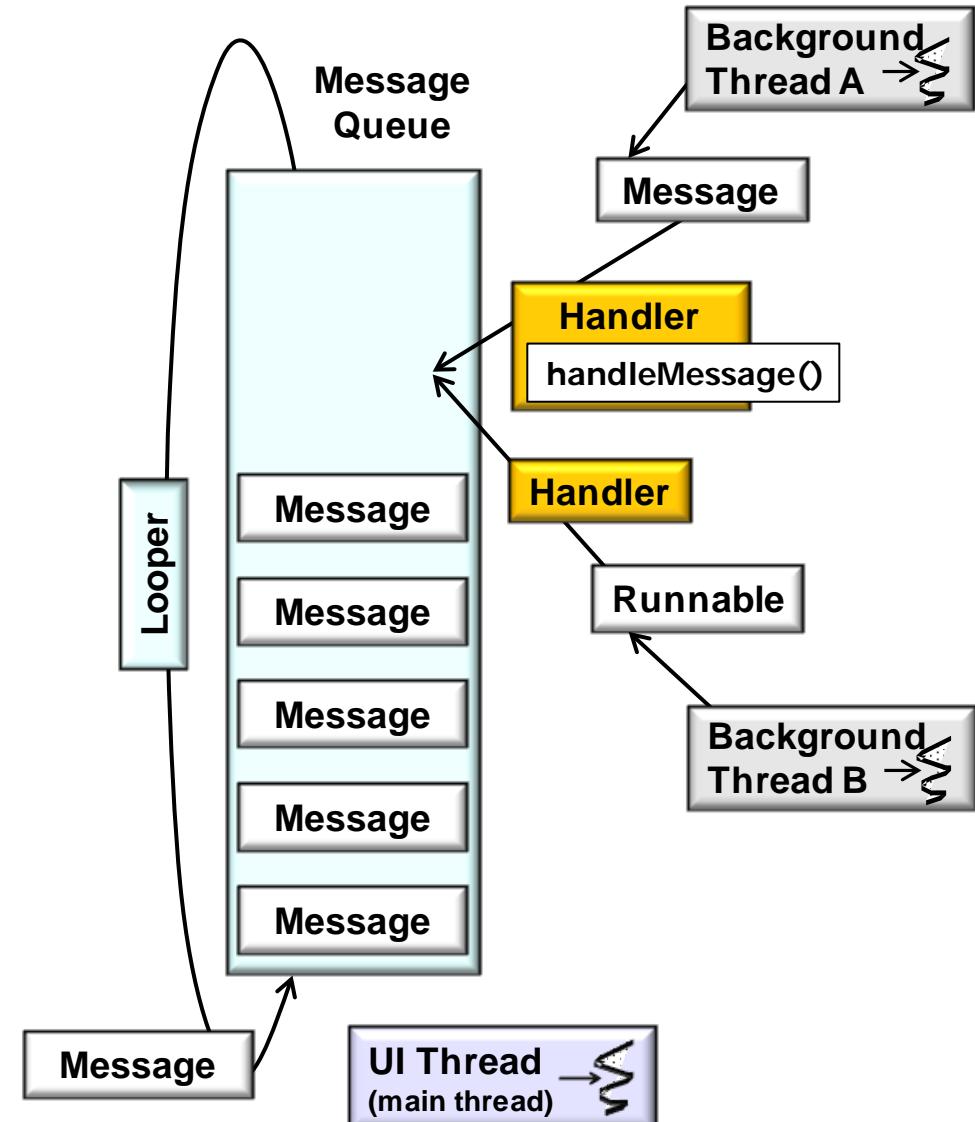
- **Section 2: Android Concurrency Frameworks**
 - Android's concurrency infrastructure uses Java
 - Android's concurrency frameworks overcome its design constraints



Overview of Topics Covered in Section 2

- **Section 2: Android Concurrency Frameworks**

- Android's concurrency infrastructure uses Java
- Android's concurrency frameworks overcome its design constraints
 - Handlers, Messages, & Runnables (HaMeR)

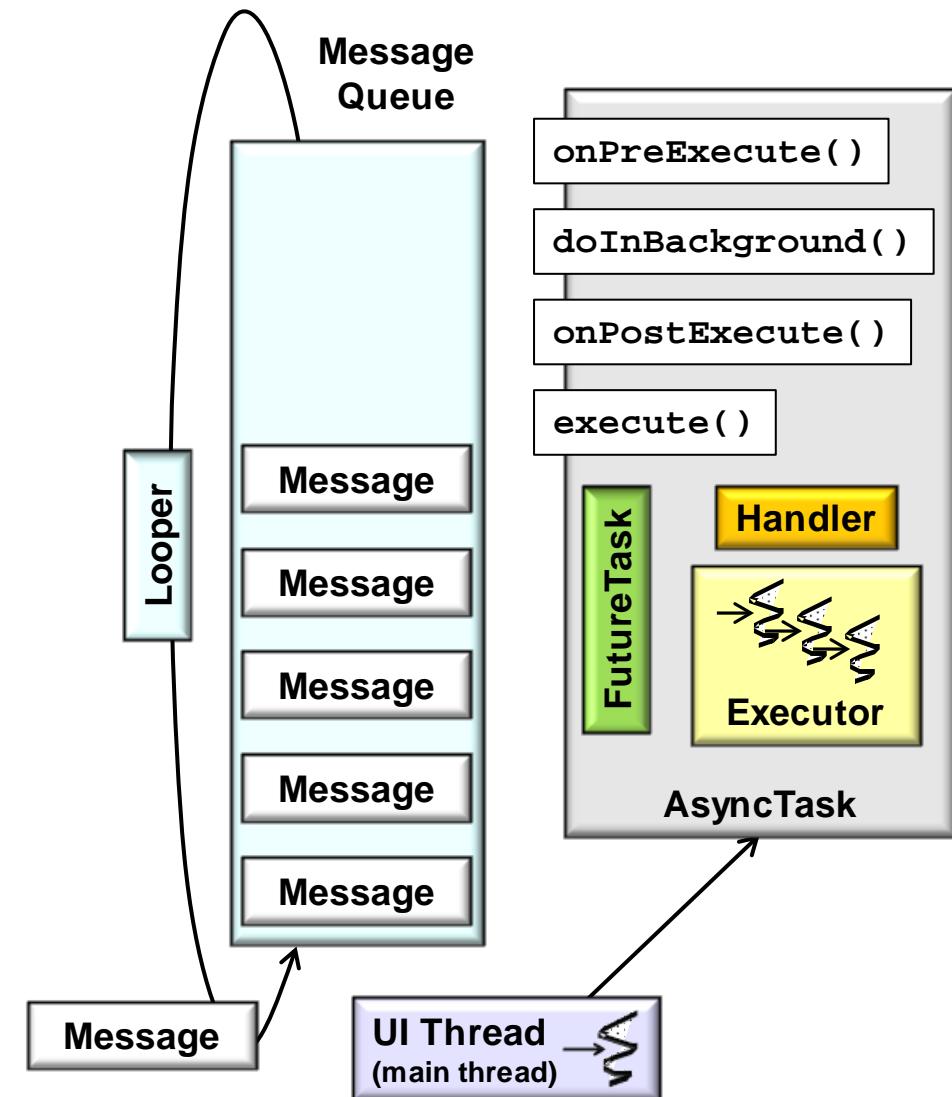


See developer.android.com/guide/components/processes-and-threads.html#Threads

Overview of Topics Covered in Section 2

- **Section 2: Android Concurrency Frameworks**

- Android's concurrency infrastructure uses Java
- Android's concurrency frameworks overcome its design constraints
 - Handlers, Messages, & Runnables (HaMeR)
- AsyncTask framework

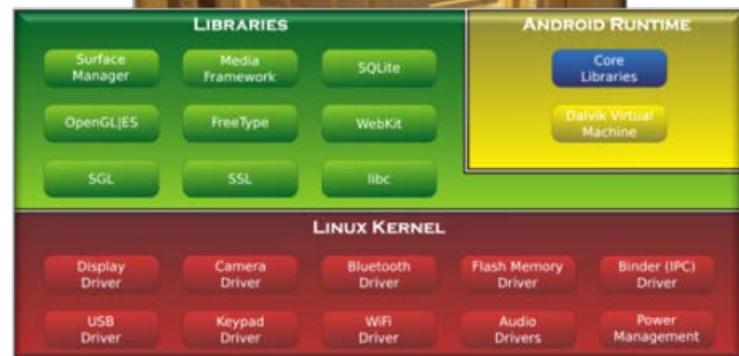


See [developer.android.com/reference/
android/os/AsyncTask.html](http://developer.android.com/reference/android/os/AsyncTask.html)

Overview of Topics Covered in Section 2

- **Section 2: Android Concurrency Frameworks**

- Android's concurrency infrastructure uses Java
- Android's concurrency frameworks overcome its design constraints

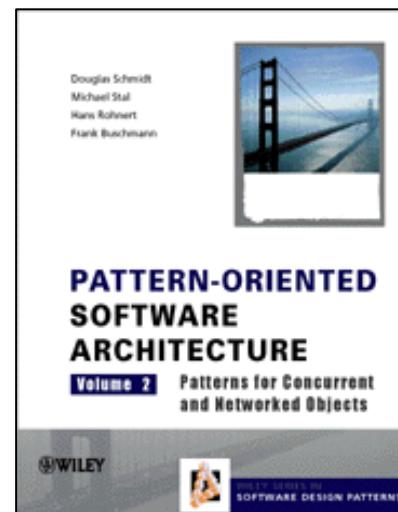
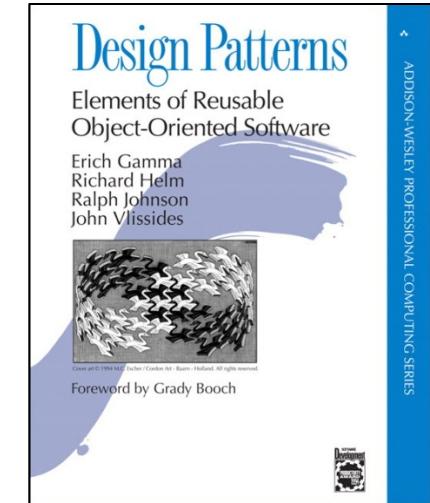
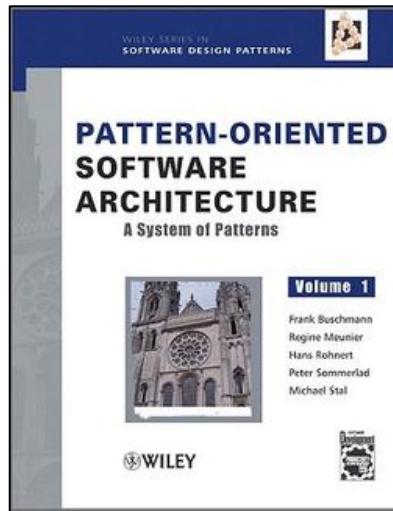


Android's concurrency frameworks are applied throughout its software stack

Overview of Topics Covered in Section 2

- **Section 2: Android Concurrency Frameworks**

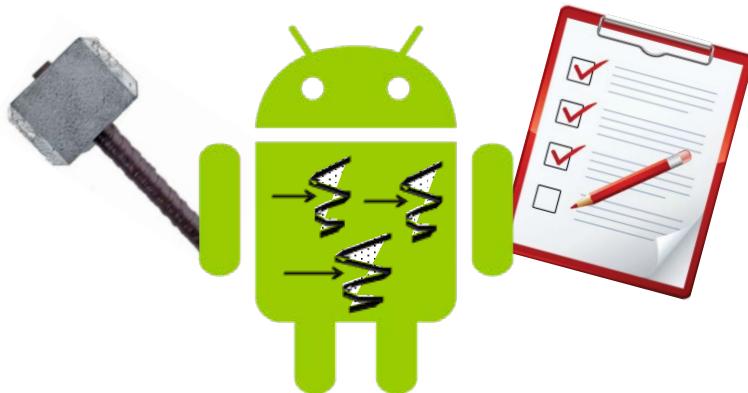
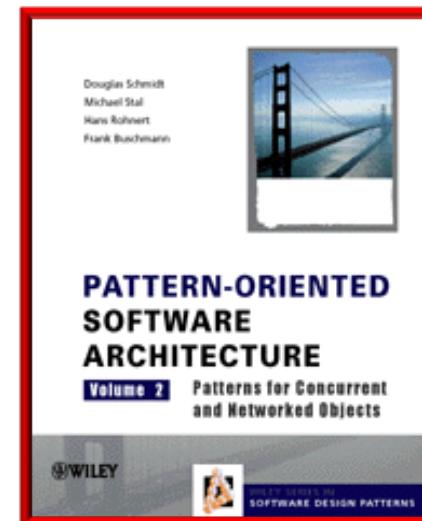
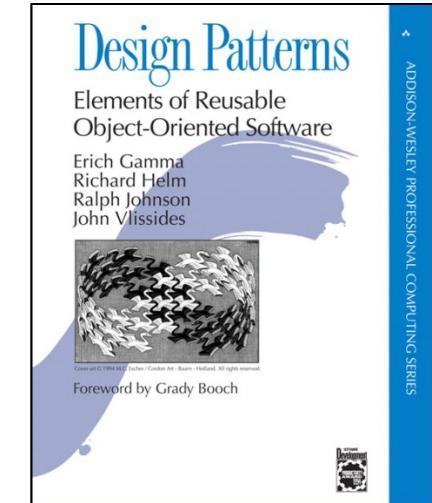
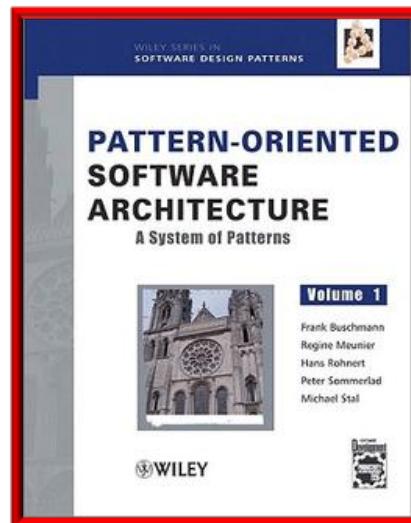
- Android's concurrency infrastructure uses Java
- Android's concurrency frameworks overcome its design constraints
- Android's concurrency frameworks apply many software patterns



Overview of Topics Covered in Section 2

• Section 2: Android Concurrency Frameworks

- Android's concurrency infrastructure uses Java
- Android's concurrency frameworks overcome its design constraints
- Android's concurrency frameworks apply many software patterns

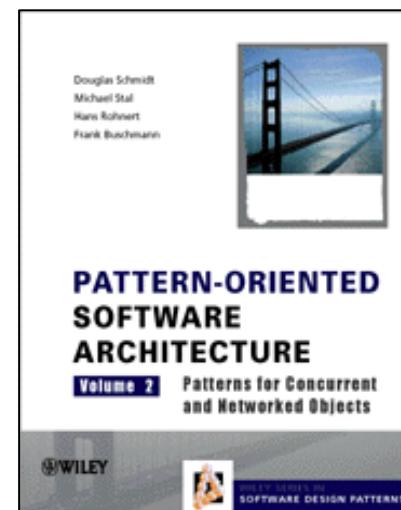
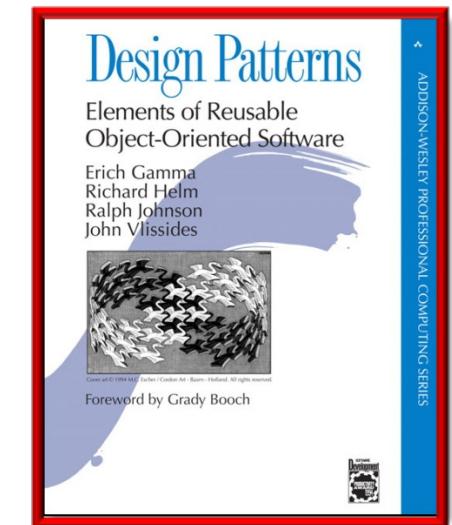
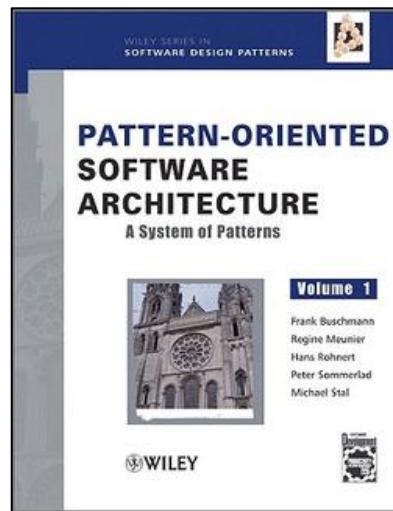


See www.dre.vanderbilt.edu/~schmidt/POSA

Overview of Topics Covered in Section 2

- **Section 2: Android Concurrency Frameworks**

- Android's concurrency infrastructure uses Java
- Android's concurrency frameworks overcome its design constraints
- Android's concurrency frameworks apply many software patterns

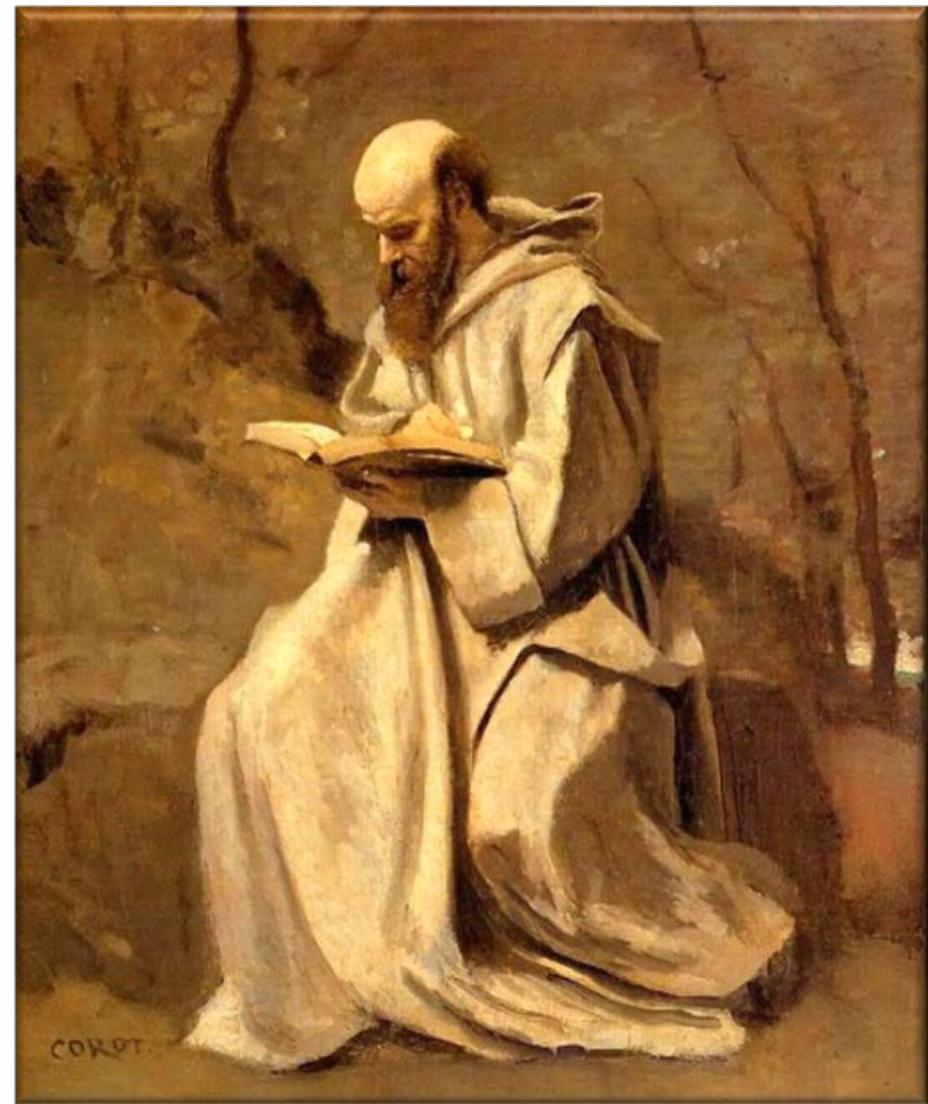


See [en.wikipedia.org/
wiki/Design_Patterns](https://en.wikipedia.org/wiki/Design_Patterns)

Overview of Topics Covered in Section 2

- **Section 2: Android Concurrency Frameworks**

- Android's concurrency infrastructure uses Java
- Android's concurrency frameworks overcome its design constraints
- Android's concurrency frameworks apply many software patterns



Overview of the MOOC Topics in Section 3

Overview of Topics Covered in Section 3

Section 0: MOOC Introduction

- Part 1: MOOC Organization & Topics
- Part 2: MOOC Prereqs, Workload, & Learning Strategies
- Part 3&4: Concurrency Motivations & Challenges
- Part 5&6: Overview of Patterns and Frameworks

Section 1: Overview of Android & Java Concurrency

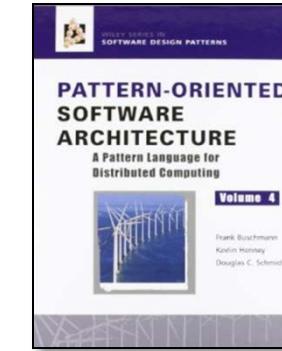
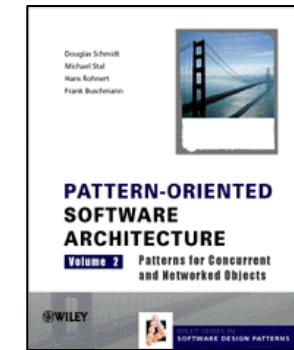
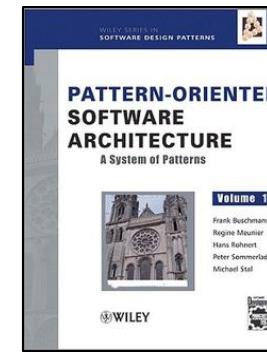
- Part 1: Android Layers
- Part 2: Java Threading Mechanisms
- Part 3: Java Built-in Synchronization Mechanisms

Section 2: Android Concurrency Frameworks

- Part 1&2: Concurrency Framework Classes & Patterns
- Part 3: The Threaded Downloads Application
- Part 4: Android Looper
- Part 5: Android Handler & HaMeR Framework
- Part 6: Posting/Processing Runnables with the HaMeR Framework
- Part 7: Sending/Handling Messages with the HaMeR Framework
- Part 8: The AsyncTask Framework
- Part 9: Blackbox & Whitebox Frameworks with AsyncTask
- Part 10: Evaluating Android Concurrency Frameworks

Section 3: Concurrency Patterns in Android

- Part 1: The Thread-Specific Storage Pattern
- Part 2: The Command Processor Pattern
- Part 3: The Active Object Pattern
- Part 4: The Half-Sync/Half-Async Pattern



Overview of Topics Covered in Section 3

Section 0: MOOC Introduction

- Part 1: MOOC Organization & Topics
- Part 2: MOOC Prereqs, Workload, & Learning Strategies
- Part 3&4: Concurrency Motivations & Challenges
- Part 5&6: Overview of Patterns and Frameworks

Section 1: Overview of Android & Java Concurrency

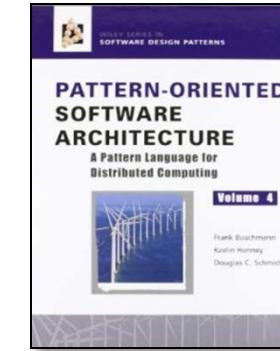
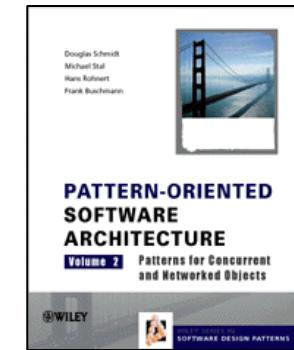
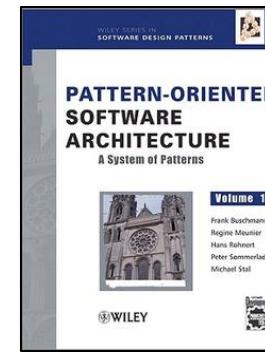
- Part 1: Android Layers
- Part 2: Java Threading Mechanisms
- Part 3: Java Built-in Synchronization Mechanisms

Section 2: Android Concurrency Frameworks

- Part 1&2: Concurrency Framework Classes & Patterns
- Part 3: The Threaded Downloads Application
- Part 4: Android Looper
- Part 5: Android Handler & HaMeR Framework
- Part 6: Posting/Processing Runnables with the HaMeR Framework
- Part 7: Sending/Handling Messages with the HaMeR Framework
- Part 8: The AsyncTask Framework
- Part 9: Blackbox & Whitebox Frameworks with AsyncTask
- Part 10: Evaluating Android Concurrency Frameworks

Section 3: Concurrency Patterns in Android

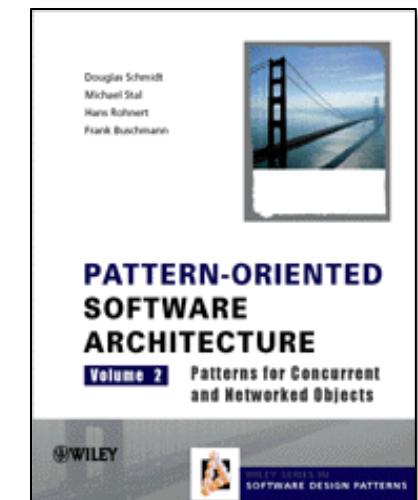
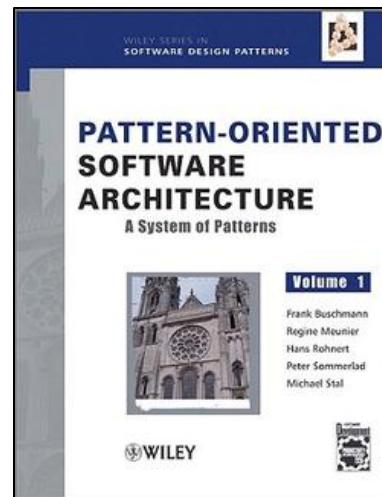
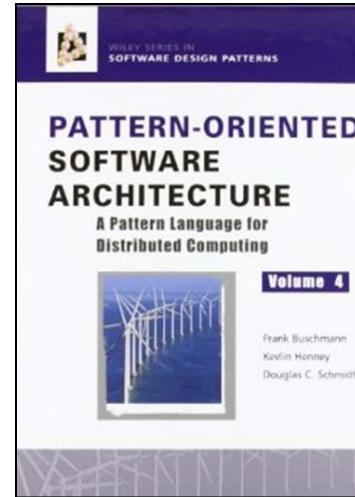
- Part 1: The Thread-Specific Storage Pattern
- Part 2: The Command Processor Pattern
- Part 3: The Active Object Pattern
- Part 4: The Half-Sync/Half-Async Pattern



Overview of Topics Covered in Section 3

- **Section 3: Concurrency Patterns Applied in Android**

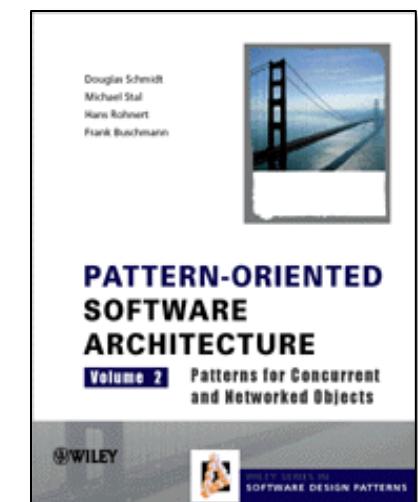
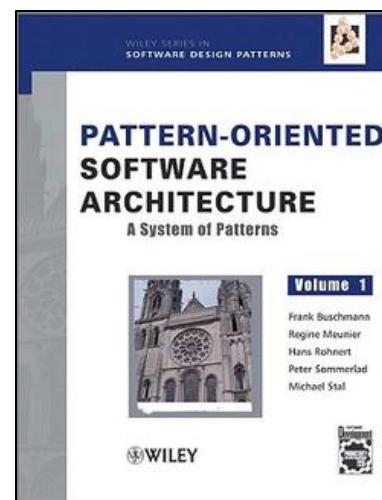
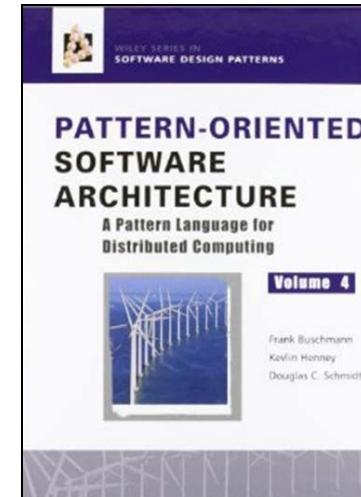
- *Monitor Object*
- *Thread-Specific Storage*
- *Command Processor*
- *Active Object*
- *Half-Sync/Half-Async*
- *Guarded Suspension*
- *Future*
- *Specific Notification*
- *Thread-Safe Interface*
- *Scoped Locking*



Overview of Topics Covered in Section 3

- **Section 3: Concurrency Patterns Applied in Android**

- *Monitor Object*
- *Thread-Specific Storage*
- *Command Processor*
- *Active Object*
- *Half-Sync/Half-Async*
- *Guarded Suspension*
- *Future*
- *Specific Notification*
- *Thread-Safe Interface*
- *Scoped Locking*

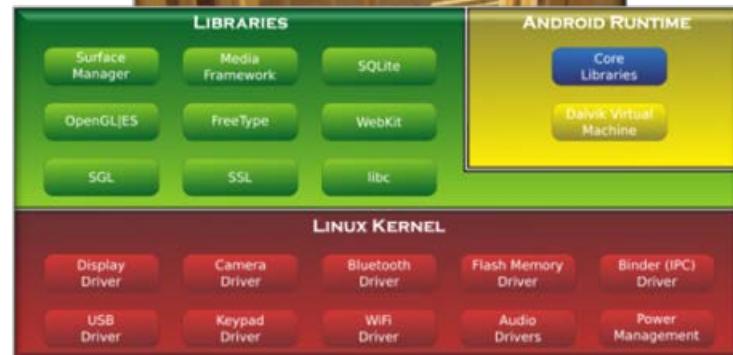
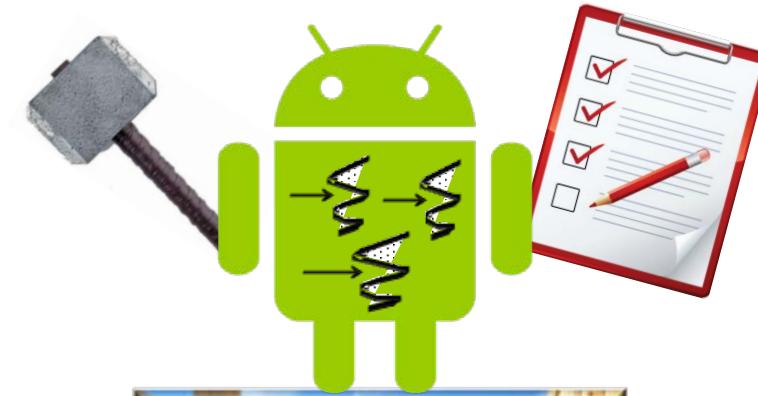


We provide in-depth analysis of how key POSA patterns are applied in concurrent Android & Java source code

Overview of Topics Covered in Section 3

- **Section 3: Concurrency Patterns Applied in Android**

- *Monitor Object*
- *Thread-Specific Storage*
- *Command Processor*
- *Active Object*
- *Half-Sync/Half-Async*
- *Guarded Suspension*
- *Future*
- *Specific Notification*
- *Thread-Safe Interface*
- *Scoped Locking*

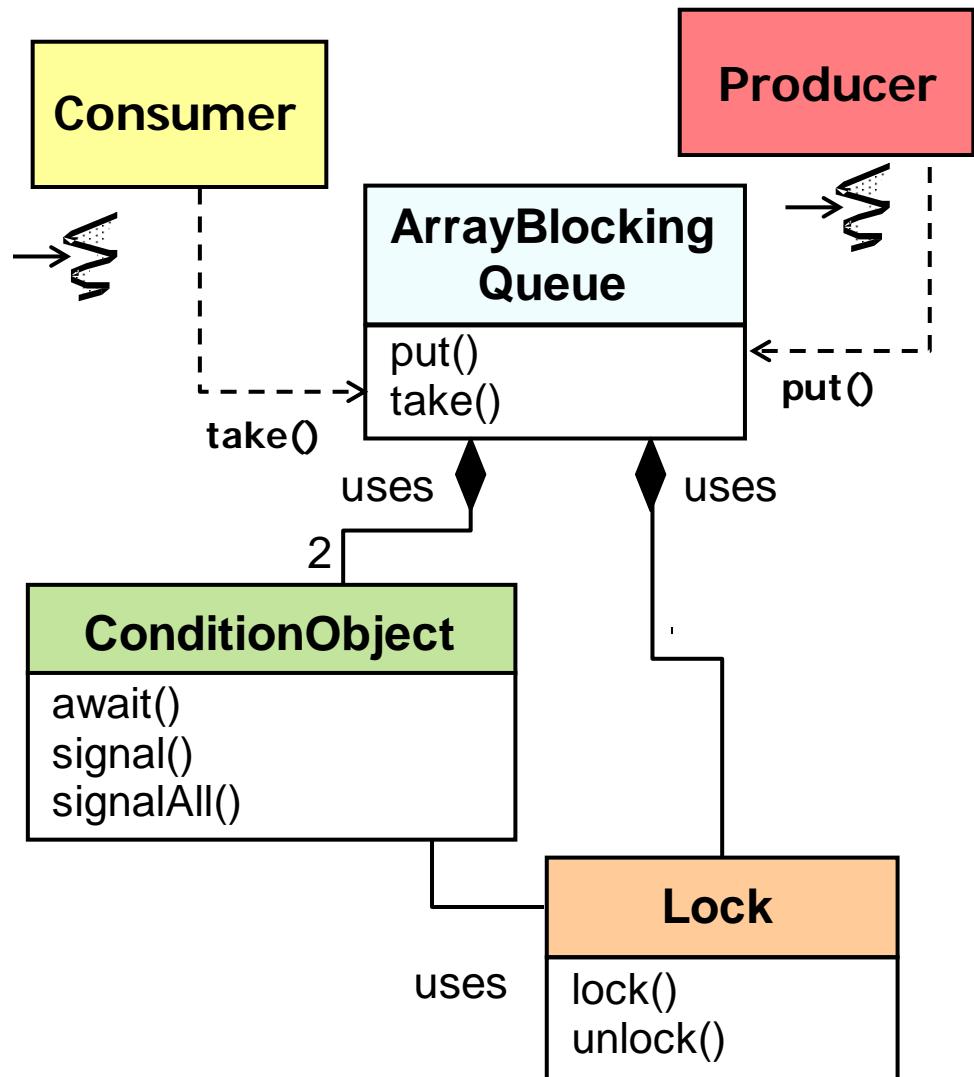


We provide in-depth analysis of how key POSA patterns are applied in concurrent Android & Java source code

Overview of Topics Covered in Section 3

- Section 3: Concurrency Patterns Applied in Android

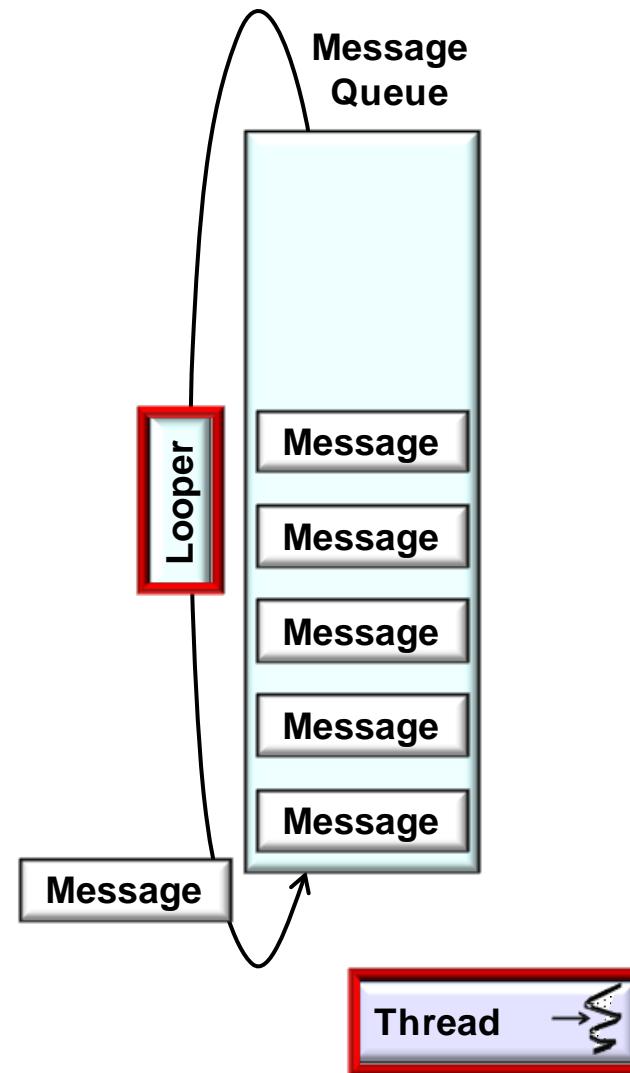
- *Monitor Object*
 - Coordinate concurrent access to shared state



See www.dre.vanderbilt.edu/~schmidt/PDF/monitor.pdf

Overview of Topics Covered in Section 3

- Section 3: Concurrency Patterns Applied in Android
 - *Monitor Object*
 - *Thread-Specific Storage*
 - Ensure only one Looper per Thread

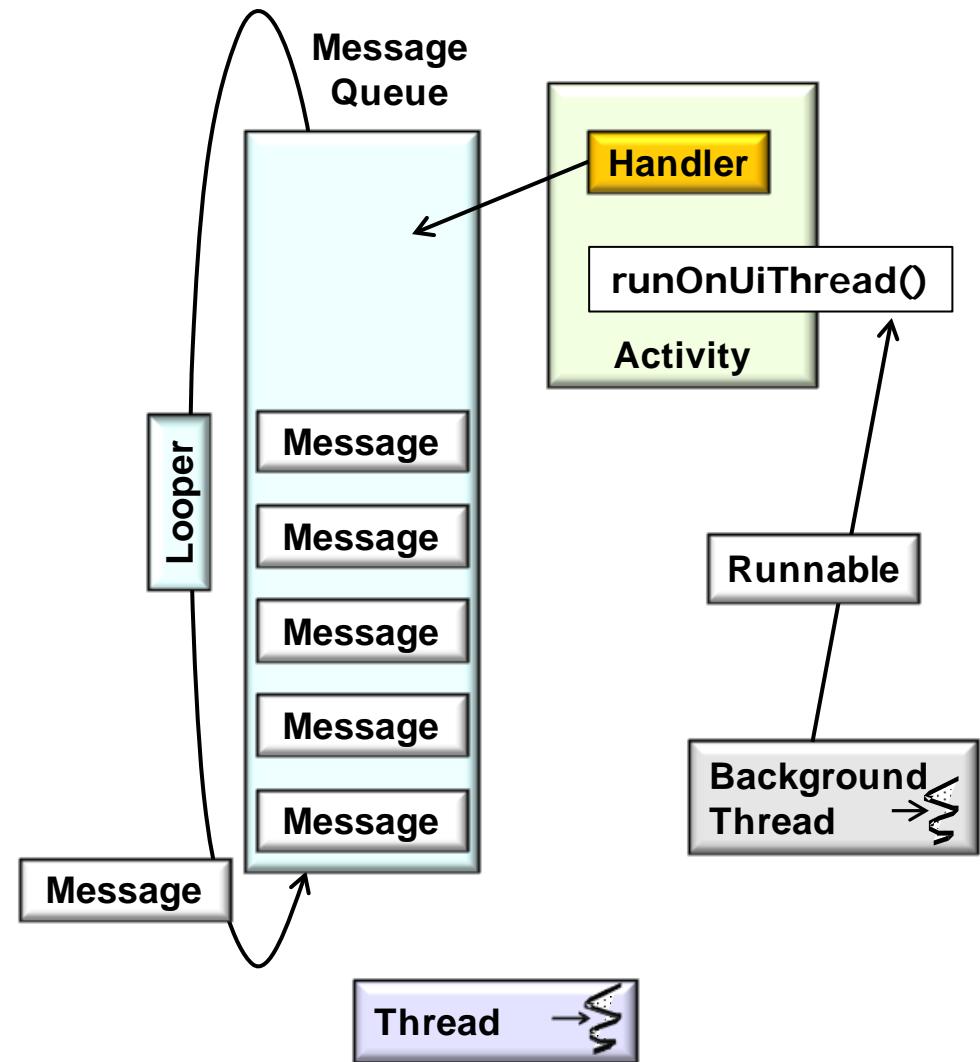


See www.dre.vanderbilt.edu/~schmidt/PDF/TSS-pattern.pdf

Overview of Topics Covered in Section 3

- **Section 3: Concurrency Patterns Applied in Android**

- *Monitor Object*
- *Thread-Specific Storage*
- *Command Processor*
 - Pass commands to Threads

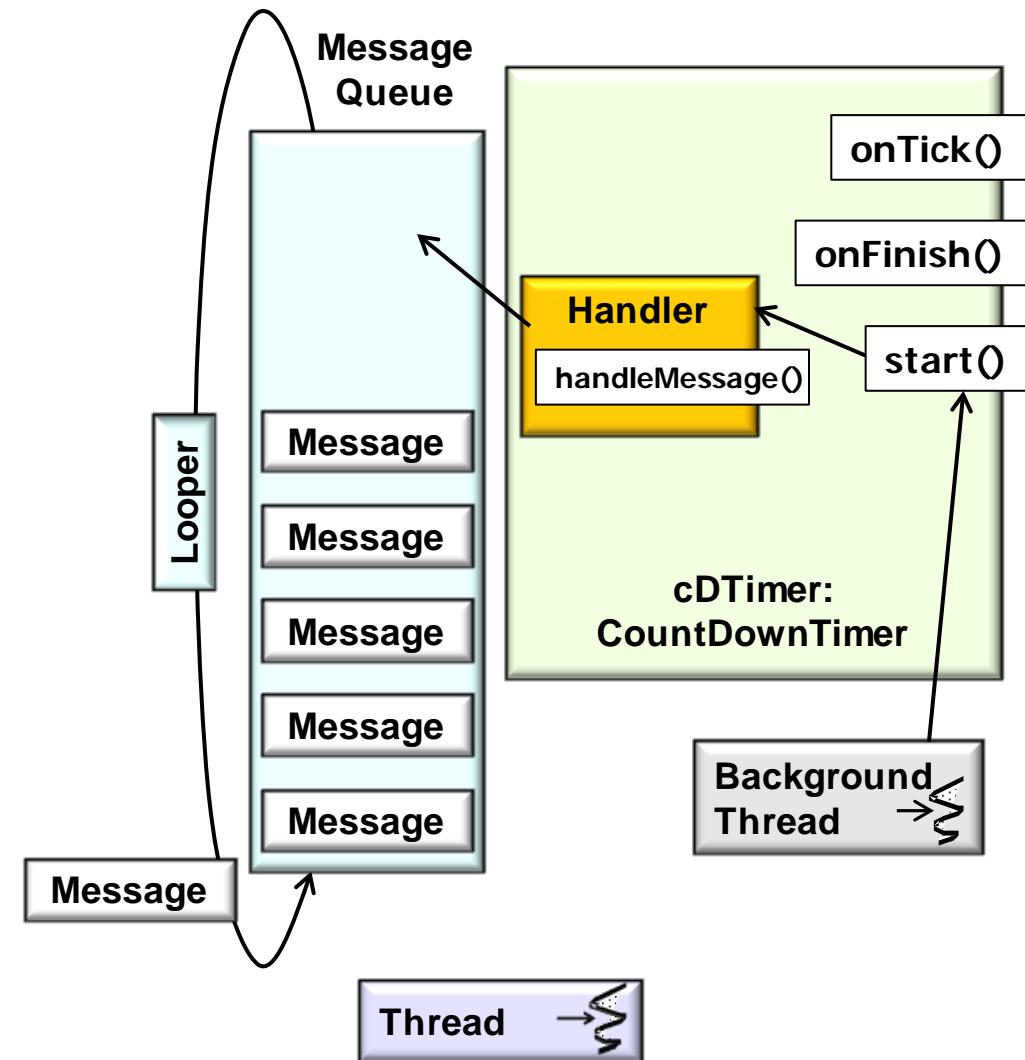


See [www.dre.vanderbilt.edu/~schmidt/
PDF/CommandProcessor.pdf](http://www.dre.vanderbilt.edu/~schmidt/PDF/CommandProcessor.pdf)

Overview of Topics Covered in Section 3

- **Section 3: Concurrency Patterns Applied in Android**

- *Monitor Object*
- *Thread-Specific Storage*
- *Command Processor*
- *Active Object*
 - Pass Message requests to Threads

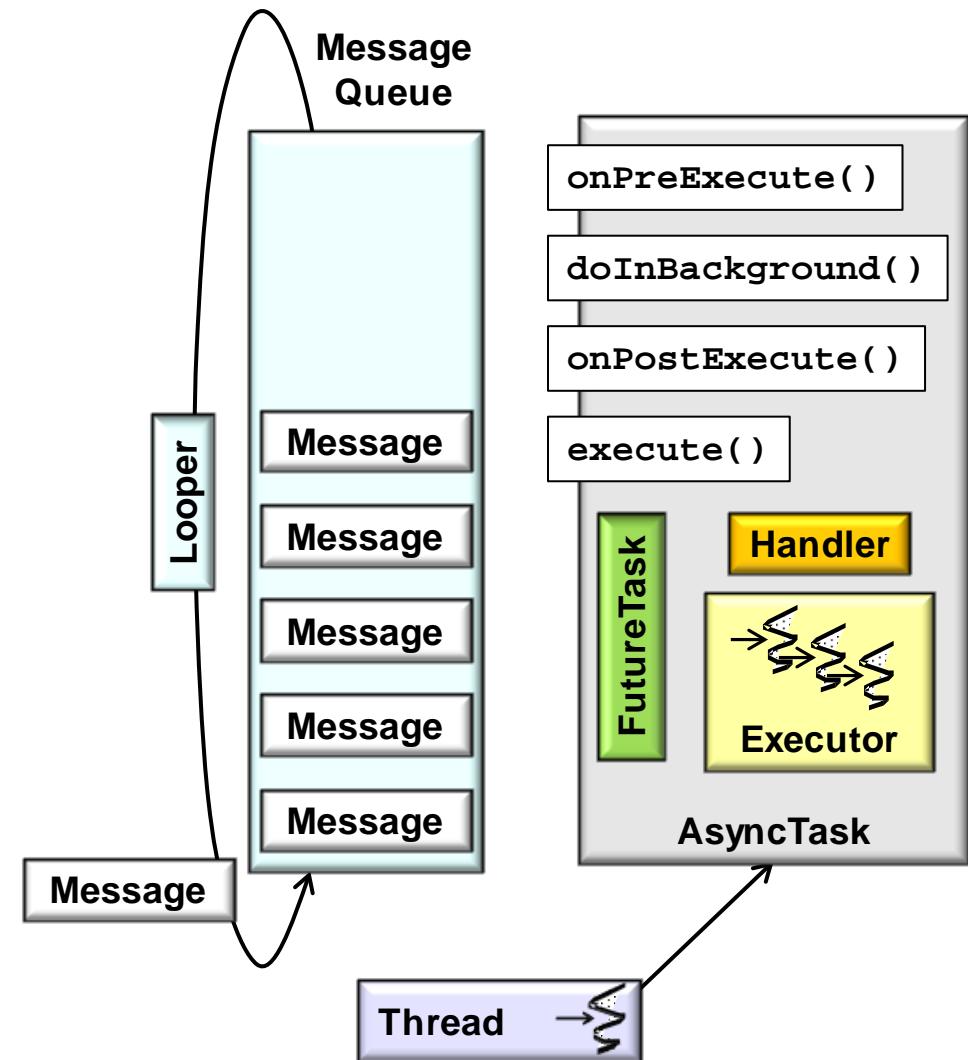


See [en.wikipedia.org/
wiki/Active_object](https://en.wikipedia.org/wiki/Active_object)

Overview of Topics Covered in Section 3

- **Section 3: Concurrency Patterns Applied in Android**

- *Monitor Object*
- *Thread-Specific Storage*
- *Command Processor*
- *Active Object*
- *Half-Sync/Half-Async*
 - Decouple synchronous & asynchronous processing

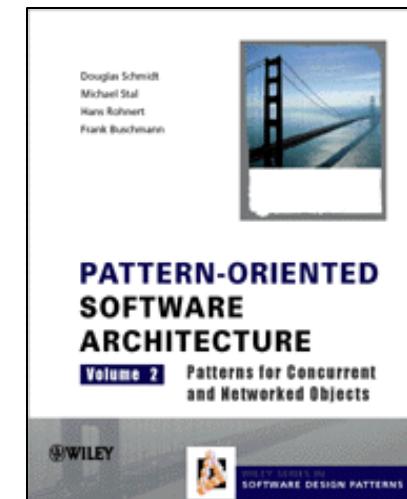
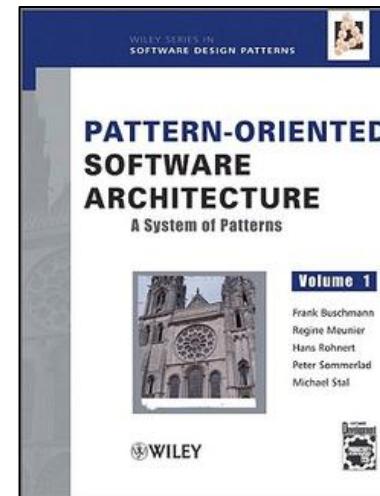
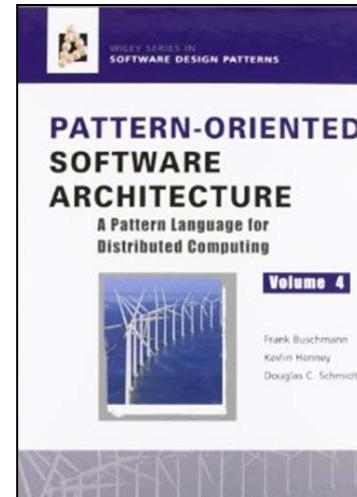


See www.dre.vanderbilt.edu/~schmidt/PDF/HS-HA.pdf

Overview of Topics Covered in Section 3

- **Section 3: Concurrency Patterns Applied in Android**

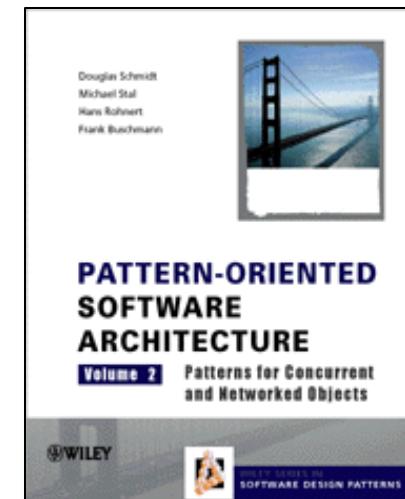
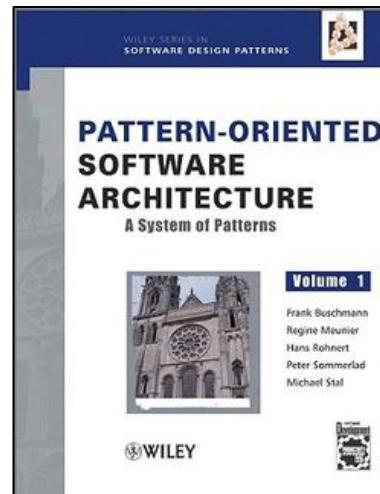
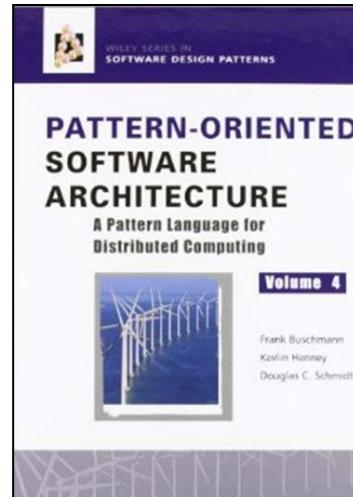
- *Monitor Object*
- *Thread-Specific Storage*
- *Command Processor*
- *Active Object*
- *Half-Sync/Half-Async*
- *Guarded Suspension*
- *Future*
- *Specific Notification*
- *Thread-Safe Interface*
- *Scoped Locking*



See www.dre.vanderbilt.edu/~schmidt/DigitalLearning

Overview of Topics Covered in Section 3

- Section 3: Concurrency Patterns Applied in Android



These patterns aren't limited to Android, Java, or mobile device programming

Overview of the MOOC Organization

Overview of the MOOC Organization

- The lectures in the four sections of this MOOC are structured uniformly



Overview of the MOOC Organization

- The lectures in the four sections of this MOOC are structured uniformly
 - Each Section is composed of Parts



Overview of the MOOC Organization

- The lectures in the four sections of this MOOC are structured uniformly
 - Each Section is composed of Parts
 - Each Part is a single lecture



Some lectures are pre-recorded,
whereas others are recorded live

Overview of the MOOC Organization

- The lectures in the four sections of this MOOC are structured uniformly
 - Each Section is composed of Parts
 - Each Part is a single lecture
 - Each Part is composed of segments



Overview of the MOOC Organization

- The lectures in the four sections of this MOOC are structured uniformly
 - Each Section is composed of Parts
 - Each Part is a single lecture
 - Each Part is composed of segments
 - Quizzes pop up periodically



If you download the videos & watch them locally the quizzes won't appear!

Overview of the MOOC Organization

- The lectures in the four sections of this MOOC are structured uniformly
 - Each Section is composed of Parts
 - Each Part is a single lecture
 - Each Part is composed of segments
 - Quizzes pop up periodically



See upcoming part on "MOOC Prerequisites, Workload, & Learning Strategies"

Summary

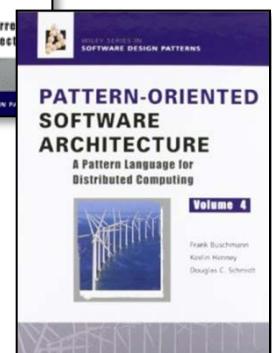
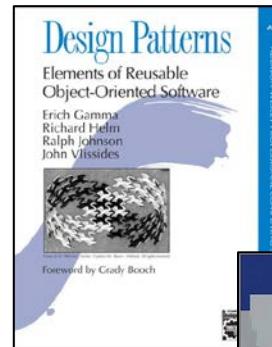


Summary

- This MOOC covers a spectrum of topics

Summary

- This MOOC covers a spectrum of topics



Summary

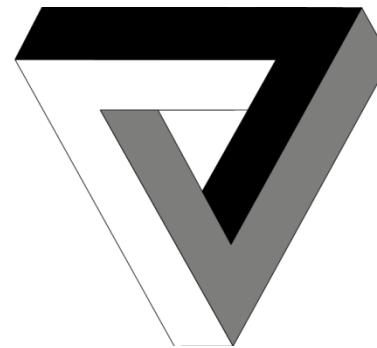
- This MOOC covers a spectrum of topics
- Key learning objectives



See [github.com/douglasraigschmidt/
POSA-15/wiki/POSA-15-FAQ](https://github.com/douglasraigschmidt/POSA-15/wiki/POSA-15-FAQ) item #1

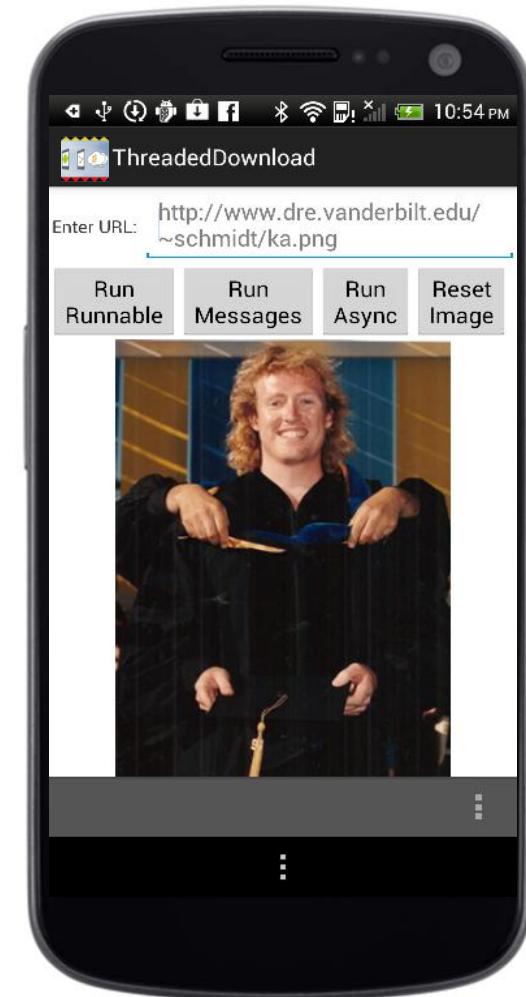
Summary

- This MOOC covers a spectrum of topics
- Key learning objectives
 - Recognize inherent & accidental complexities of concurrent software



Summary

- This MOOC covers a spectrum of topics
- Key learning objectives
 - Recognize inherent & accidental complexities of concurrent software
 - Learn how to apply patterns & frameworks to develop concurrent Android apps



Summary

- This MOOC covers a spectrum of topics
- Key learning objectives
 - Recognize inherent & accidental complexities of concurrent software
 - Learn how to apply patterns & frameworks to develop concurrent Android apps
 - Understand how patterns & frameworks can & cannot help address complexities



Summary



- This MOOC covers a spectrum of topics
- Key learning objectives
 - Recognize inherent & accidental complexities of concurrent software
 - Learn how to apply patterns & frameworks to develop concurrent Android apps
 - Understand how patterns & frameworks can & cannot help address complexities
 - Know where to find other sources of information

Digital Learning Offerings



Douglas C. Schmidt
d.schmidt@vanderbilt.edu
Associate Chair of [Computer Science and Engineering](#),
[Professor](#) of Computer Science,
and Senior Researcher
in the [Institute for Software Integrated Systems \(ISIS\)](#)
at [Vanderbilt University](#)



Coursera MOOCs on Pattern-Oriented Software Architecture (POSA)

- [Mobile Cloud Computing with Android Specialization](#)
- [Spring 2015 Offering of Programming Mobile Services for Android Handheld Systems: Concurrency](#)
- [Spring 2015 Offering of Programming Mobile Services for Android Handheld Systems: Communication](#)
- [Spring 2014 Offering of Pattern-Oriented Software Architecture: Programming Mobile Services for Android Handheld Systems](#)
- [Spring 2013 Offering of Pattern-Oriented Software Architectures for Concurrent and Networked Software](#)

Vanderbilt University Courses

- [Playlist](#) from my [YouTube Channel](#) videos from [CS 251: Intermediate Software Design with Java](#)
- [Playlist](#) from my [YouTube Channel](#) videos from [CS 282: Concurrent Java Network Programming in Android](#)
- [Playlist](#) from my [YouTube Channel](#) videos from [CS 251: Intermediate Software Design with C++](#)
- [Playlist](#) from my [YouTube Channel](#) videos from [CS 282: Systems Programming for Android](#)

Pearson LiveLessons Courses

- [Design Patterns in Java](#)
- [Concurrent Programming in Java](#)

Additional Resources

See www.dre.vanderbilt.edu/~schmidt/DigitalLearning

Summary

- This MOOC covers a spectrum of topics
- Key learning objectives
- We analyze lots of Android software

USE THE
SOURCE LUKE!



See [github.com/douglasraigschmidt/
POSA-15/wiki/POSA-15-FAQ](https://github.com/douglasraigschmidt/POSA-15/wiki/POSA-15-FAQ) item #25

Summary

- This MOOC covers a spectrum of topics
- Key learning objectives
- We analyze lots of Android software
 - It's *essential* to understand Java!!



Summary

- This MOOC covers a spectrum of topics
- Key learning objectives
- We analyze lots of Android software
 - It's *essential* to understand Java!!



See [github.com/douglasraigschmidt/
POSA-15/wiki/POSA-15-FAQ](https://github.com/douglasraigschmidt/POSA-15/wiki/POSA-15-FAQ) item #7