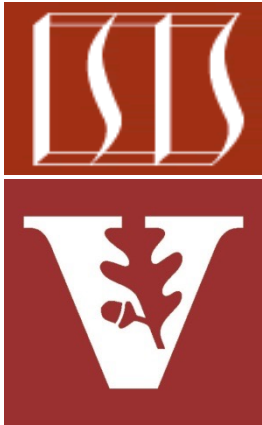


CS 253: Parallel Functional Programming with Java, Android, & Spring WebFlux: Overview (Part 1)

Douglas C. Schmidt

d.schmidt@vanderbilt.edu

www.dre.vanderbilt.edu/~schmidt



Professor of Computer Science

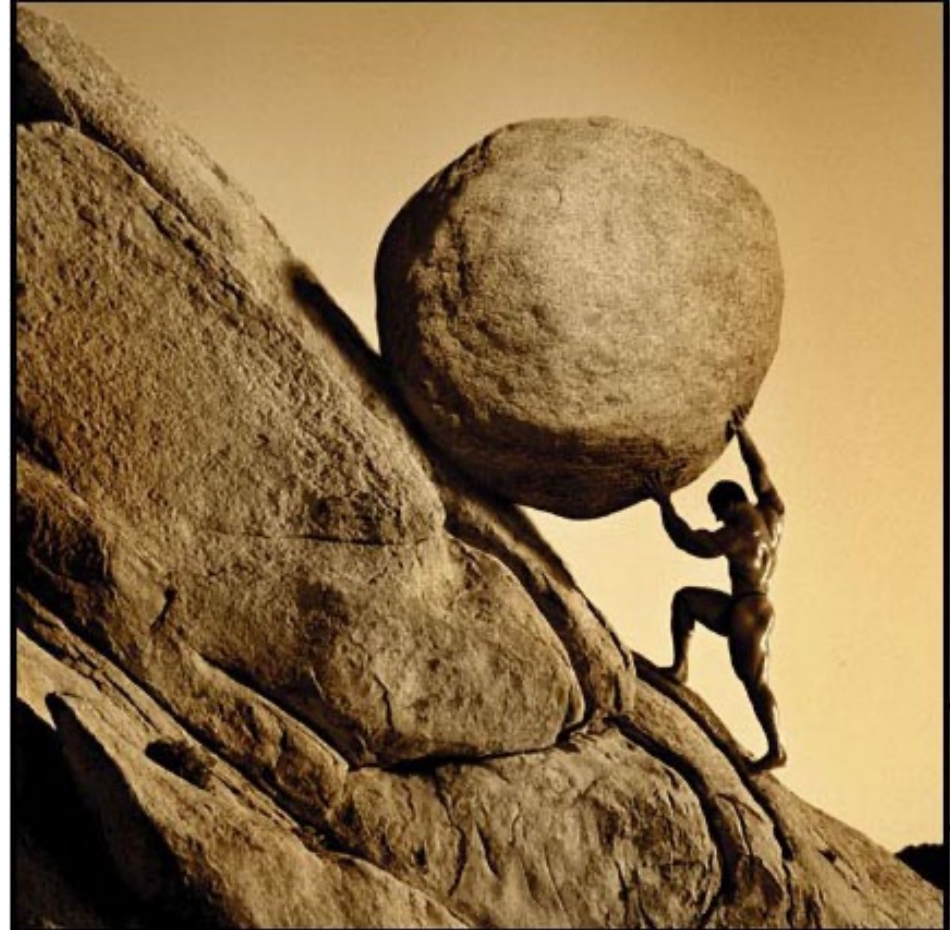
**Institute for Software
Integrated Systems**

**Vanderbilt University
Nashville, Tennessee, USA**



Learning Objectives in this Lesson

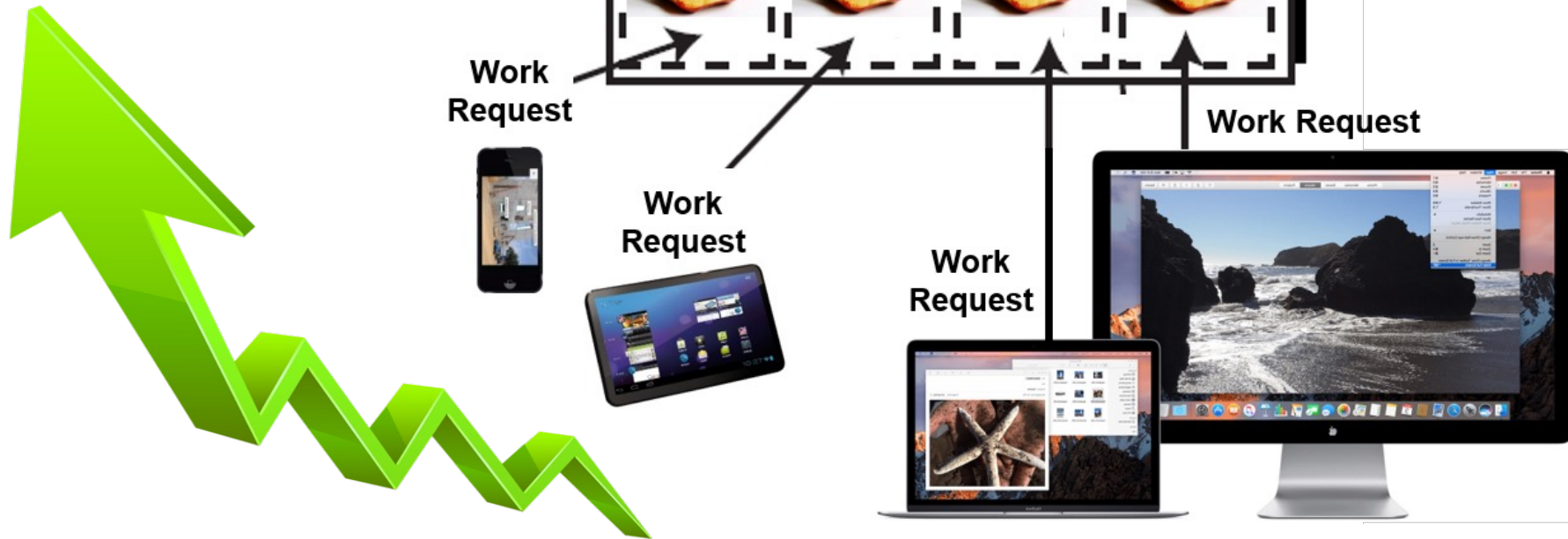
- Understand the course topics & logistics
 - Course philosophy
 - Course contents
 - Structure of the lecture material



Course Philosophy

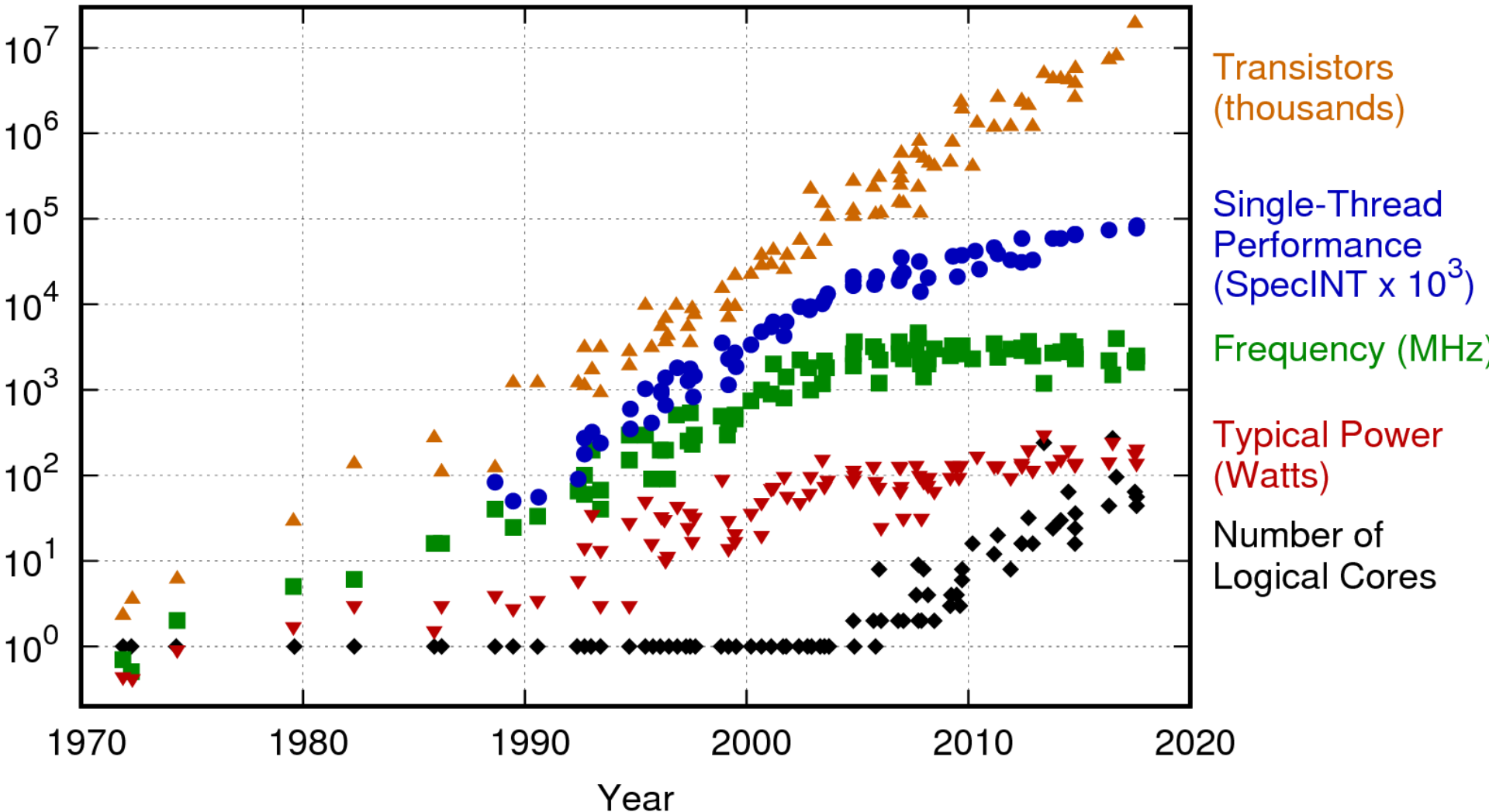
Course Philosophy

- There's a growing need for software developers who know how to write parallel programs for a range of computing platforms
- e.g., mobile devices, laptops, desktops, & cloud environments



Course Philosophy

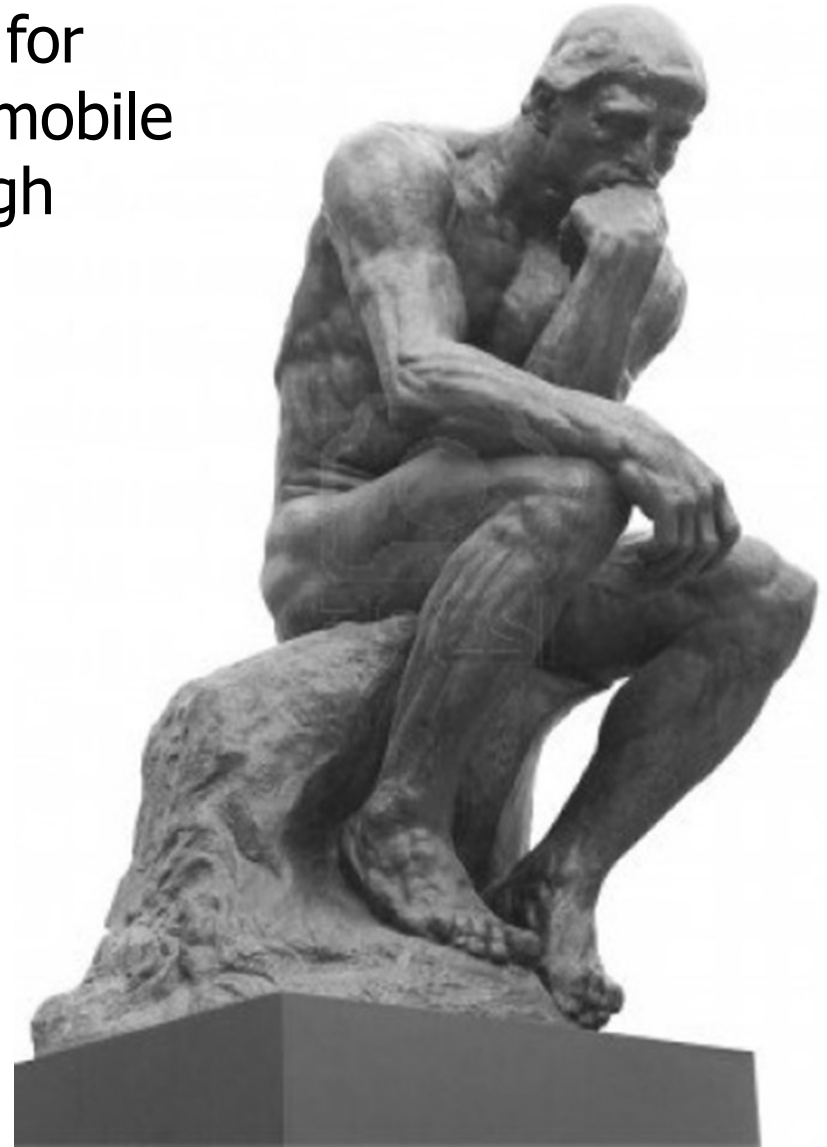
- Demand is driven by software/hardware infrastructure advances



See www.gotw.ca/publications/concurrency-ddj.htm

Course Philosophy

- Effective techniques & practices for developing parallel programs & mobile apps are *not* best learned through generalities & platitudes



“Sitting & thinking” is not sufficient...

Course Philosophy

- Instead, it's better to see *by example* how these programs can be made
 - *easier* to write & read,
 - *easier* to maintain & modify,
 - *more* efficient & resilientby applying time-proven software patterns & object-oriented & functional design & programming techniques



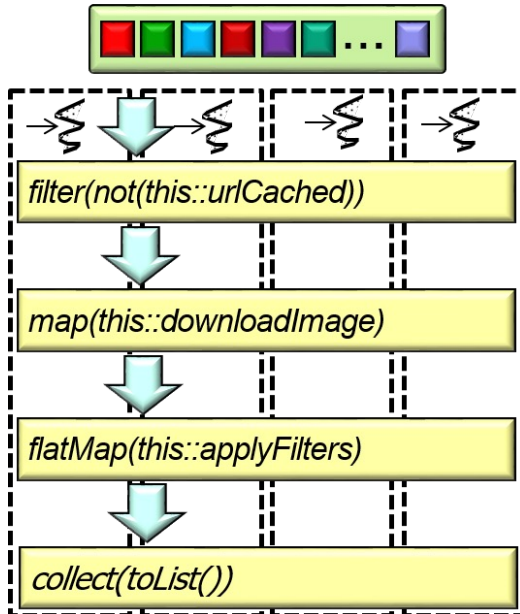
This course involves lots of hands-on software development & testing!

Summary of the Course Contents

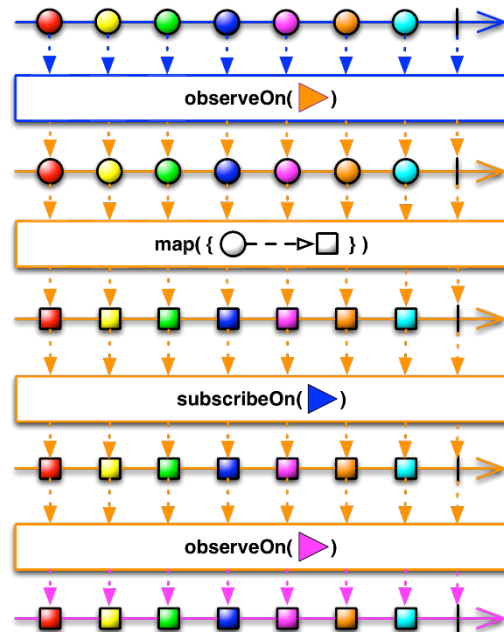
Summary of Course Contents

- Key Java parallelism frameworks

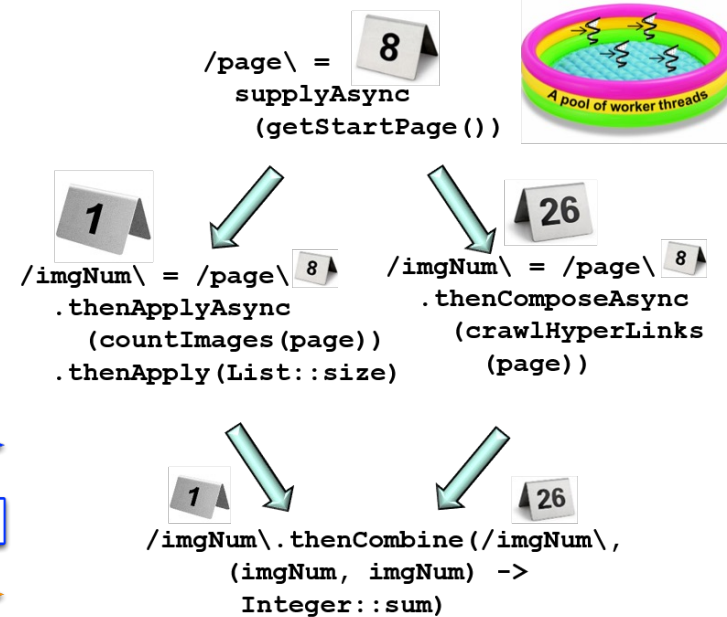
Parallel Streams



Reactive Streams



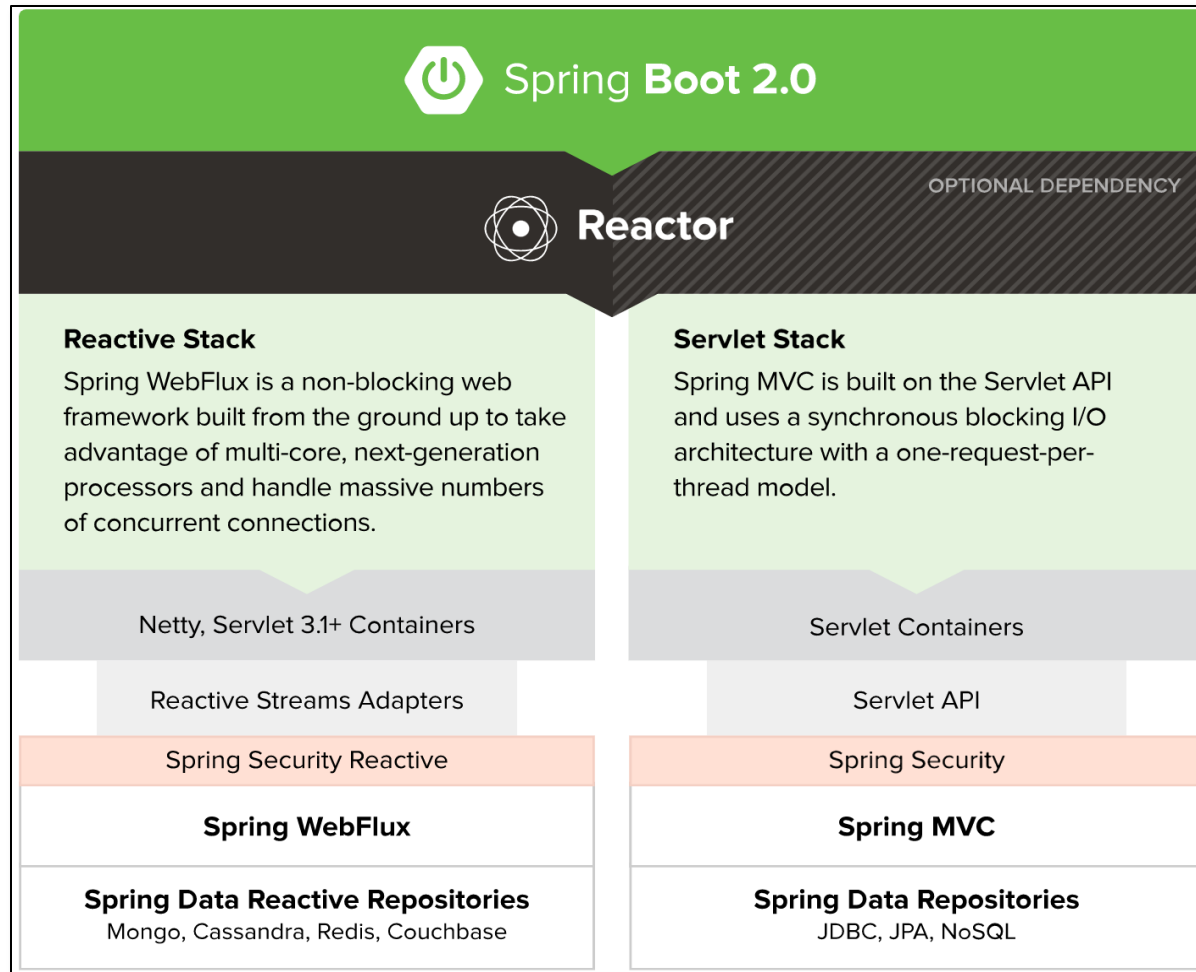
Completable Futures



Assumes knowledge of Java object-oriented & functional language features

Summary of Course Contents

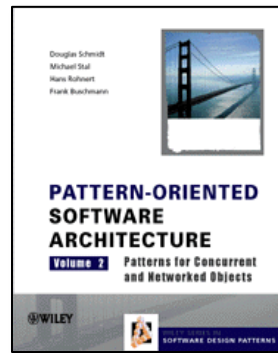
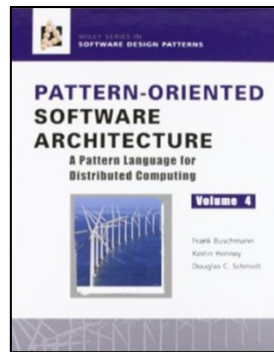
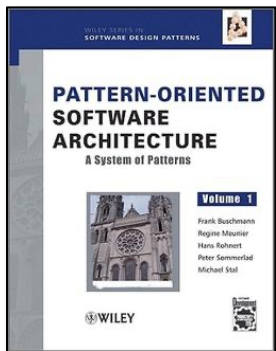
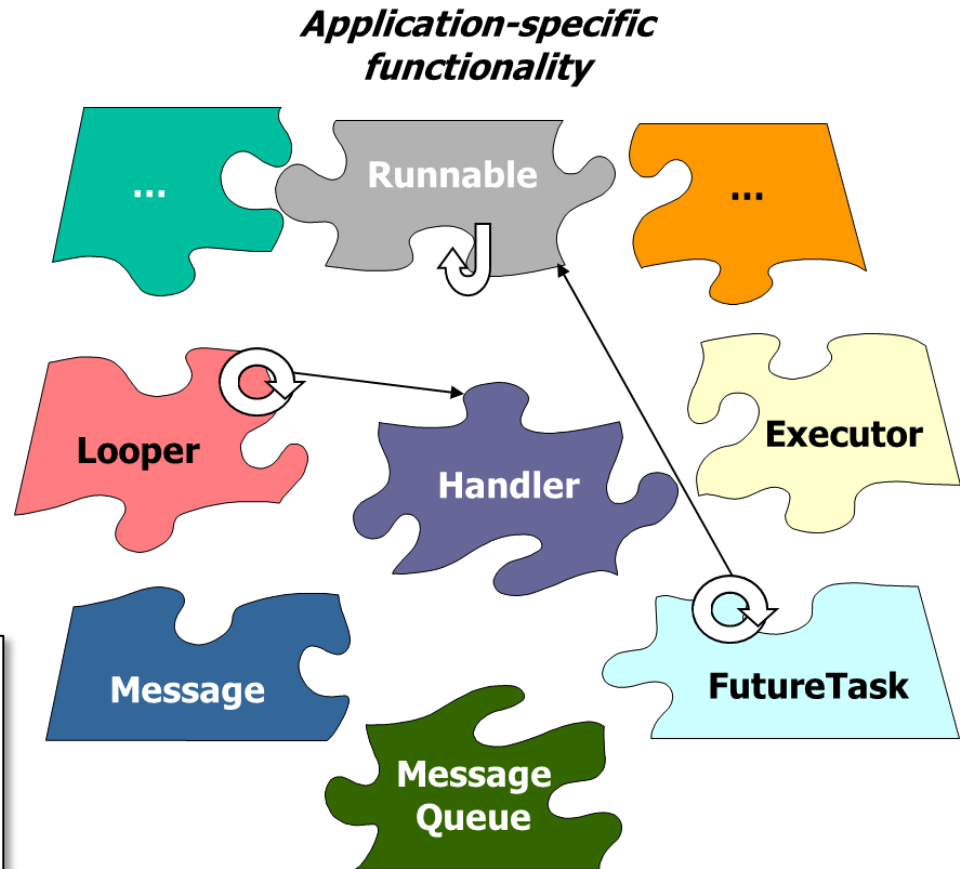
- Key Java parallelism frameworks
- Modern web programming platforms



See spring.io/projects/spring-boot

Summary of Course Contents

- Key Java parallelism frameworks
- Modern web programming platforms
- Patterns for parallel programming



See www.dre.Vanderbilt.edu/~Schmidt/POSA

Summary of Course Contents

- Key Java parallelism frameworks
- Modern web programming platforms
- Patterns for parallel programming
- We assume you know (or can quickly learn) modern Java, Android, & Git



Structure of the Lecture Material

Structure of the Lecture Material

- This course has three main modules

Section	Topics
Java Parallelism	<ul style="list-style-type: none">• Coverage of modern Java parallelism frameworks, e.g.<ul style="list-style-type: none">• Java sequential & parallel streams• Java completable futures• Reactive streams (e.g., RxJava & Project Reactor)

Structure of the Lecture Material

- This course has three main modules

Section	Topics
Java Parallelism	<ul style="list-style-type: none">• Coverage of modern Java parallelism frameworks, e.g.<ul style="list-style-type: none">• Java sequential & parallel streams• Java completable futures• Reactive streams (e.g., RxJava & Project Reactor)
Mobile \leftrightarrow Web Communication	<ul style="list-style-type: none">• Spring WebMVC & WebFlux

Structure of the Lecture Material

- This course has three main modules

Section	Topics
Java Parallelism	<ul style="list-style-type: none">• Coverage of modern Java parallelism frameworks, e.g.<ul style="list-style-type: none">• Java sequential & parallel streams• Java completable futures• Reactive streams (e.g., RxJava & Project Reactor)
Mobile ⇔ Web Communication	<ul style="list-style-type: none">• Spring WebMVC & WebFlux
Software Patterns	<ul style="list-style-type: none">• Parallel programming & communication patterns

Structure of the Lecture Material

- This course has three main modules
 - Each module is composed of lessons



Structure of the Lecture Material

- This course has three main modules
 - Each module is composed of lessons
 - Each lesson is composed of parts



Structure of the Lecture Material

- This course has three main modules
 - Each module is composed of lessons
 - Each lesson is composed of parts
 - Each part is a single lecture



Screencasts of each lesson “part” & PDF versions of the slides will be uploaded to www.dre.vanderbilt.edu/~schmidt/cs253#lectures

Structure of the Lecture Material

- This course has three main modules
 - Each module is composed of lessons
 - Each lesson is composed of parts
 - Each part is a single lecture
 - Each part is composed of segments



Structure of the Lecture Material

- There will be bi-weekly quizzes on material covered in the lectures



Structure of the Lecture Material

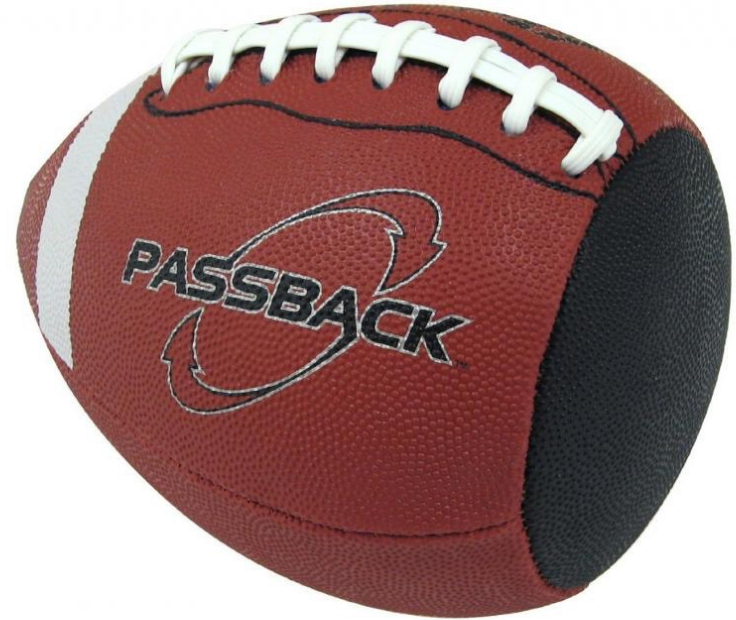
- There will be bi-weekly quizzes on material covered in the lectures
- 1st quiz will be on Wednesday, August 31st



All quizzes are “closed book/note/Internet” & are given on Brightspace

Structure of the Lecture Material

- There will be bi-weekly quizzes on material covered in the lectures
 - 1st quiz will be on Wednesday, August 31st
- We strive to hand back & review quizzes at the start of next class



One of the benefits of a smaller class ;-)

Structure of the Lecture Material

- There will be bi-weekly quizzes on material covered in the lectures
 - 1st quiz will be on Wednesday, August 31st
- We strive to hand back & review quizzes at the start of next class



I recommend that you study for quizzes by reviewing slides & watching screencasts available at www.dre.vanderbilt.edu/~schmidt/cs253#lectures

Structure of the Lecture Material

- There *may* be a cumulative final exam that covers all the lectures
- The focus will be on the last week(s) of the semester



The final exam *may* be held 2 to 5pm, Friday, December 16th via Brightspace

CS 253: Parallel Functional Programming with Java, Android, & Spring WebFlux: Overview (Part 1)