

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

Douglas C. Schmidt

d.schmidt@vanderbilt.edu

www.dre.vanderbilt.edu/~schmidt



Professor of Computer Science

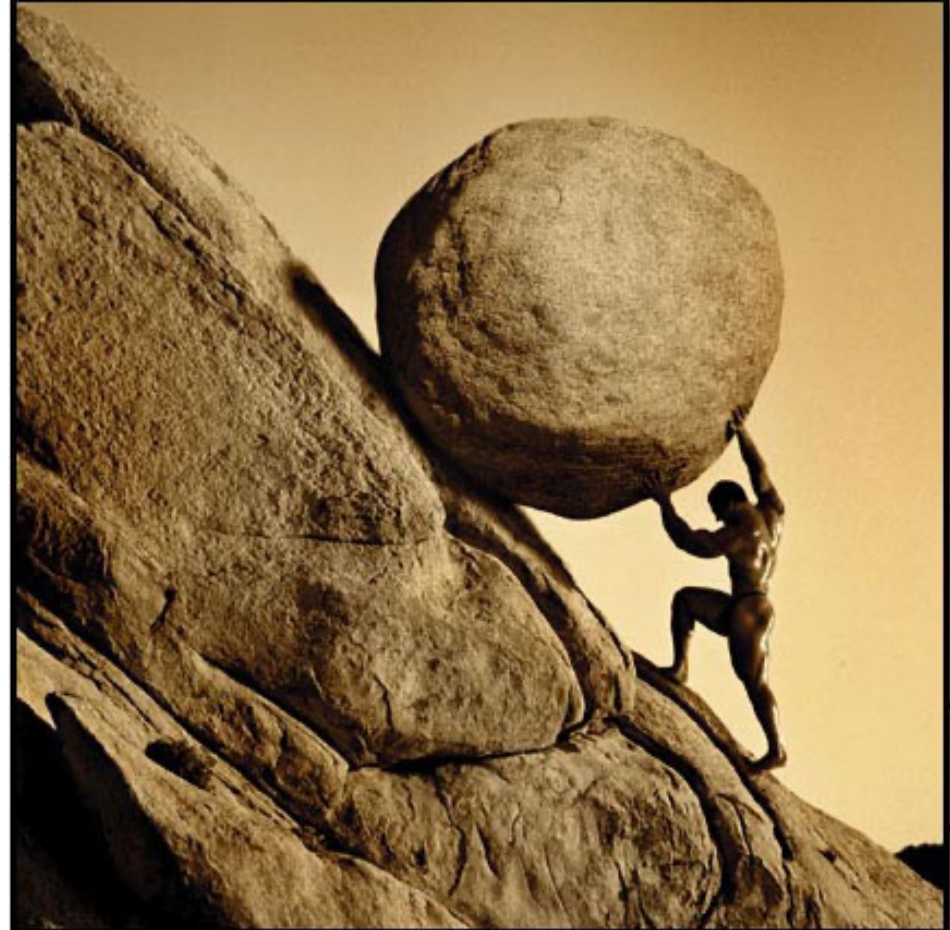
**Institute for Software
Integrated Systems**

**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



Work Request



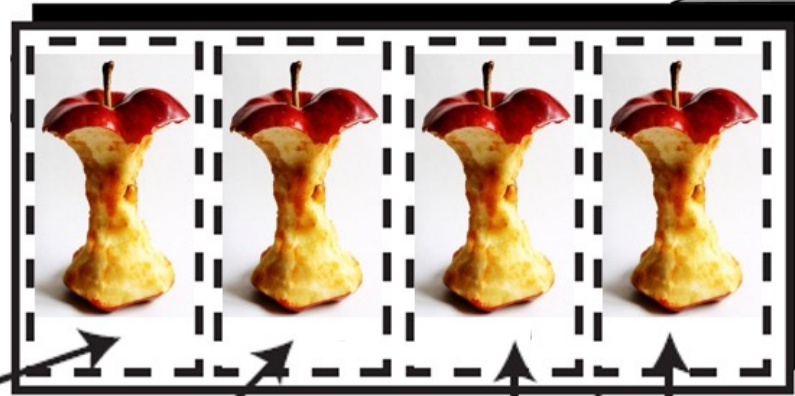
Work Request



Work Request



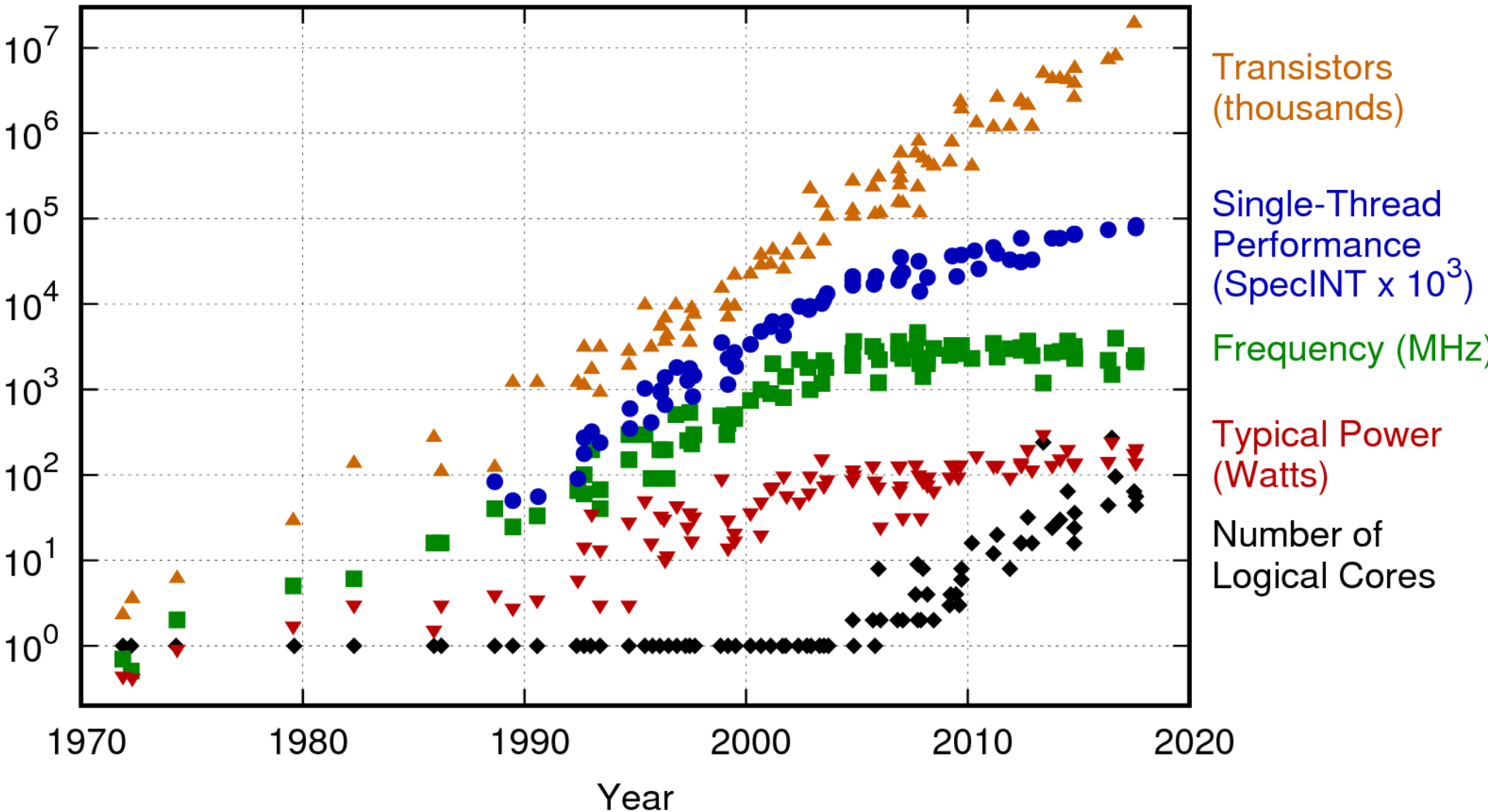
Work Request



Cloud Server

Course Philosophy

- Demand is driven by software/hardware infrastructure advances



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

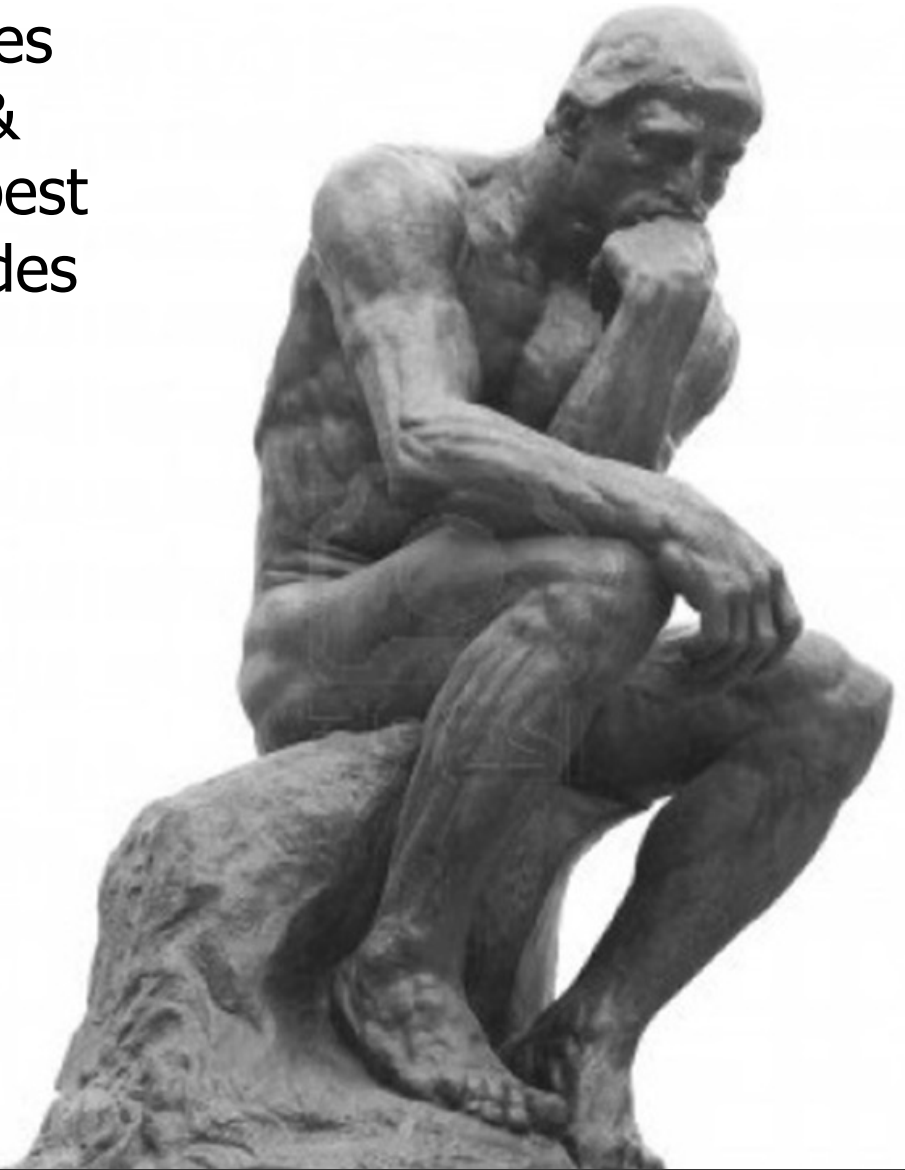
Course Philosophy

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



Course Philosophy

- Effective techniques & practices for writing parallel programs & mobile apps are *not* learned best 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

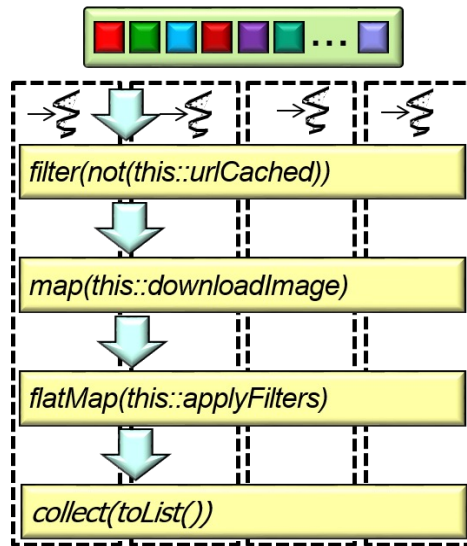
Summary of Course Contents

- Key Java parallelism frameworks

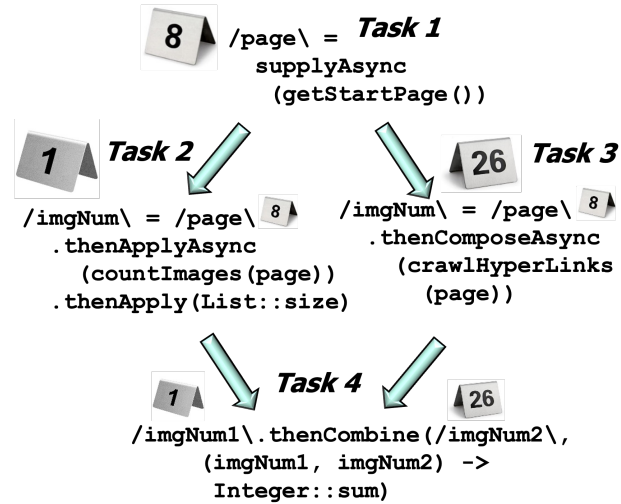
Fork-Join Pool



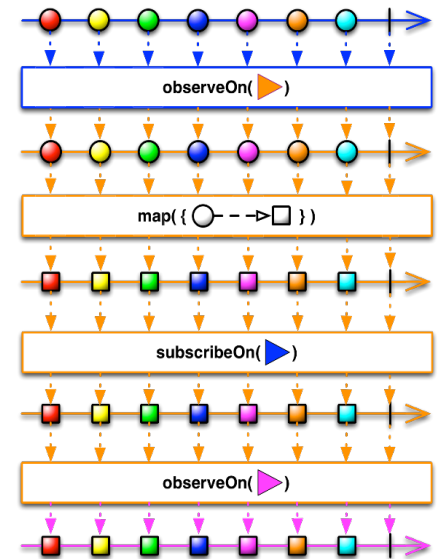
Parallel Streams



Completable Futures



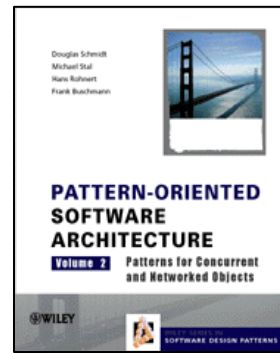
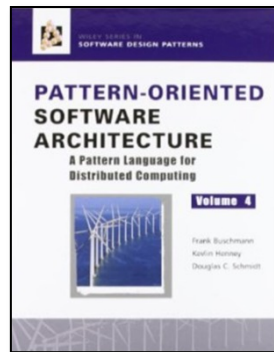
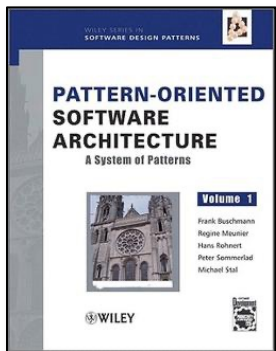
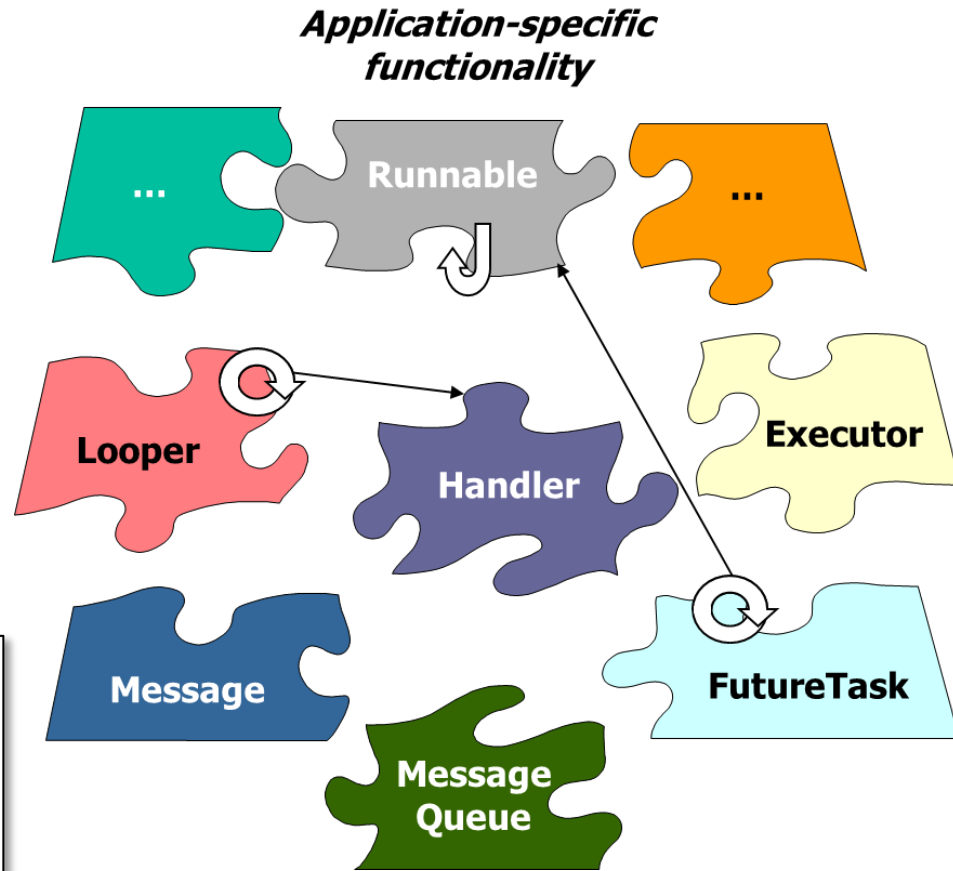
Reactive Streams



Assumes knowledge of Java object-oriented & functional language features

Summary of Course Contents

- Key Java parallelism frameworks
- Patterns for parallel programming



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

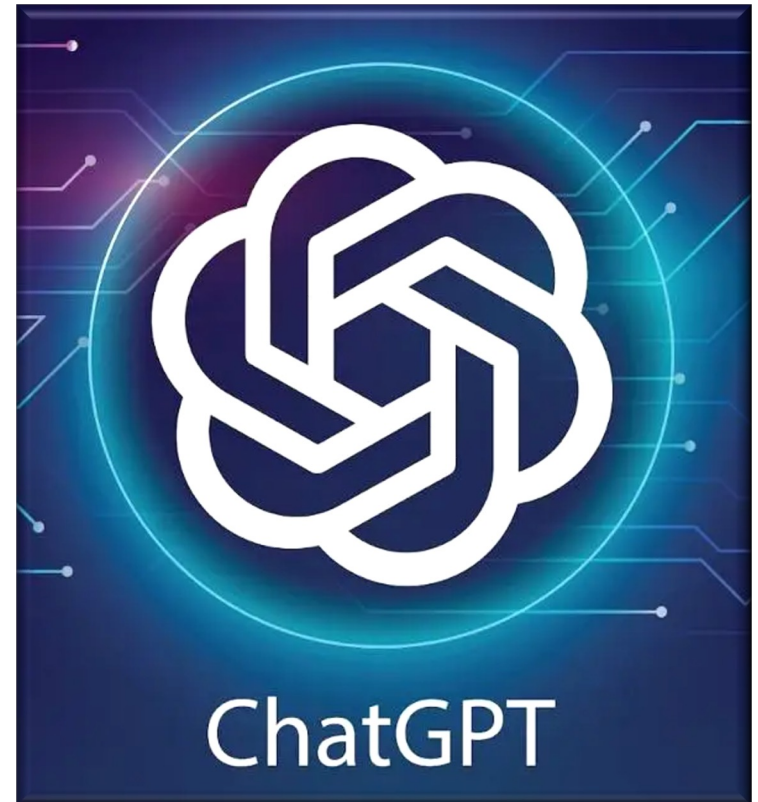
Summary of Course Contents

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



Summary of Course Contents

- Key Java parallelism frameworks
- Patterns for parallel programming
- We assume you know (or can quickly learn) modern Java, Android, & Git
- We'll apply large-language models (LLMs) throughout this course where appropriate



See cdn.vanderbilt.edu/vu-URL/wp-content/uploads/sites/241/2023/08/16143452/Vanderbilt-University-Academic-Affairs-Guidance-for-Artificial-Intelligence.pdf

Summary of Course Contents

- Key Java parallelism frameworks
- Patterns for parallel programming
- We assume you know (or can quickly learn) modern Java, Android, & Git
- We'll apply large-language models (LLMs) throughout this course where appropriate
 - LLMs are having a massive impact on education & workforce productivity

Assuring the Future of Software Engineering & AI Engineering

Douglas C. Schmidt

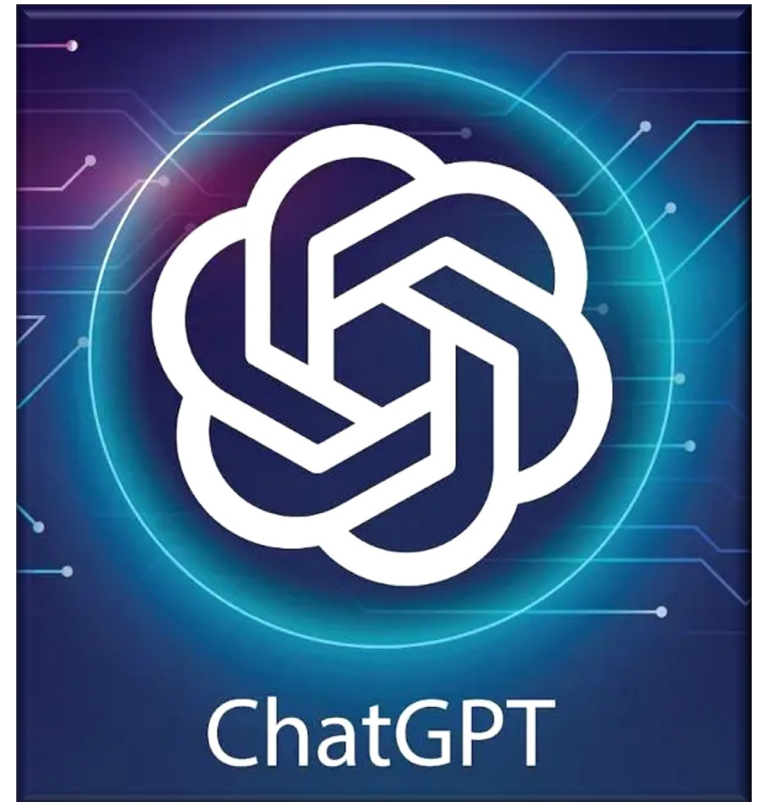
d.schmidt@vanderbilt.edu

www.dre.vanderbilt.edu/~schmidt

Professor of Computer Science

**Institute for Software
Integrated Systems**

**Vanderbilt University
Nashville, Tennessee, USA**

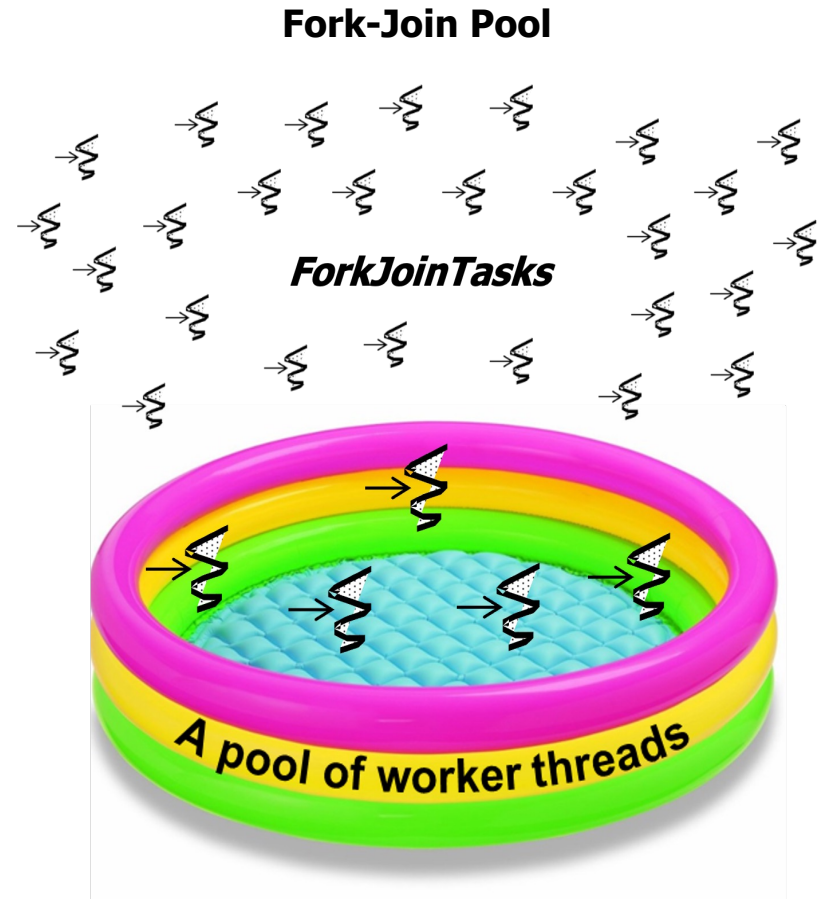


See www.youtube.com/watch?v=fDnNT7z9aT8

Structure of the Lecture Material

Structure of the Lecture Material

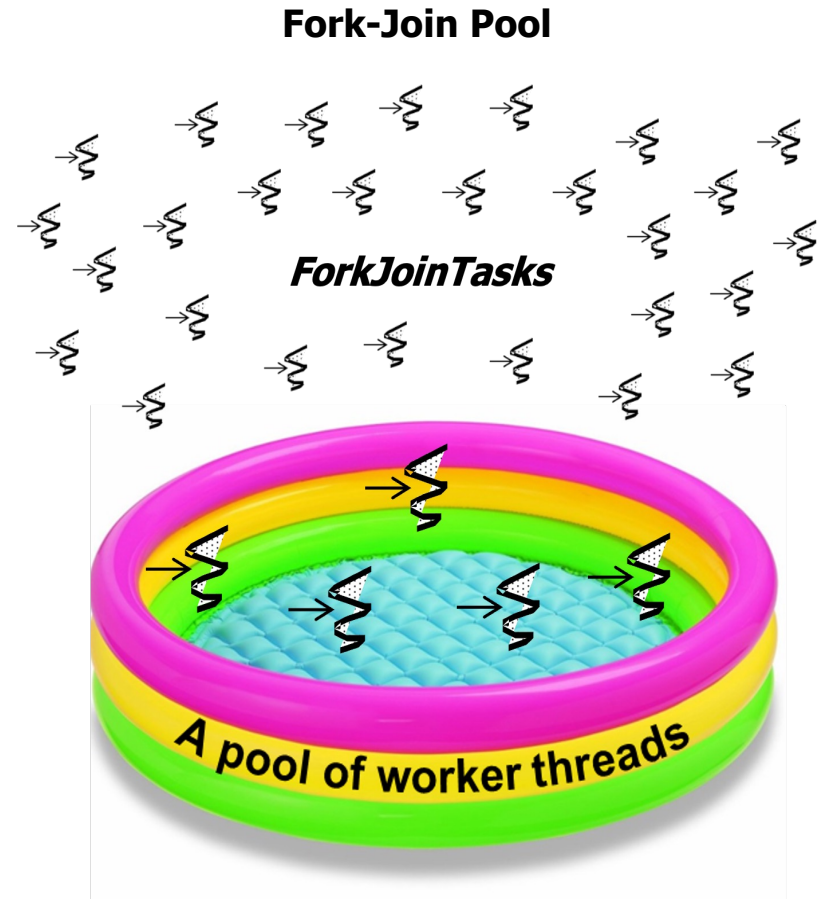
- This course has four main modules
 - Java fork-join framework
 - Provides a parallel execution engine designed to recursively split tasks into smaller subtasks & then combine their results



See www.baeldung.com/java-fork-join

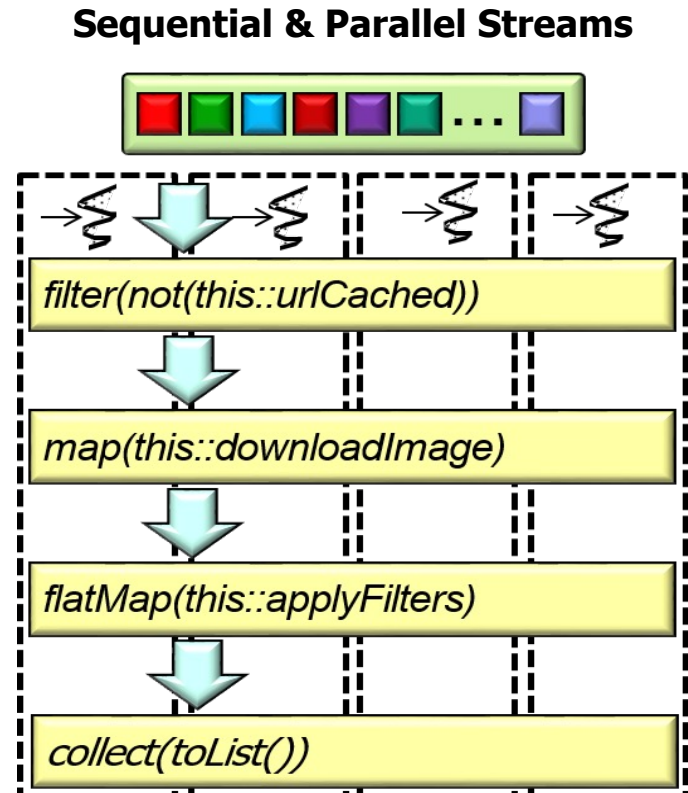
Structure of the Lecture Material

- This course has four main modules
 - Java fork-join framework
 - Provides a parallel execution engine designed to recursively split tasks into smaller subtasks & then combine their results
 - Focus is on optimizing multi-core processor utilization



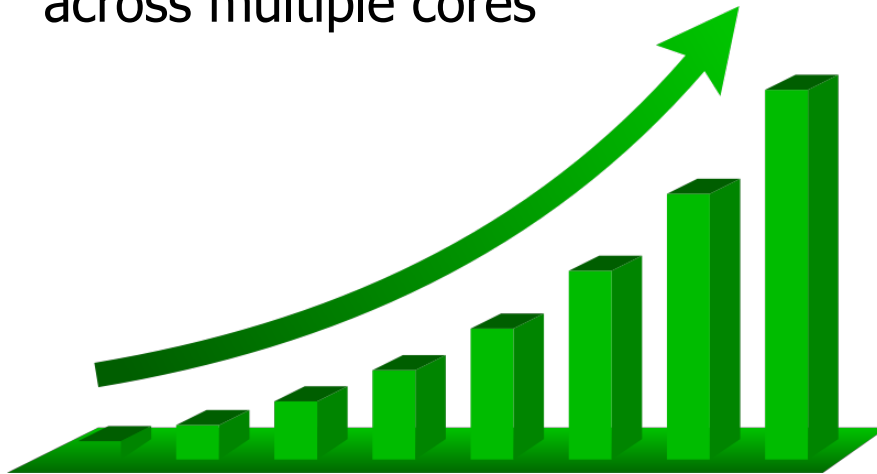
Structure of the Lecture Material

- This course has four main modules
 - Java fork-join framework
 - Sequential & parallel streams framework
 - Provides mechanisms to process a pipeline of data in either a single-threaded manner or by leveraging parallelism to scale performance

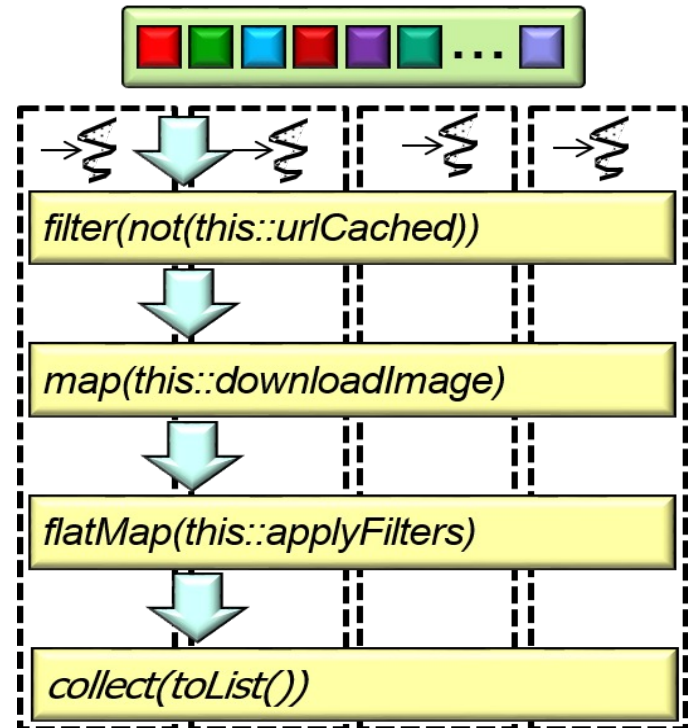


Structure of the Lecture Material

- This course has four main modules
 - Java fork-join framework
 - Sequential & parallel streams framework
 - Provides mechanisms to process a pipeline of data in either a single-threaded manner or by leveraging parallelism to scale performance
 - Focus is on transparent scalability across multiple cores



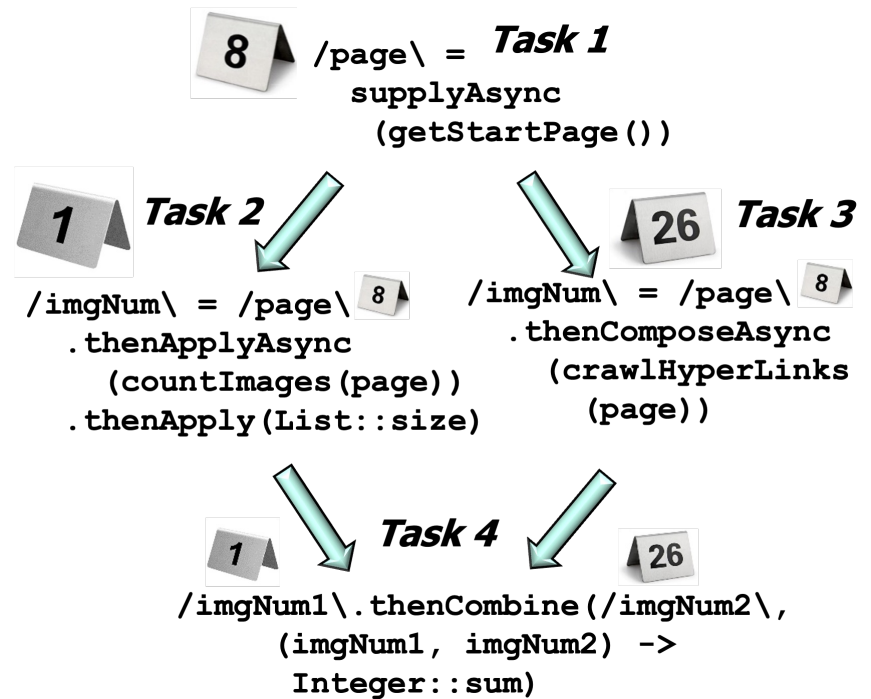
Sequential & Parallel Streams



Structure of the Lecture Material

- This course has four main modules
 - Java fork-join framework
 - Sequential & parallel streams framework
- Completable futures framework
 - Enables tasks to execute asynchronously & provides a means to combine, chain, & handle task results & exceptions

Completable Futures

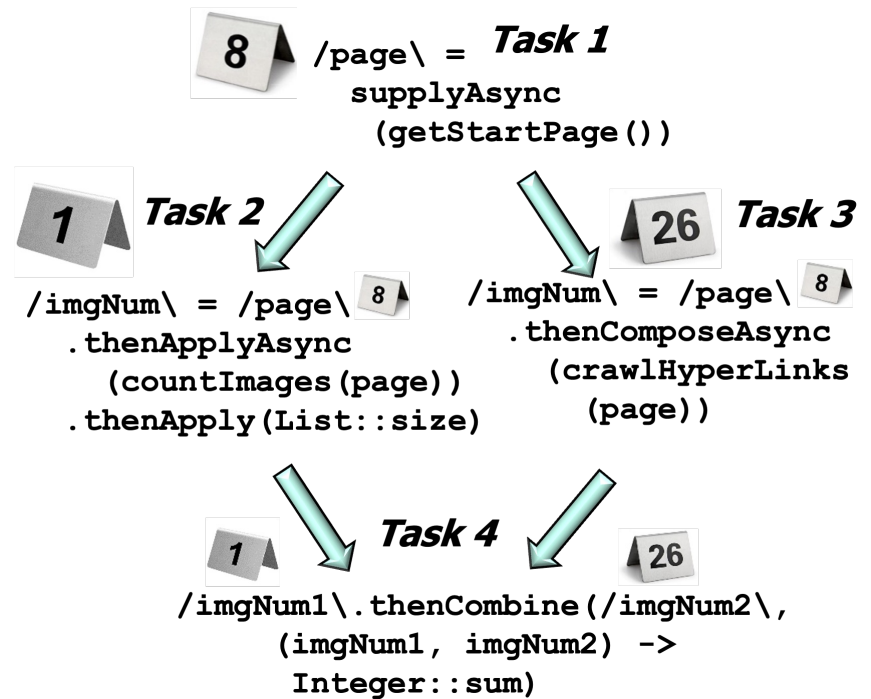


See www.baeldung.com/java-completablefuture

Structure of the Lecture Material

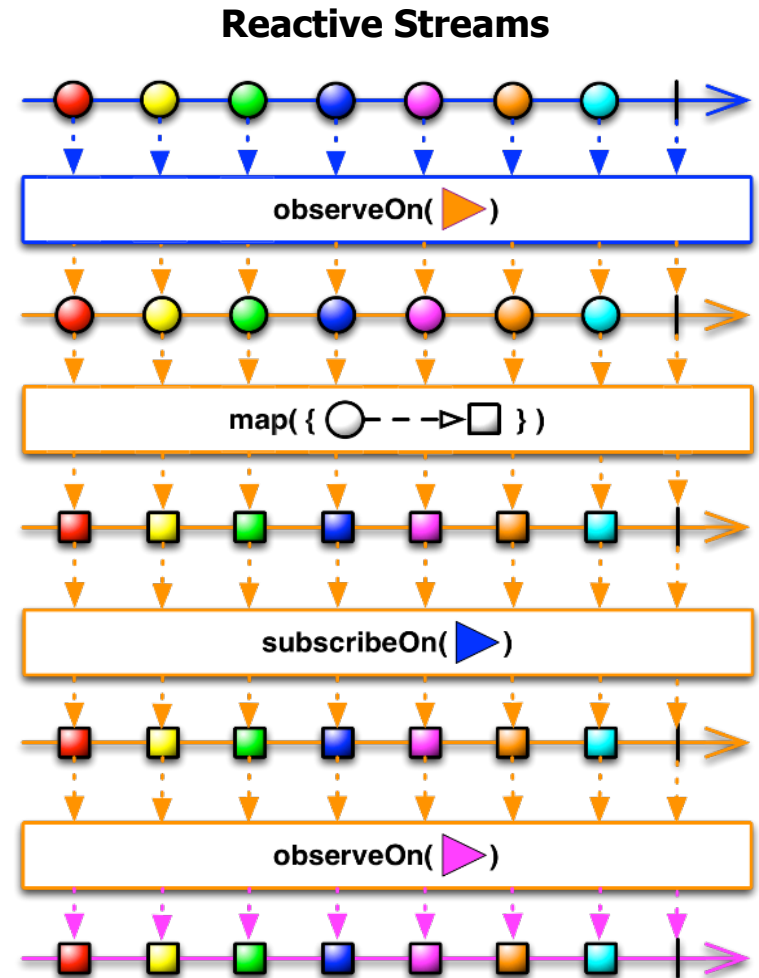
- This course has four main modules
 - Java fork-join framework
 - Sequential & parallel streams framework
- Completable futures framework
 - Enables tasks to execute asynchronously & provides a means to combine, chain, & handle task results & exceptions
 - Focus is on executing tasks concurrently to leverage multi-core processors for enhanced performance & non-blocking operations

Completable Futures



Structure of the Lecture Material

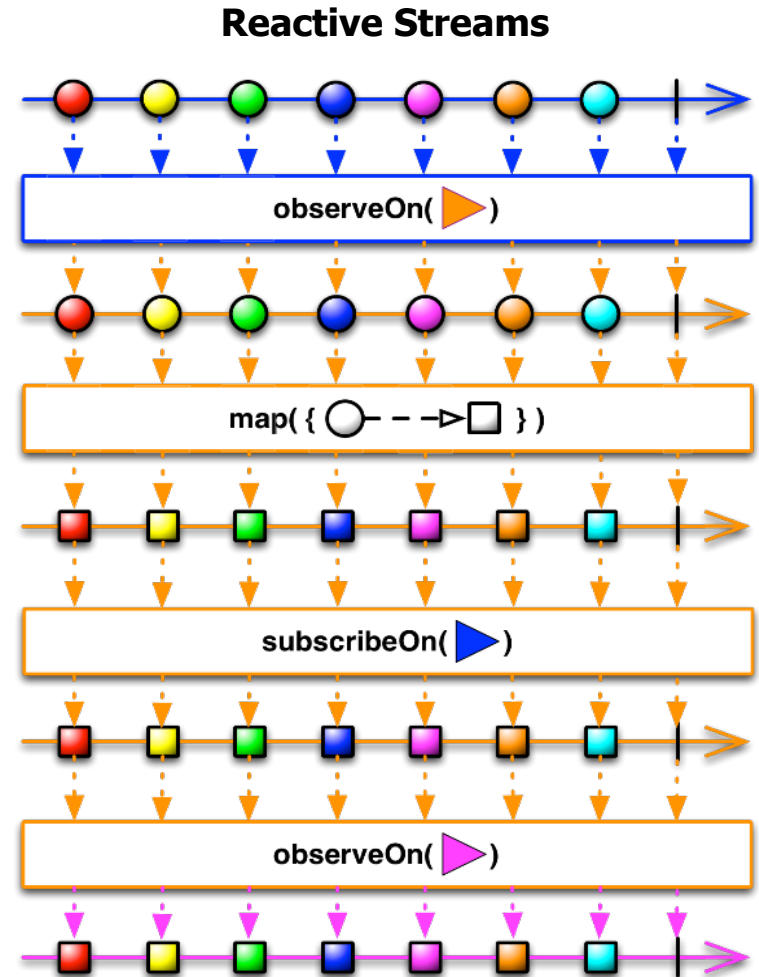
- This course has four main modules
 - Java fork-join framework
 - Sequential & parallel streams framework
 - Completable futures framework
 - Reactive streams framework
 - Facilitates asynchronous data processing & flow control



See dzone.com/articles/rxjava-idiomatic-concurrency-flatmap-vs-parallel

Structure of the Lecture Material

- This course has four main modules
 - Java fork-join framework
 - Sequential & parallel streams framework
 - Completable futures framework
 - Reactive streams framework
 - Facilitates asynchronous data processing & flow control
 - Focus is on parallelism to handle large streams of data efficiently across multiple cores



Structure of the Lecture Material

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



Structure of the Lecture Material

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



Structure of the Lecture Material

- This course has four 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 four 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



We'll often interactively cover discussion questions at the end of each part

Structure of the Lecture Material

- This course has four 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

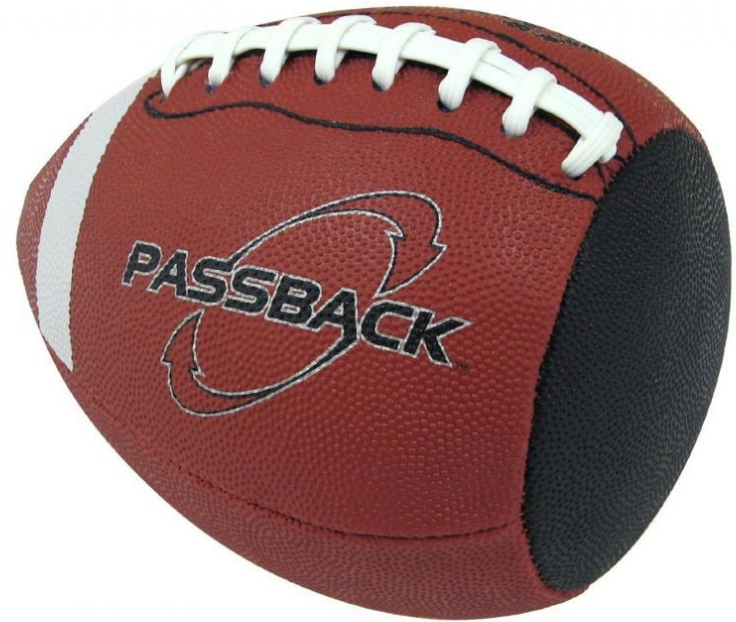
- There will be bi-weekly quizzes on material covered in the lectures
- 1st quiz will be on Wednesday, August 30th



All quizzes are "closed book/note/Internet/ChatGPT/etc." & are given in 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 30th
 - 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 30th
 - 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 will be bi-weekly quizzes on material covered in the lectures
 - 1st quiz will be on Wednesday, August 30th
- We strive to hand back & review quizzes at the start of next class
 - If you don't attend the next class & don't get your quiz you will be penalized 50%



See www.dre.vanderbilt.edu/~schmidt/cs253/work-summary.html#quizzes

Structure of the Lecture Material

- There will be bi-weekly quizzes on material covered in the lectures
 - 1st quiz will be on Wednesday, August 30th
- We strive to hand back & review quizzes at the start of next class
 - If you don't attend the next class & don't get your quiz you will be penalized 50%
 - Likewise, if you just show up for the quiz & don't attend class you'll be penalized 50%



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 9am to noon, Saturday, December 16th in class

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