CS 5254: Concurrent Object-Oriented & Functional Programming

Course Overview (Part 1)

Douglas C. Schmidt
d.schmidt@vanderbilt.edu
www.dre.vanderbilt.edu/~schmidt

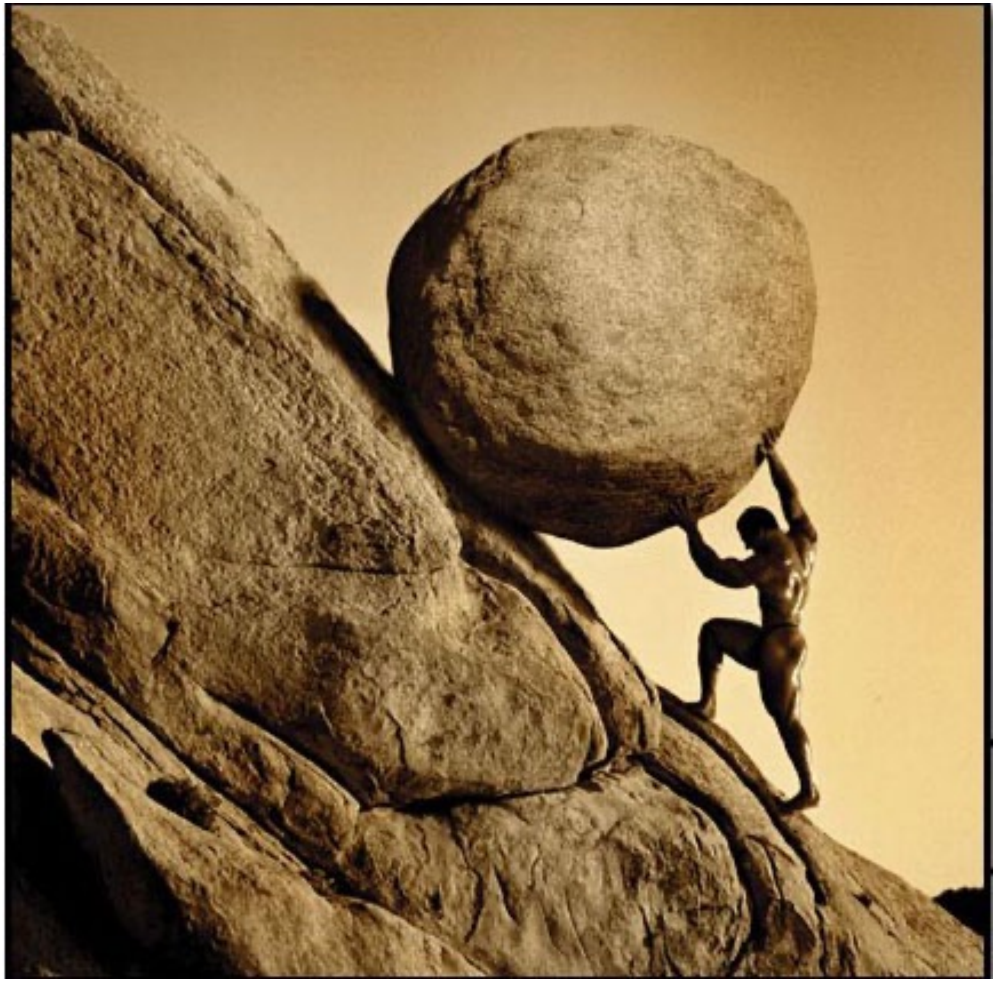
Professor of Computer Science

Institute for Software
Integrated Systems

Vanderbilt University
Nashville, Tennessee, USA

Learning Objectives in this Lesson

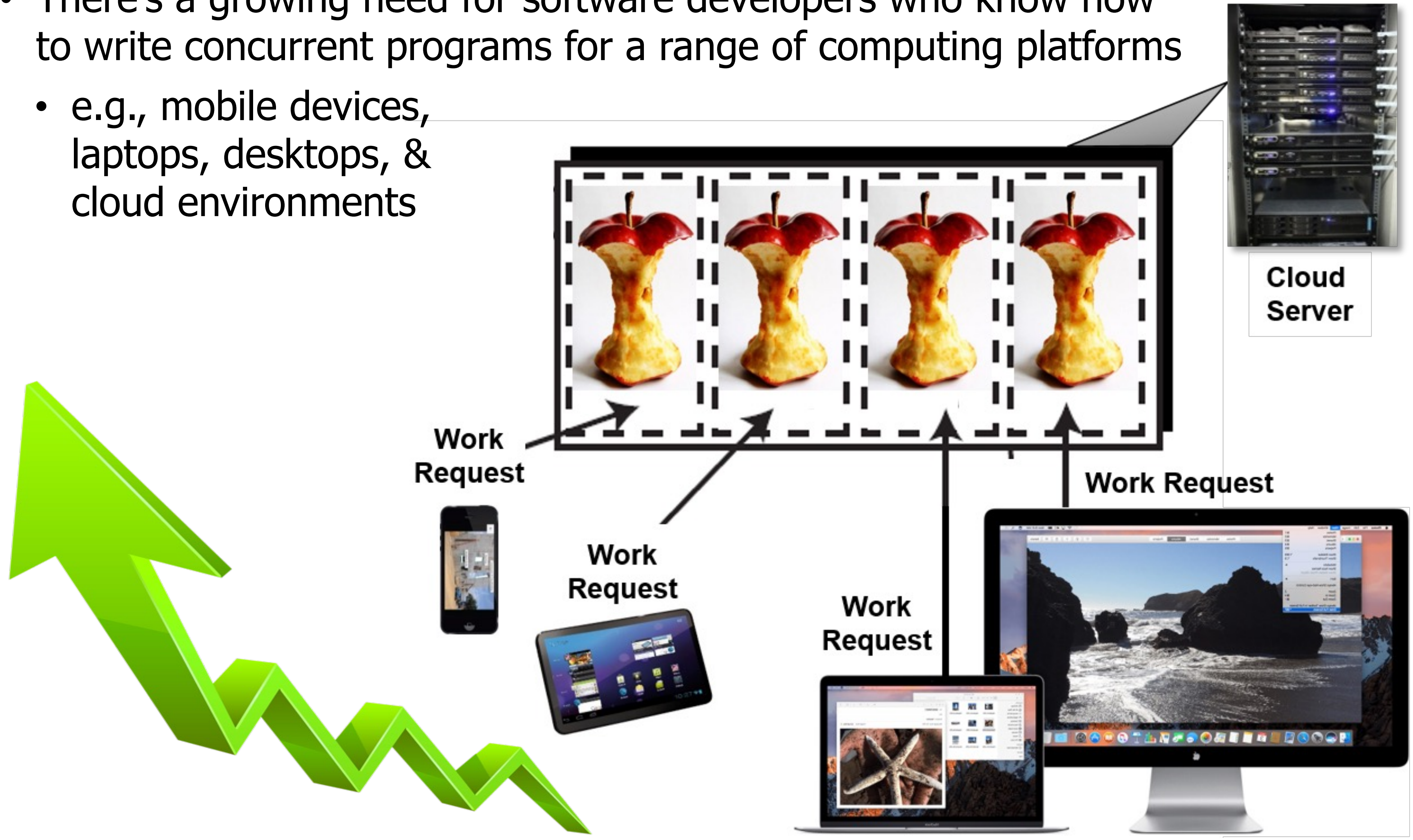
- Understand the course topics & logistics
  - Course philosophy
  - Course contents
  - Structure of the lecture material



Course Philosophy

Course Philosophy

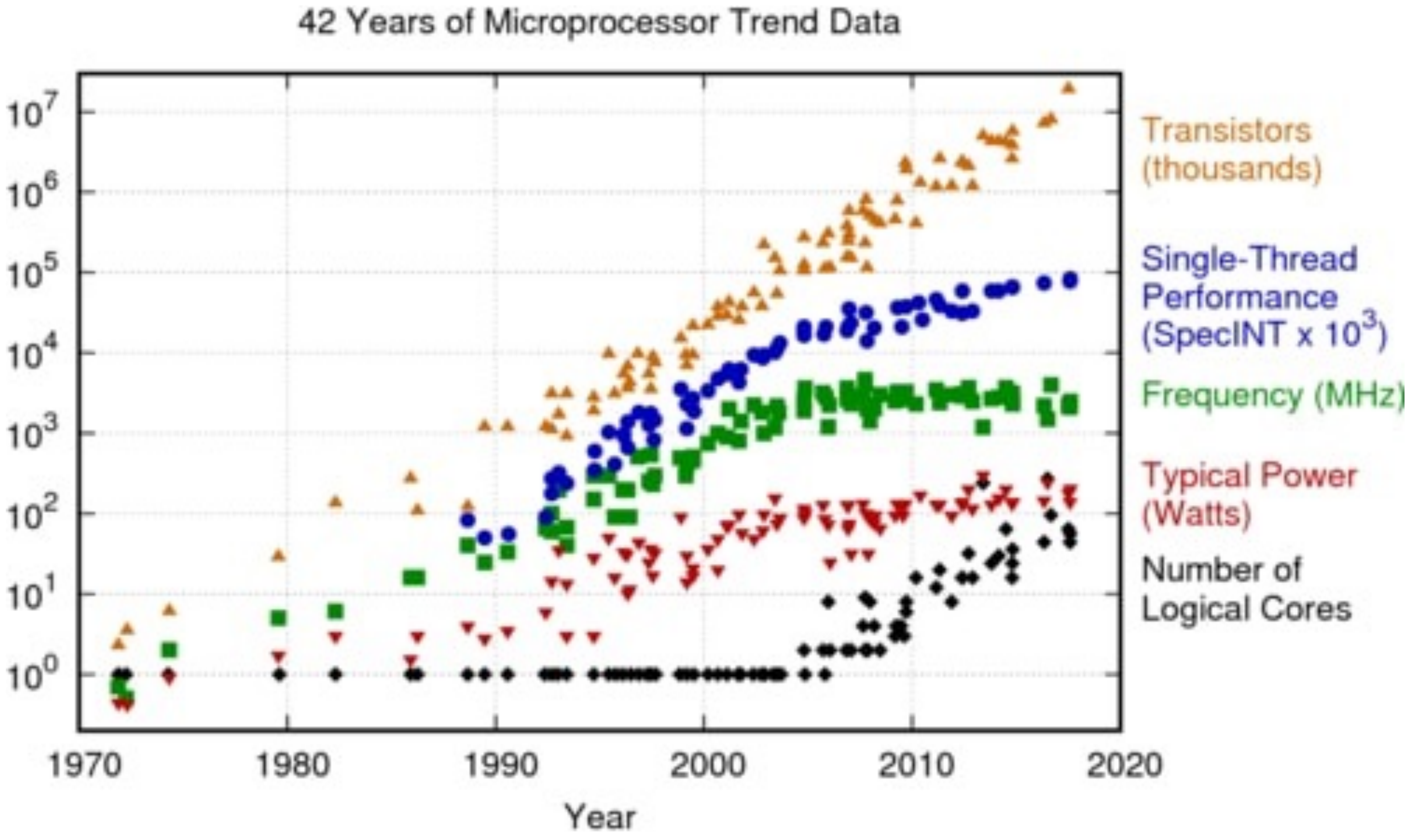
- There’s a growing need for software developers who know how to write concurrent programs for a range of computing platforms
    - e.g., mobile devices, laptops, desktops, & cloud environments



The topics covered in this course apply to many platforms, not just Android

Course Philosophy

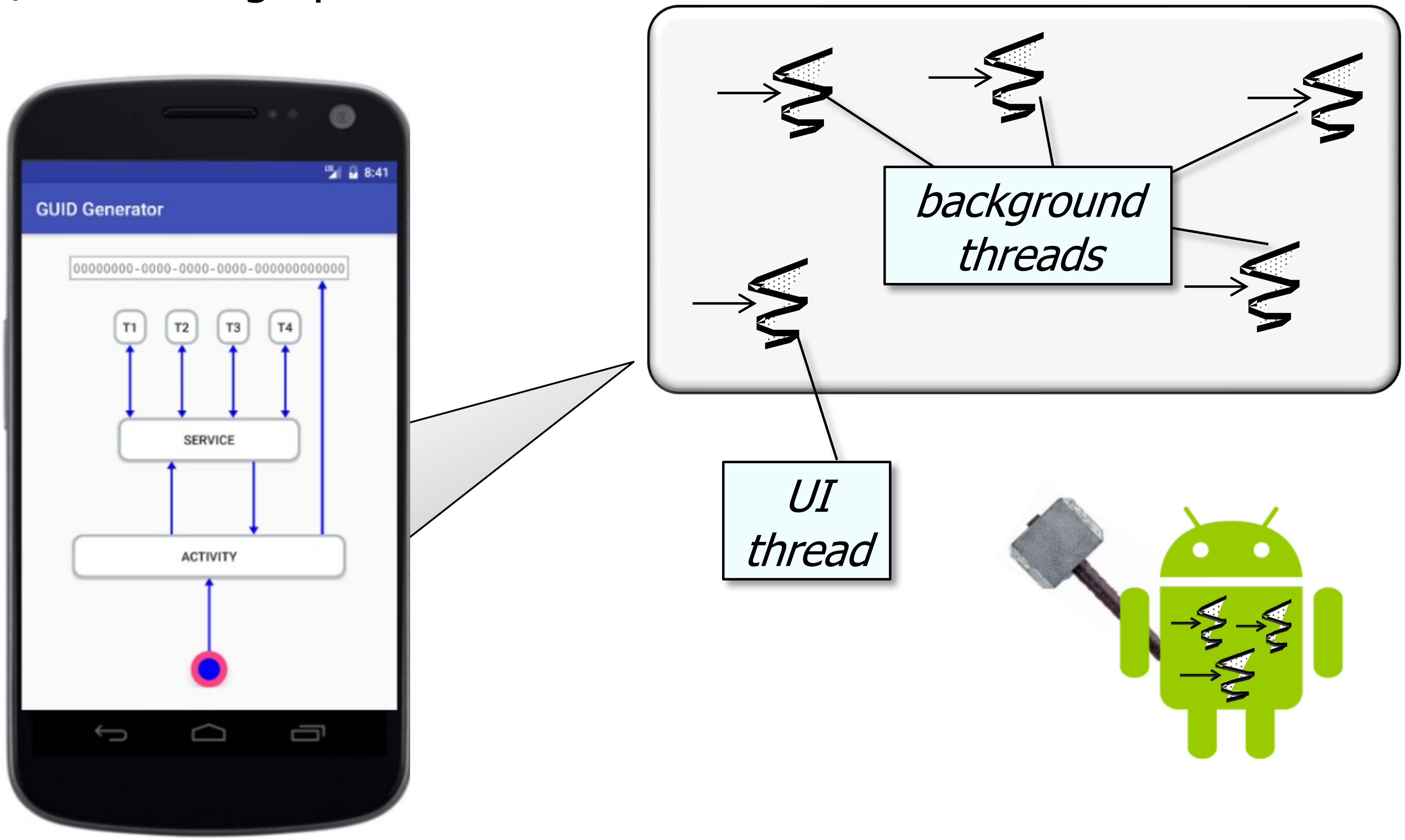
- This demand is driven by advances in software & hardware infrastructure
  - e.g., multi-core & many core processors, mass storage, ubiquitous network connectivity, & commodity hardware & software platforms



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

Course Philosophy

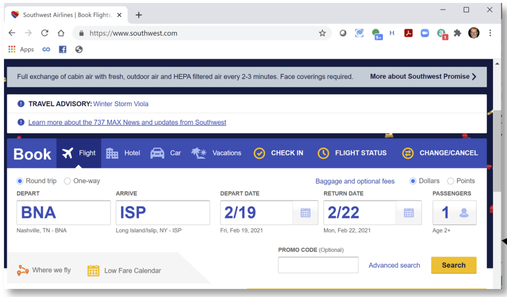
- Concurrency is commonly used in mobile devices to offload work from the user interface (UI) thread to background thread(s) that perform long-running and/or blocking operations



See developer.android.com/training/multiple-threads/communicate-ui.html

Course Philosophy

- Concurrency is also commonly used to process client requests in multi-core backend computing environments, such as data centers or cloud servers



Southwest Airlines | Book Flights

https://www.southwest.com

Apps

Full exchange of cabin air with fresh, outdoor air and HEPA filtered air every 2-3 minutes. Face coverings required. **More about Southwest Promise ›**

TRAVEL ADVISORY: Winter Storm Viola

Learn more about the 737 MAX News and updates from Southwest

Book Flight Hotel Car Vacations CHECK IN FLIGHT STATUS CHANGE/CANCEL

● Round trip ○ One-way   Baggage and optional fees   ● Dollars ○ Points

DEPART	ARRIVE	DEPART DATE	RETURN DATE	PASSENGERS
BNA	ISP	2/19	2/22	1
Nashville, TN - BNA	Long Island/Islip, NY - ISP	Fri, Feb 19, 2021	Mon, Feb 22, 2021	Age 2+

PROMO CODE (Optional)

Advanced search   Search

Where we fly   Low Fare Calendar

Clients

Back   Boarding Pass

SOUTHWEST

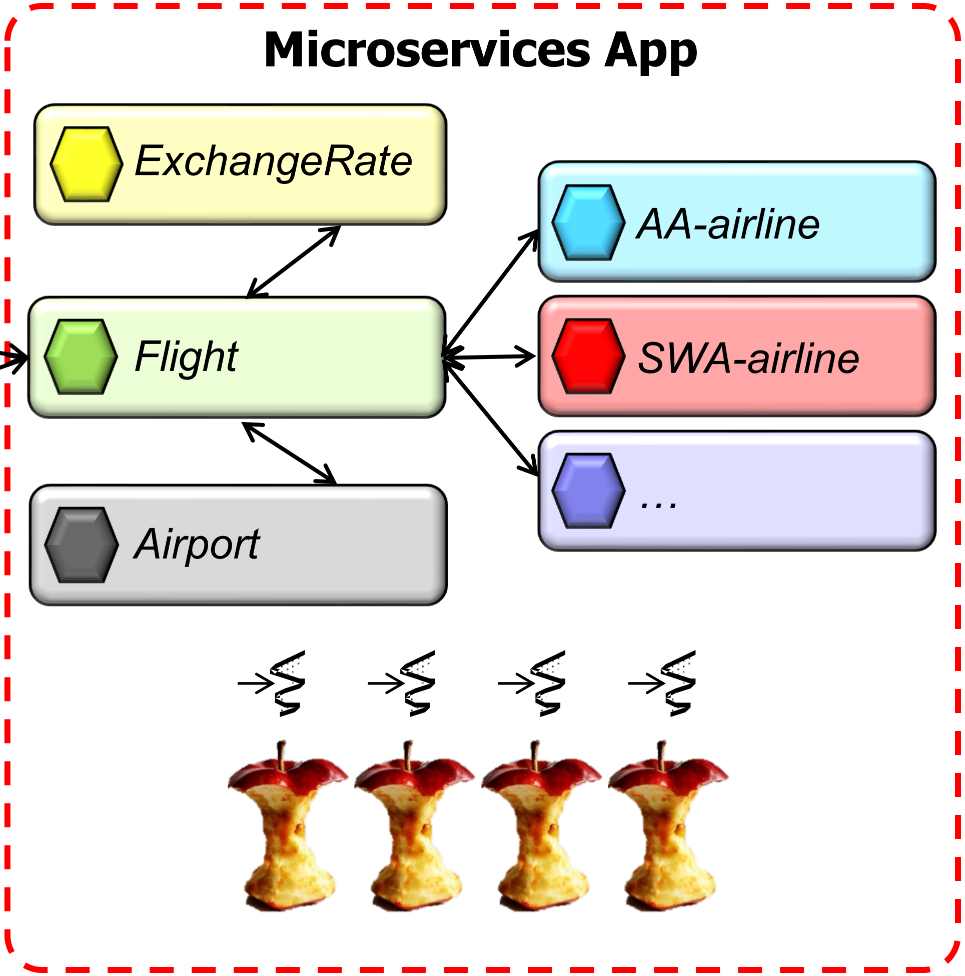
TSA Pre✓

Passenger Kelly / Gary

Boarding Group **A 15** Boarding Position

Flight 531

Austin, TX (AUS) - Dallas, TX (DAL)



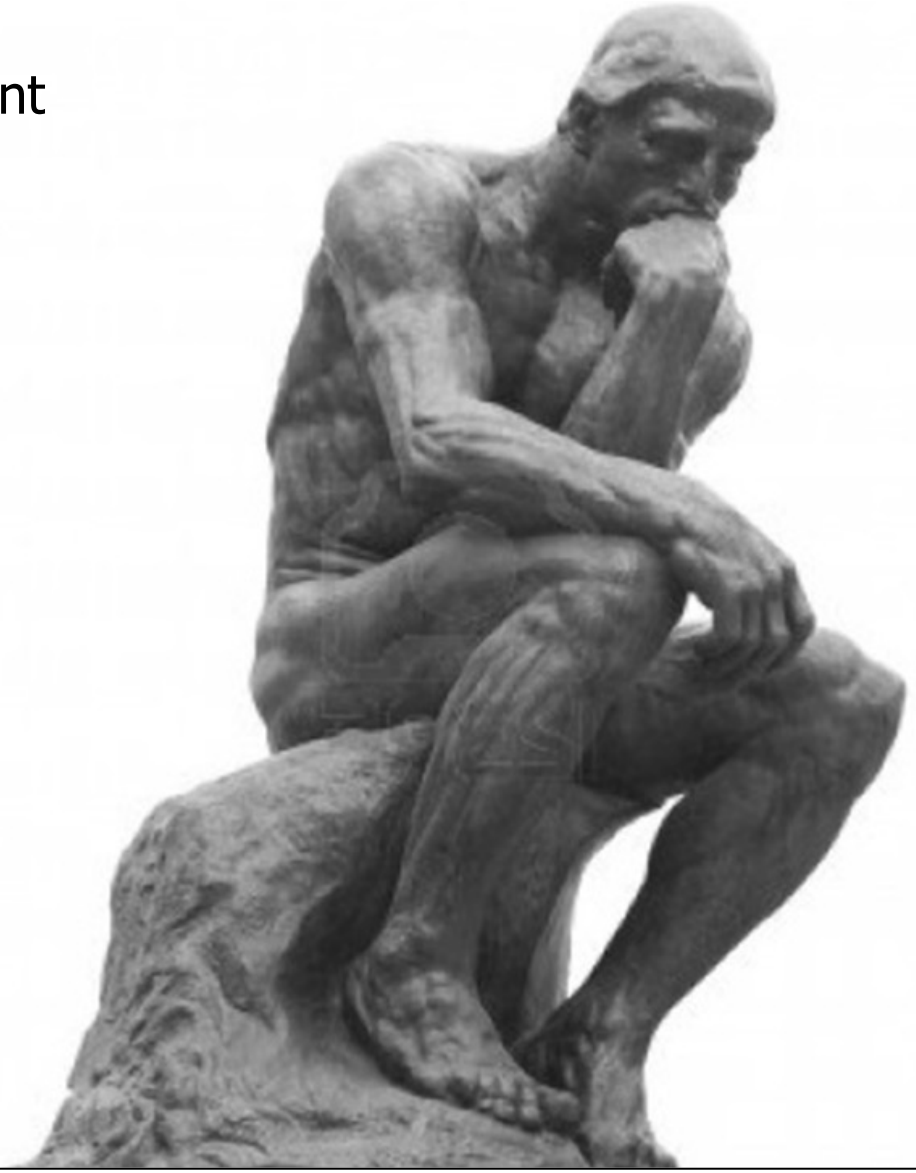
See www.dre.vanderbilt.edu/~schmidt/cs891

Course Philosophy

- Effective techniques & practices for designing & programming concurrent (mobile) apps are *not* best learned through generalities & platitudes

Course Philosophy

- Effective techniques & practices for designing & programming concurrent (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 concurrent programs can be made
  - *easier* to write & read,
  - *easier* to maintain & modify,
  - *more* efficient & resilient

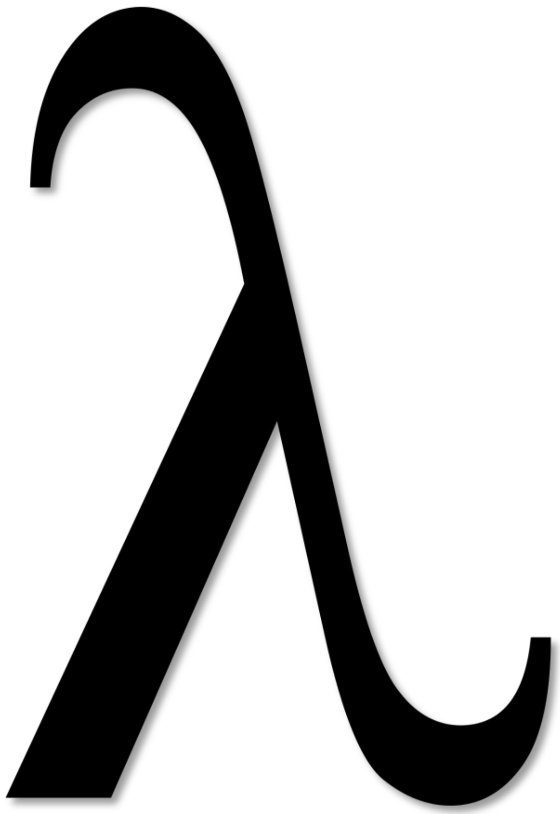
  by applying time-proven software patterns, as well as object-oriented, & functional design & programming techniques

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

Summary of the Course Contents

Summary of Course Contents

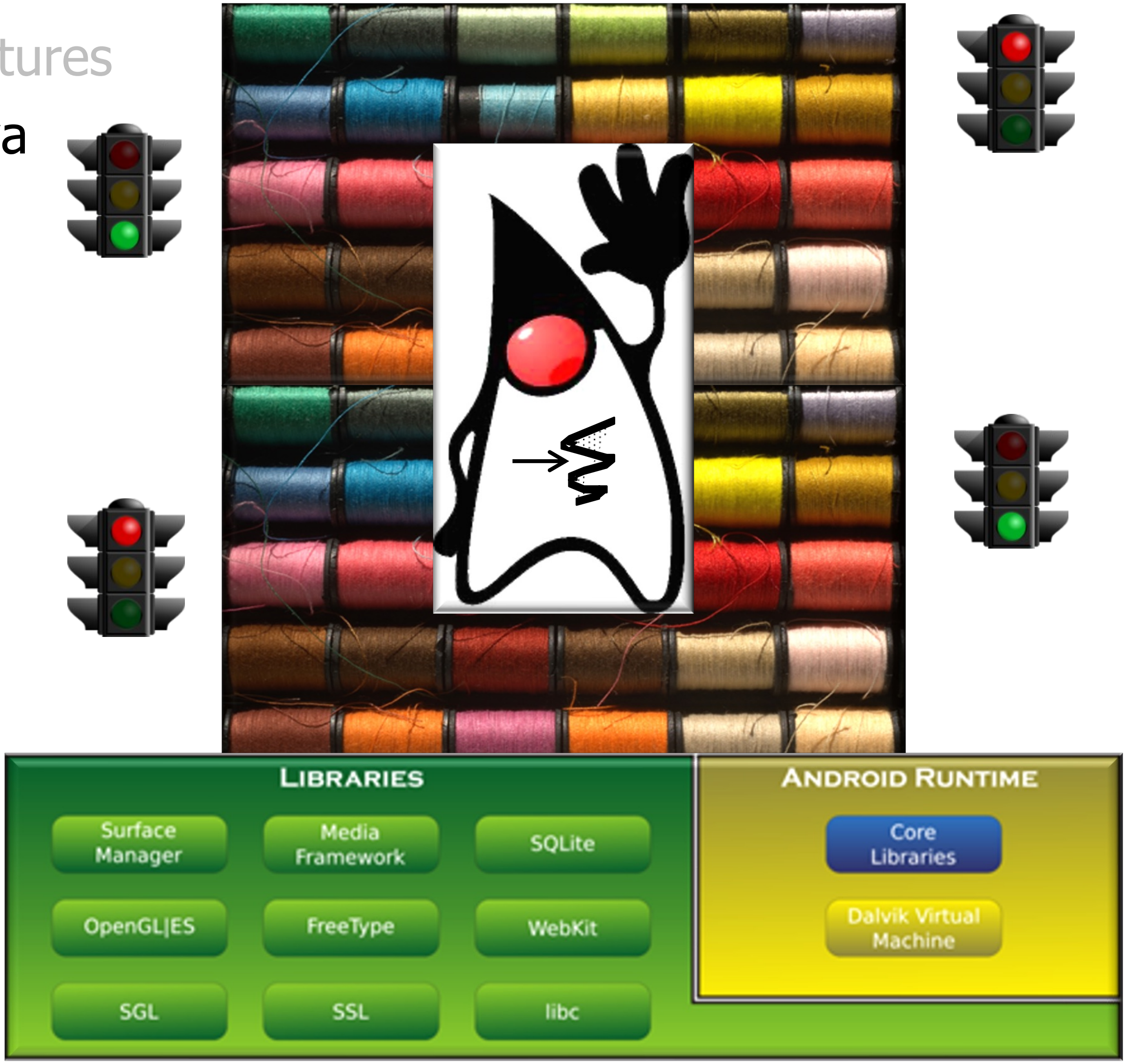
- Foundational Java functional programming concepts & features
  - e.g., lambda expressions, method references, & functional interfaces



These features were added in Java 8 & expanded thereafter

Summary of Course Contents

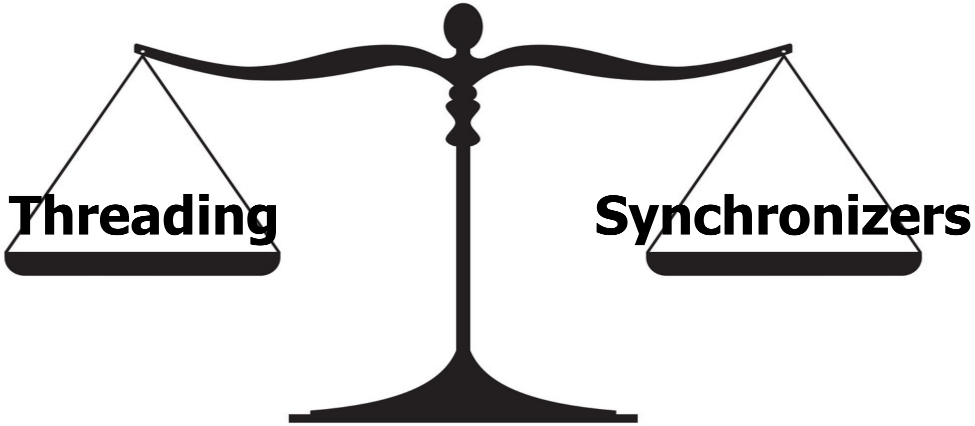
- Foundational Java functional programming concepts & features
- Coverage of foundational Java concurrency mechanisms
  - e.g., Java threading & synchronizer mechanisms



See www.orctom.com/2017/04/21/java-locks-and-concurrency

Summary of Course Contents

- Foundational Java functional programming concepts & features
- Coverage of foundational Java concurrency mechanisms
  - e.g., Java threading & synchronizer mechanisms



Threading mechanisms

- Thread
- Runnable
- Executor
- ExecutorService
- ScheduledExecutorService
- ExecutorCompletionService
- Future
- FutureTask
- ThreadPoolExecutor
- ForkJoinPool

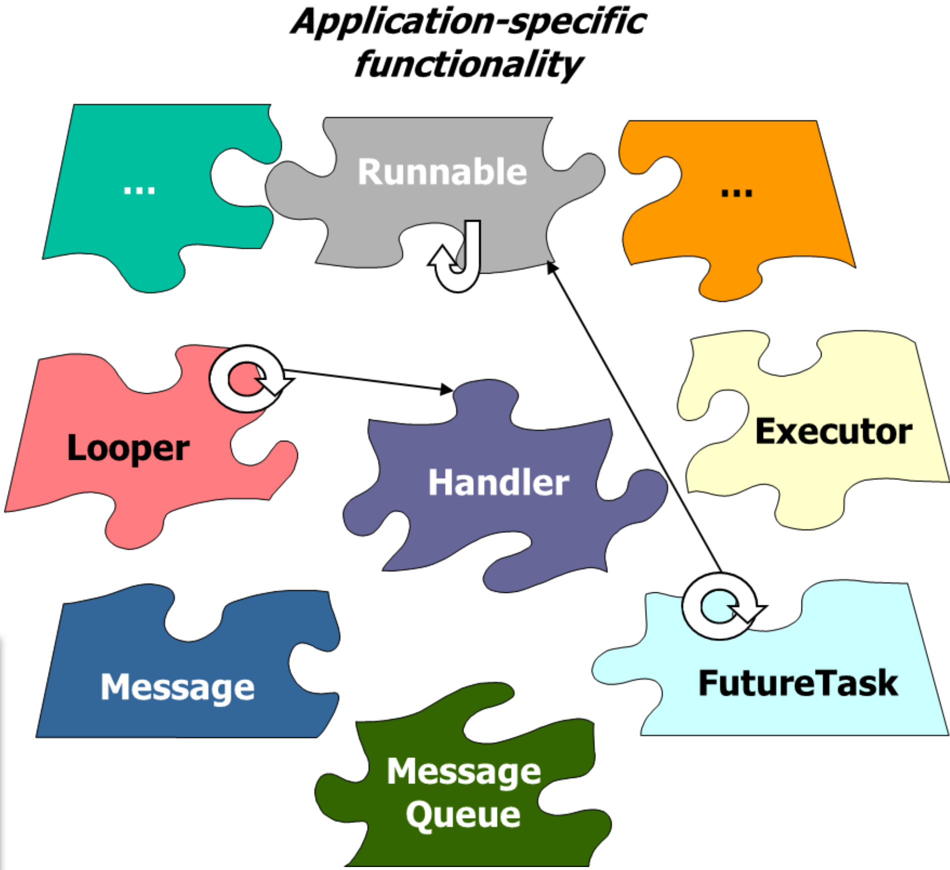
Synchronizer mechanisms

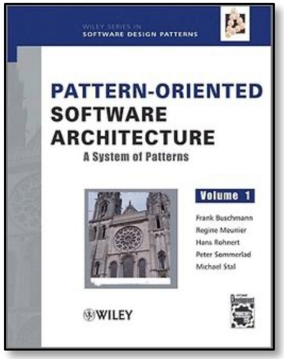
- Synchronized methods & statements
- Notification methods
- ReentrantLock
- ReentrantReadWriteLock
- StampedLock
- ConditionObject
- Semaphore
- CountDownLatch
- CyclicBarrier
- Phaser

We provide roughly equal focus on Java synchronization mechanisms & Java threading mechanisms in this course

Summary of Course Contents

- Foundational Java functional programming concepts & features
- Coverage of foundational Java concurrency mechanisms
- Patterns/frameworks for concurrent programming




WILEY SERIES IN SOFTWARE DESIGN PATTERNS

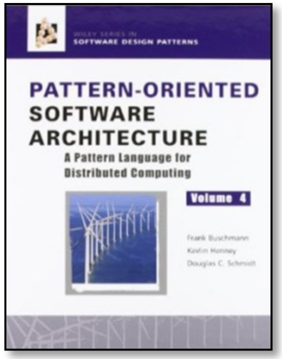
PATTERN-ORIENTED SOFTWARE ARCHITECTURE

A System of Patterns

Volume 1

Frank Buschmann
Regine Meunier
Hans Rohnert
Peter Sommerlad
Michael Stal

WILEY

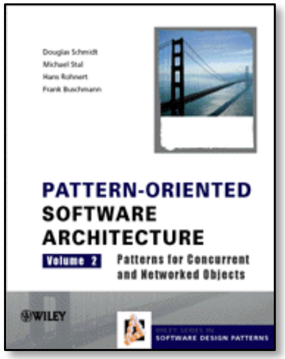

WILEY SERIES IN SOFTWARE DESIGN PATTERNS

PATTERN-ORIENTED SOFTWARE ARCHITECTURE

A Pattern Language for Distributed Computing

Volume 4

Frank Buschmann
Kevlin Henney
Douglas C. Schmidt


Douglas Schmidt
Michael Stal
Hans Rohnert
Frank Buschmann

PATTERN-ORIENTED SOFTWARE ARCHITECTURE

Volume 2 Patterns for Concurrent and Networked Objects

WILEY

WILEY SERIES IN SOFTWARE DESIGN PATTERNS

See en.wikipedia.org/wiki/Concurrency_pattern

Summary of Course Contents

- Foundational Java functional programming concepts & features
- Coverage of foundational Java concurrency mechanisms
- Patterns/frameworks for concurrent programming
- We assume you know (or can quickly learn) Android, Android Studio, Modern Java, & Git

See item #12 at github.com/douglascraigschmidt/CS5254/wiki/CS-5254-FAQ

Structure of the Lecture Material

Structure of the Lecture Material

- This course has four main topics

Section	Topics
Java Functional Programming	• Coverage of basic & advanced Java functional programming features, e.g. • Lambda expressions & method references • Functional interfaces

See www.youtube.com/playlist?list=PLZ9NgFYEMxp57EQIDnyQ3F-8EqW4ejz8V for more videos on Java functional programming

Structure of the Lecture Material

- This course has four main topics

Section	Topics
Java Functional Programming	• Coverage of basic & advanced Java functional programming features, e.g. • Lambda expressions & method references • Functional interfaces
Java Threading	• Coverage of basic & advanced Java threading mechanisms, e.g. • Java Threads & Runnables • The Java Executor framework

Structure of the Lecture Material

- This course has four main topics

Section	Topics
Java Functional Programming	• Coverage of basic & advanced Java functional programming features, e.g. • Lambda expressions & method references • Functional interfaces
Java Threading	• Coverage of basic & advanced Java threading mechanisms, e.g. • Java Threads & Runnables • The Java Executor framework
Java Synchronization	• Coverage of basic & advanced Java synchronization mechanisms, e.g., • Build-in monitor objects • Myriad synchronizer classes in java.util.concurrent

Structure of the Lecture Material

- This course has four main topics

Section	Topics
Java Functional Programming	• Coverage of basic & advanced Java functional programming features, e.g. • Lambda expressions & method references • Functional interfaces
Java Threading	• Coverage of basic & advanced Java threading mechanisms, e.g. • Java Threads & Runnables • The Java Executor framework
Java Synchronization	• Coverage of basic & advanced Java synchronization mechanisms, e.g., • Build-in monitor objects • Myriad synchronizer classes in java.util.concurrent
Software Patterns	• Concurrency patterns

See en.wikipedia.org/wiki/Concurrency_pattern

Structure of the Lecture Material

- This course has four main topics

Section	Topics
Java Functional Programming	• Coverage of basic & advanced Java functional programming features, e.g. • Lambda expressions & method references • Functional interfaces
Java Threading	• Coverage of basic & advanced Java threadi mechanisms, e.g. • Java Threads & Runnables • The Java Executor framework
Java Synchronization	• Coverage of basic & advanced Java synchronization mechanisms, e.g., • Build-in monitor objects • Myriad synchronizer classes in java.util.concurrent
Software Patterns	• Concurrency patterns

We'll bounce around when covering these topics to facilitate assignments

Structure of the Lecture Material

- This course has four main topics
  - Each topic is composed of lessons

Structure of the Lecture Material

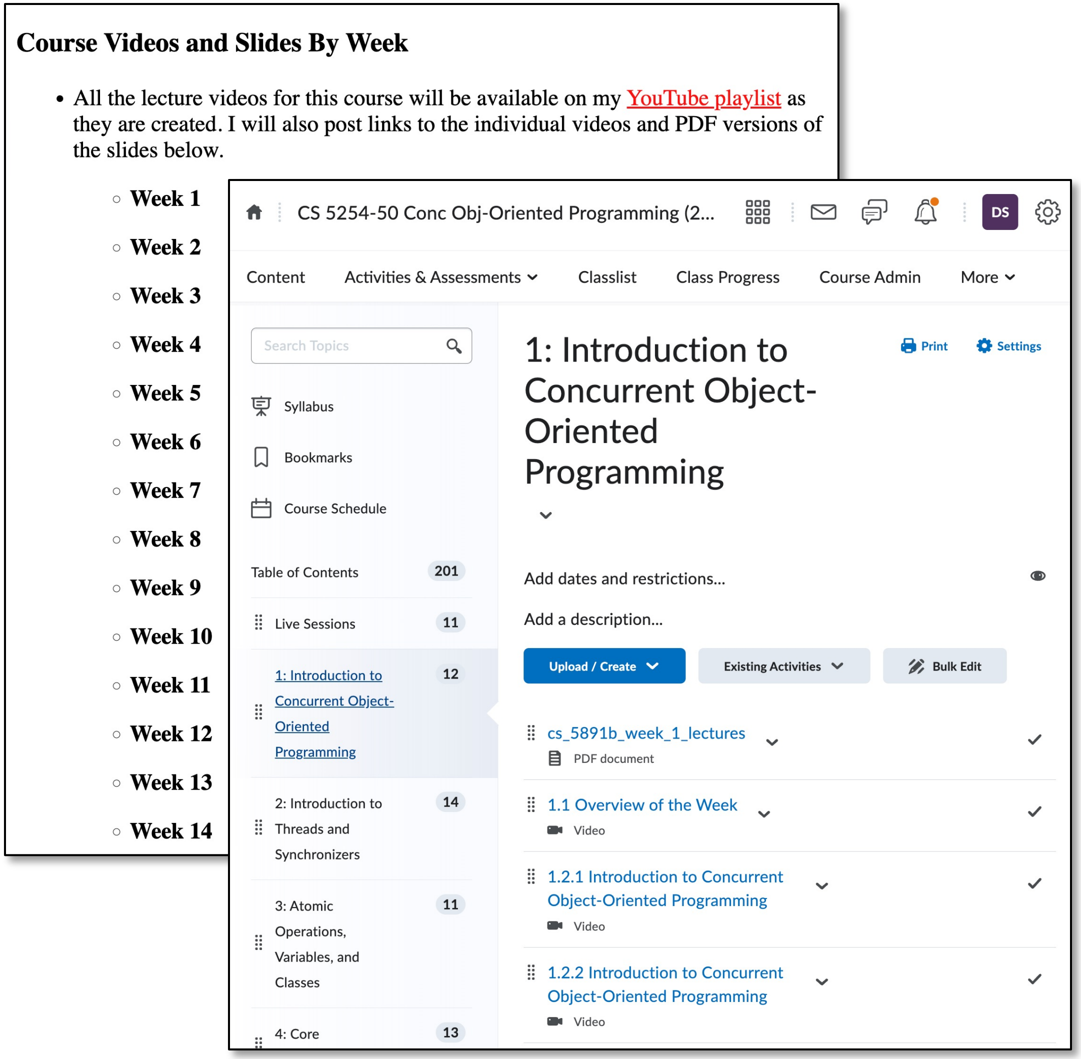
- This course has four main topics
  - Each topic is composed of lessons
  - Each lesson is composed of parts

Structure of the Lecture Material

- This course has four main topics
  - Each topic 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

- This course has four main topics
  - Each topic is composed of lessons
  - Each lesson is composed of parts
  - Each part is a single lecture
    - Each part is composed of segments



Course Videos and Slides By Week

- All the lecture videos for this course will be available on my YouTube playlist as they are created. I will also post links to the individual videos and PDF versions of the slides below.
  - **Week 1**
  - **Week 2**
  - **Week 3**
  - **Week 4**
  - **Week 5**
  - **Week 6**
  - **Week 7**
  - **Week 8**
  - **Week 9**
  - **Week 10**
  - **Week 11**
  - **Week 12**
  - **Week 13**
  - **Week 14**

Table of Contents	201
Live Sessions	11
1: Introduction to Concurrent Object-Oriented Programming	12
2: Introduction to Threads and Synchronizers	14
3: Atomic Operations, Variables, and Classes	11
4: Core	13

1: Introduction to Concurrent Object-Oriented Programming

Add dates and restrictions...

Add a description...

- cs_5891b_week_1_lectures — PDF document
- 1.1 Overview of the Week — Video
- 1.2.1 Introduction to Concurrent Object-Oriented Programming — Video
- 1.2.2 Introduction to Concurrent Object-Oriented Programming — Video

Videos of each lecture are available on brightspace & at www.dre.vanderbilt.edu/~schmidt/cs254

Structure of the Lecture Material

- There will be periodic tests on material covered in the lectures

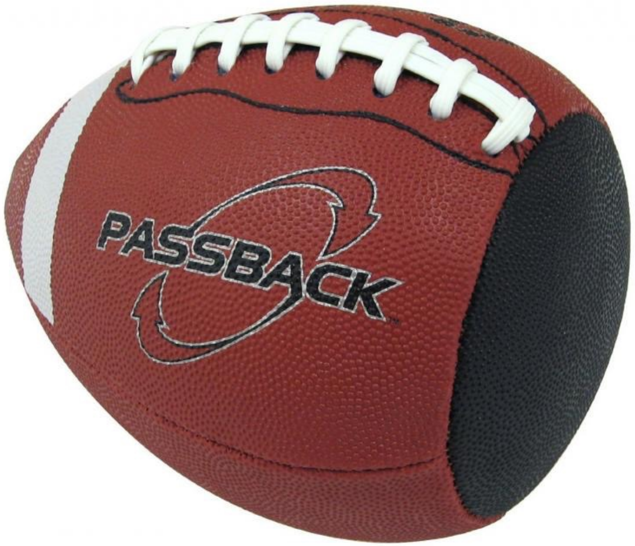
Structure of the Lecture Material

- There will be periodic tests on material covered in the lectures
  - All tests (including the final) are “closed book,” “closed Internet,” “closed ChatGPT,” “closed computing devices,” etc.

1st monthly exam will be on Tuesday, June 5th via Brightspace

Structure of the Lecture Material

- There will be periodic tests on material covered in the lectures
  - All tests (including the final) are "closed book," "closed Internet," "closed ChatGPT," "closed computing devices," etc.
  - We'll try to grade & review the tests by the next class



One of the benefits of a smaller class ;-)

Structure of the Lecture Material

- There will be periodic tests on material covered in the lectures
  - All tests (including the final) are "closed book," "closed Internet," "closed ChatGPT," "closed computing devices," etc.
  - We'll try to grade & review the tests by the next class

I recommend you study for exams by reviewing slides & (re)watching the videos on brightspace & youtube

Structure of the Lecture Material

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

CS 5254: Concurrent Object-Oriented & Functional Programming: Overview (Part 1)