

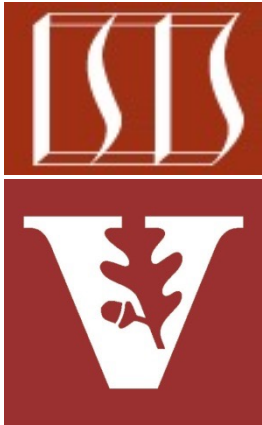
CS 254: Concurrent Object-Oriented Programming with Java & Android

Course Overview & Logistics

Douglas C. Schmidt

d.schmidt@vanderbilt.edu

www.dre.vanderbilt.edu/~schmidt



Professor of Computer Science

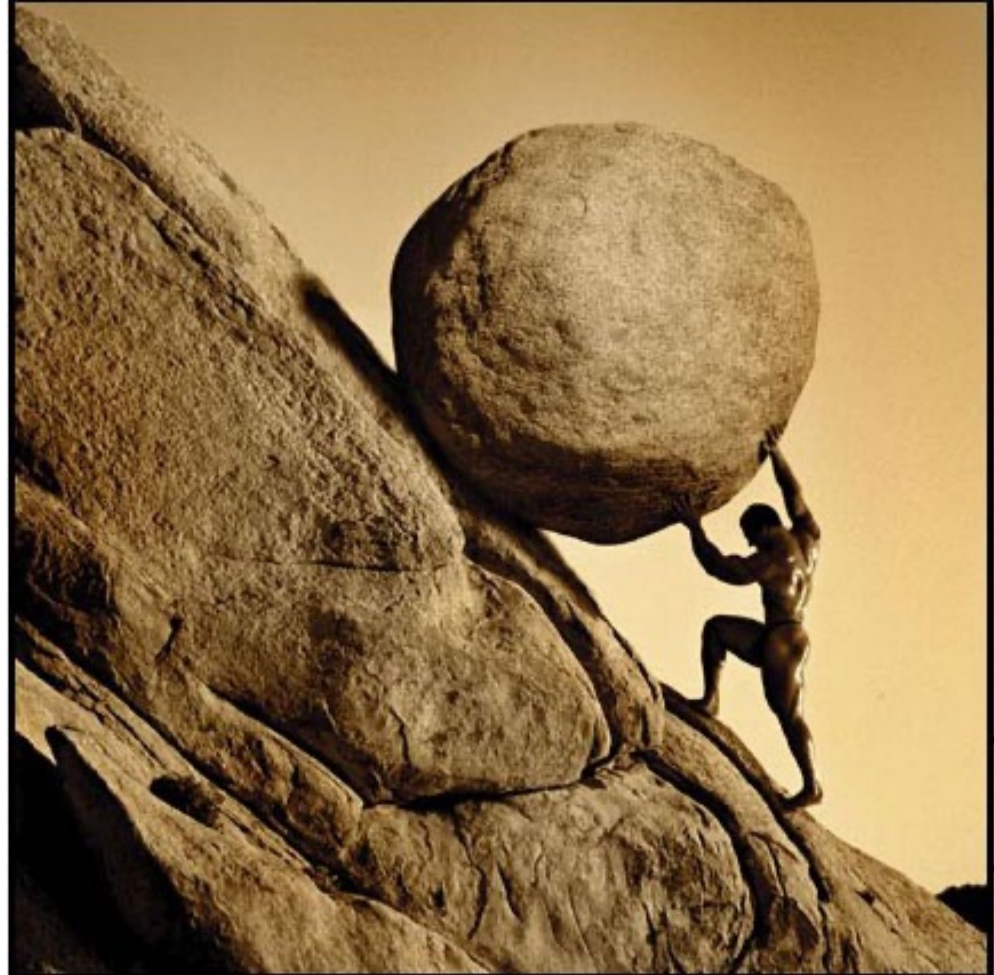
**Institute for Software
Integrated Systems**

**Vanderbilt University
Nashville, Tennessee, USA**



Learning Objectives in this Lesson

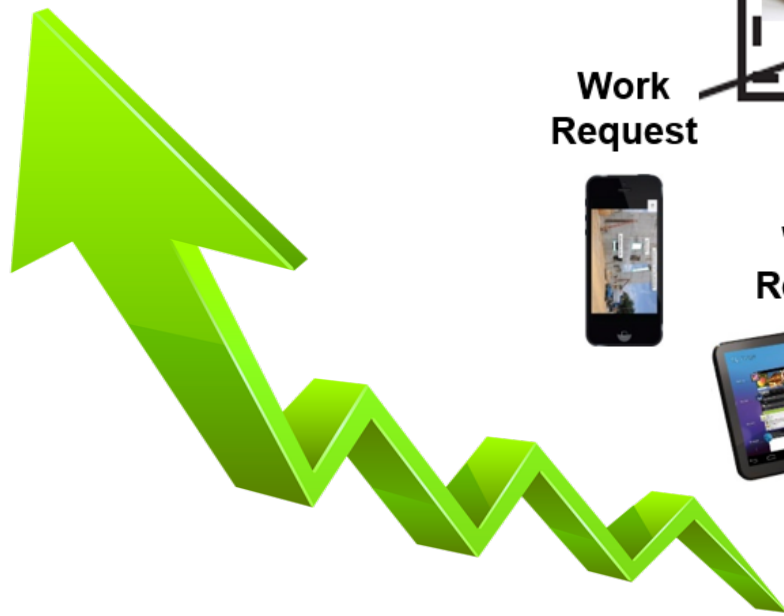
- Understand the course topics & logistics
 - Course philosophy
 - Course contents
 - Structure of the lecture material
 - Overview of the assignments & assessments
 - Setting up the Java & Android IDE on Android Studio
 - Setting up GitLab et al.
 - Accessing Android & Java source code



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



Work Request



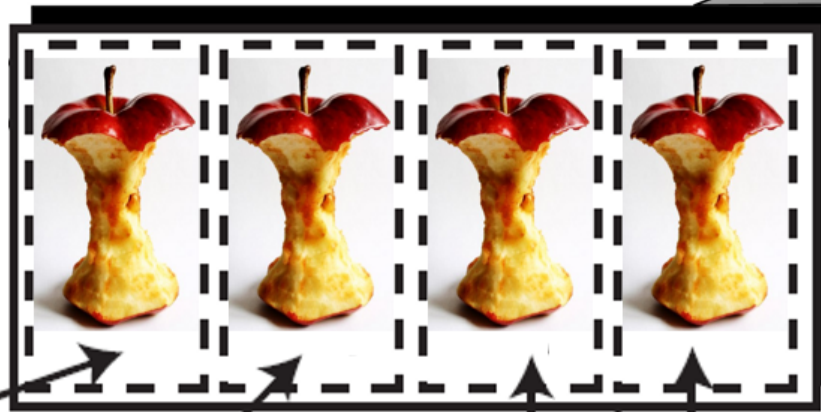
Work Request



Work Request



Work Request



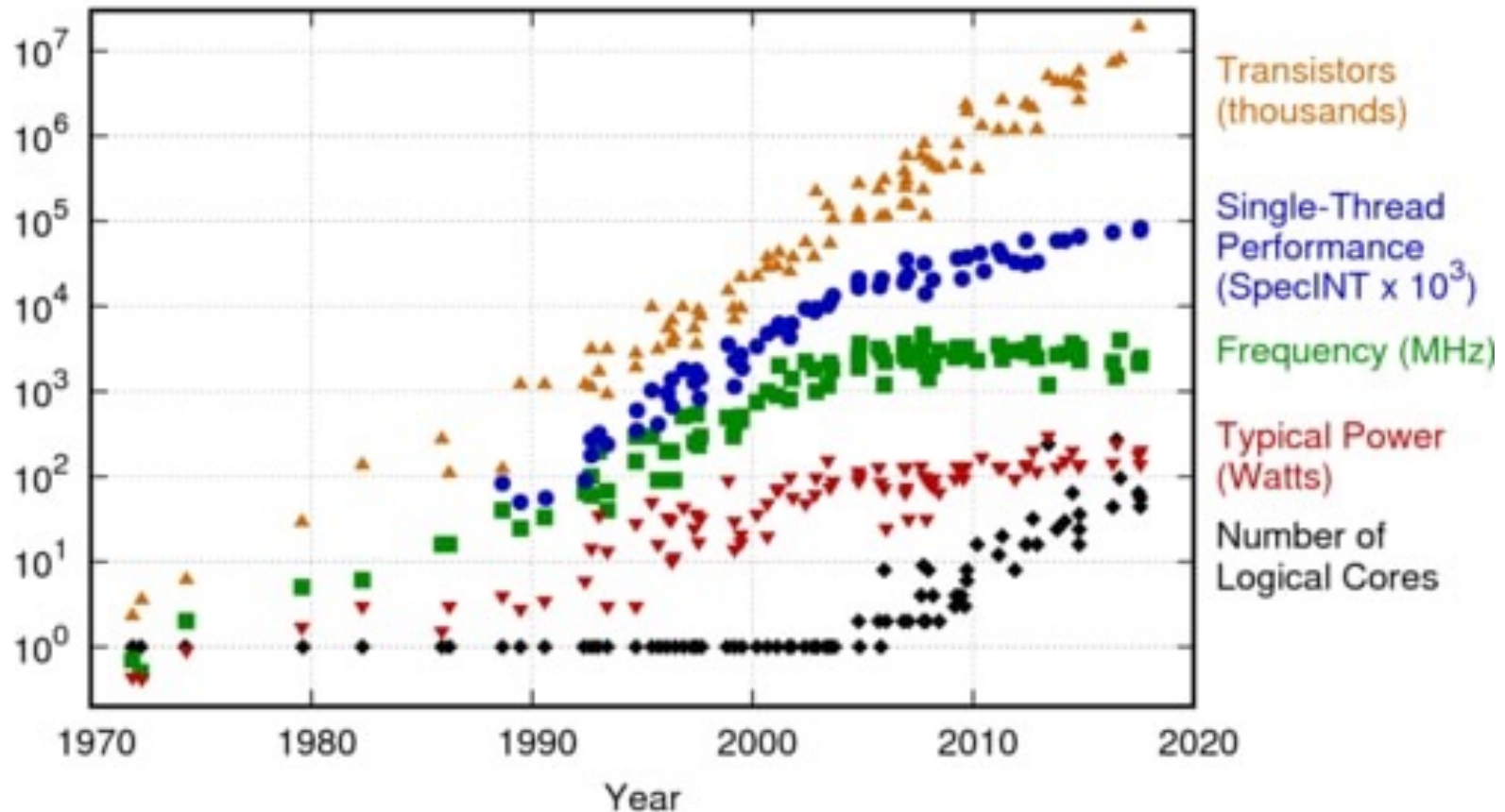
Cloud Server

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

Course Philosophy

- 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

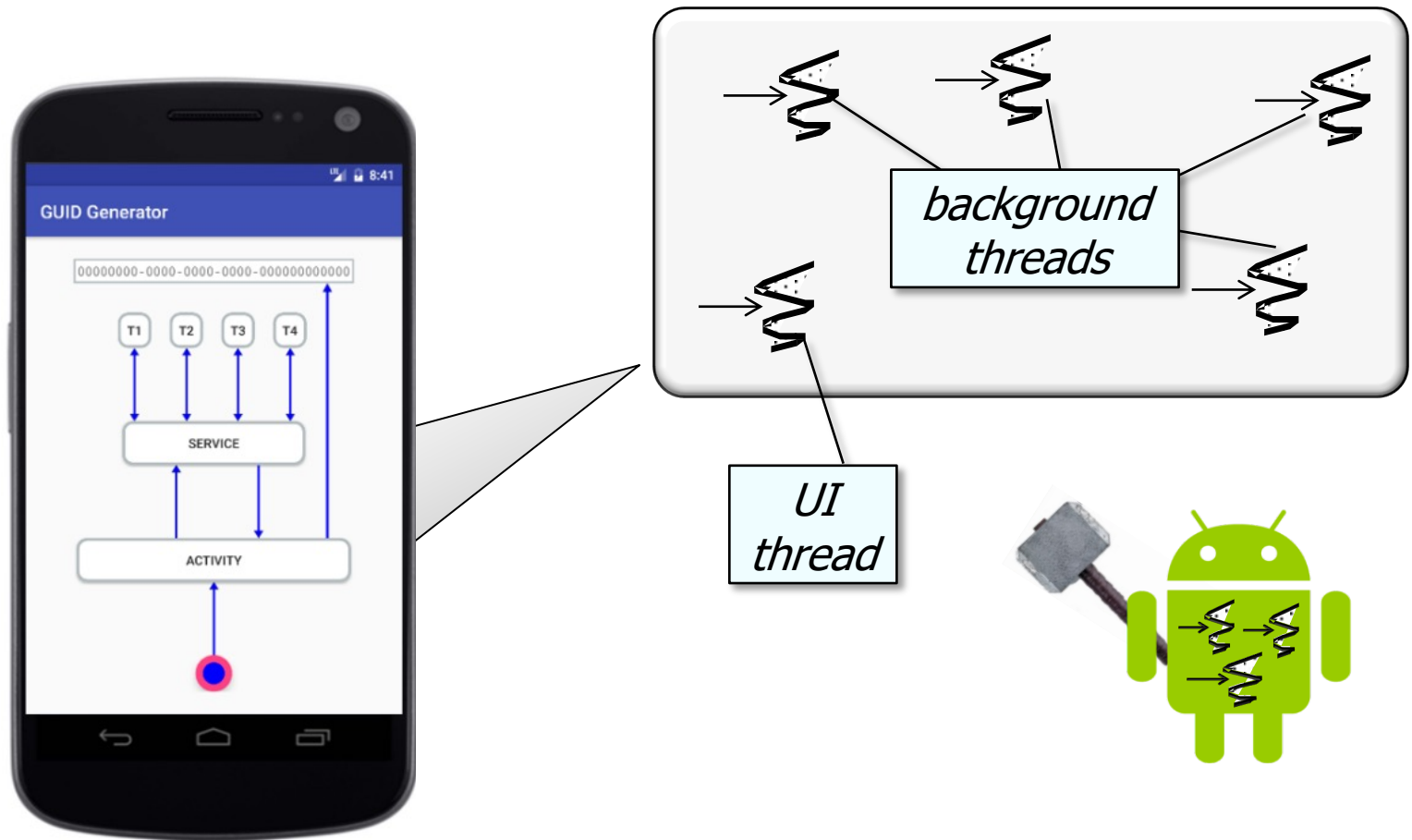
42 Years of Microprocessor Trend Data



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

Course Philosophy

- Concurrency is also commonly used to offload work from the user interface (UI) thread to background thread(s) in mobile devices



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

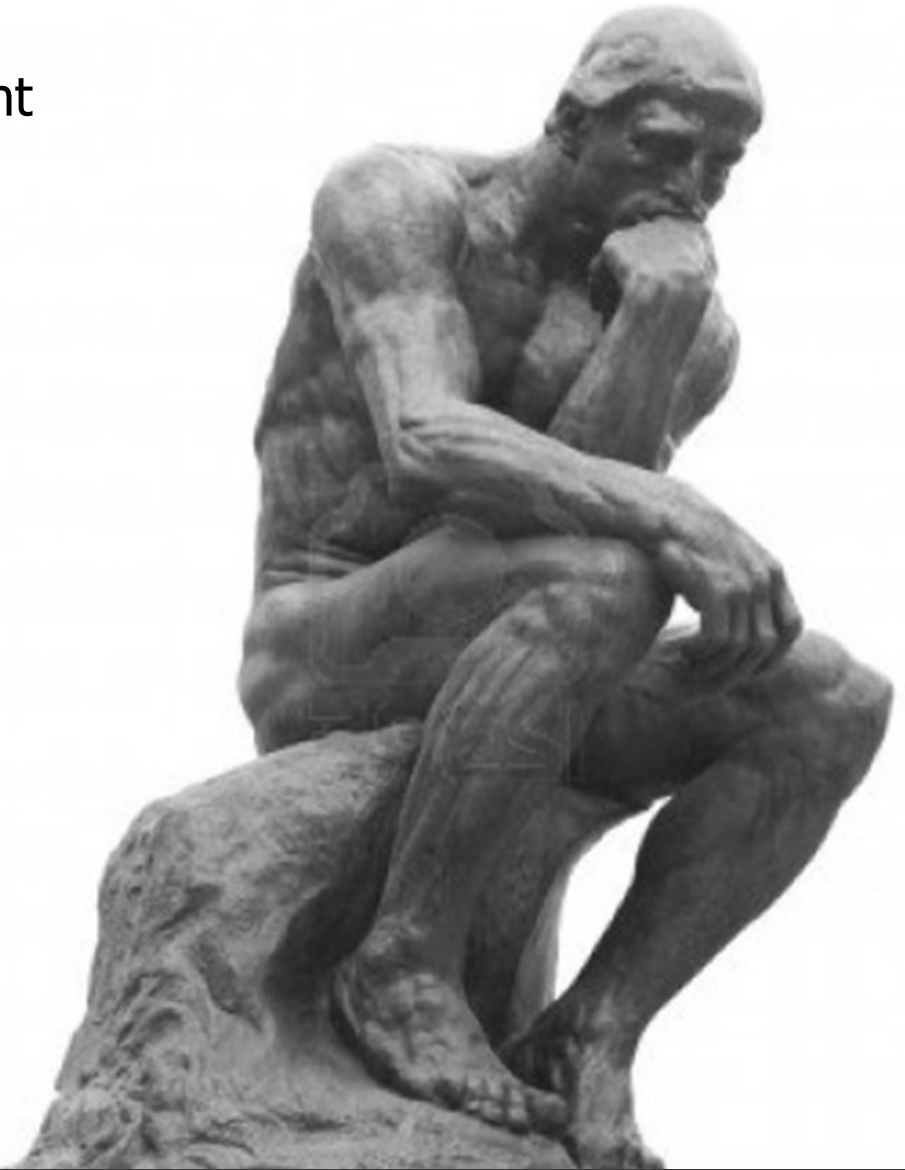
Course Philosophy

- 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 & object-oriented (& some functional) design & programming techniques

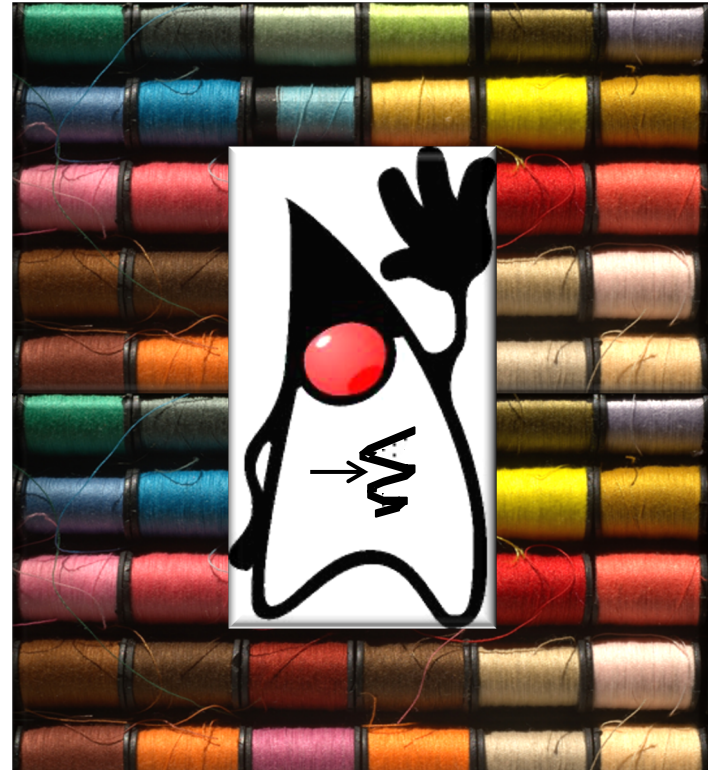


This course involves lots of hands-on software development & testing!

Summary of the Course Contents

Summary of Course Contents

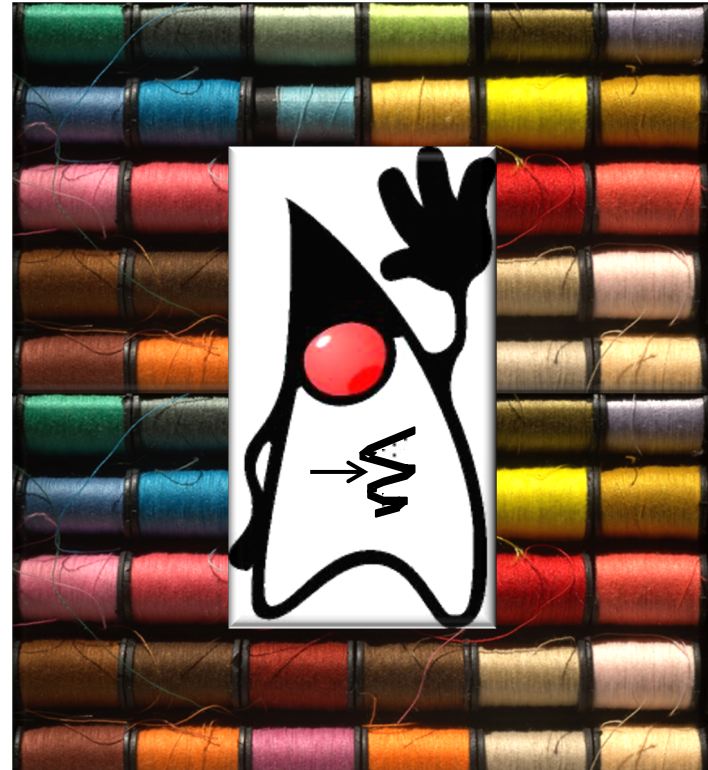
- Coverage of foundational Java concurrency mechanisms



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

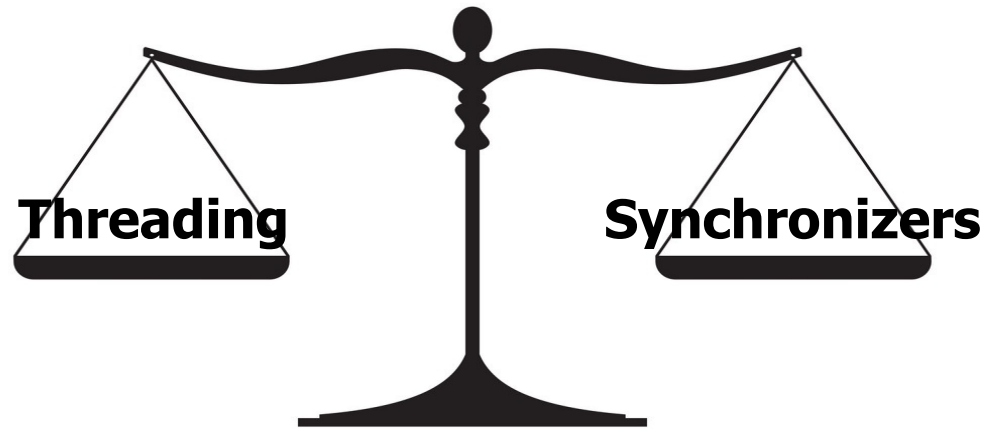
Summary of Course Contents

- Coverage of foundational Java concurrency mechanisms
 - e.g., Java threading & synchronizer mechanisms



Summary of Course Contents

- Coverage of foundational Java concurrency mechanisms
 - e.g., Java threading & synchronizer mechanisms



Threading mechanisms

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

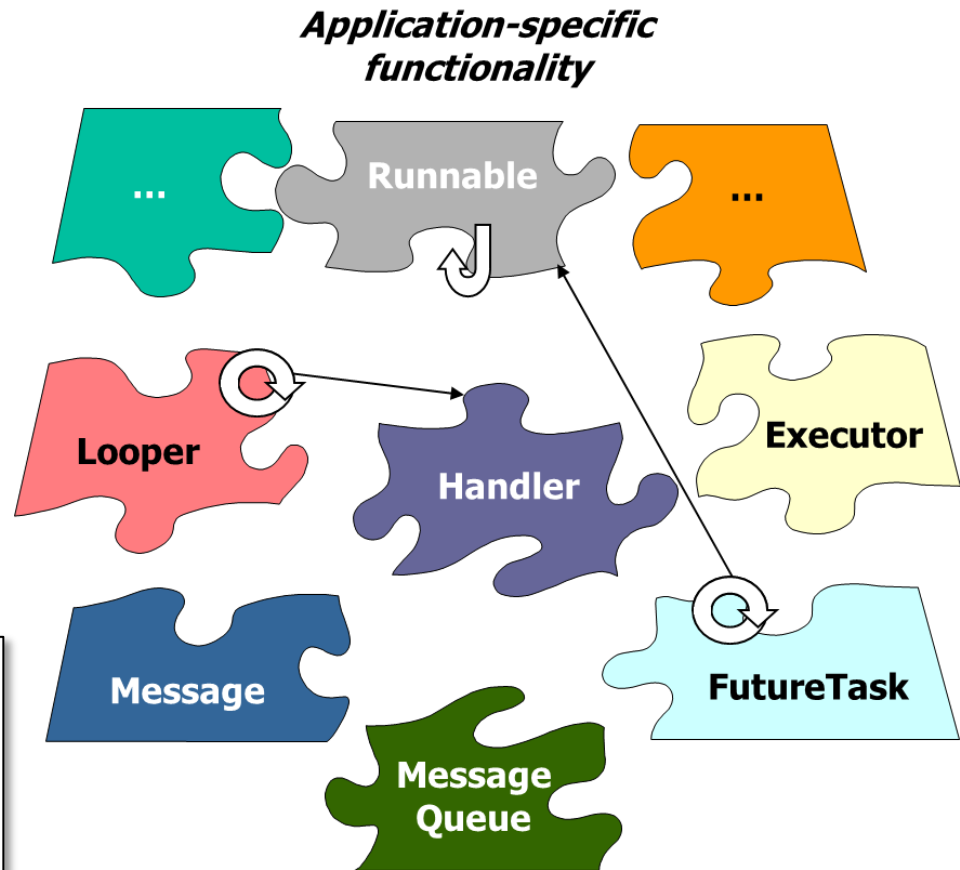
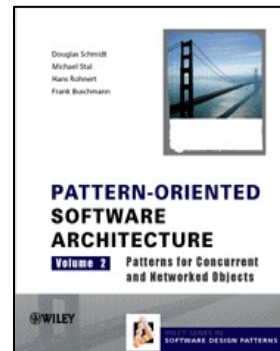
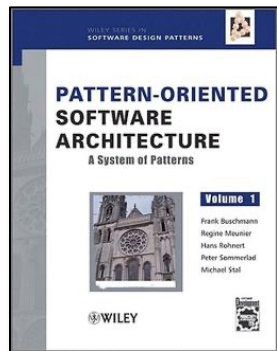
Synchronizer mechanisms

- 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

- Coverage of foundational Java concurrency mechanisms
- Patterns/frameworks for concurrent programming



See en.wikipedia.org/wiki/Concurrency_pattern

Summary of Course Contents

- 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



Structure of the Lecture Material

Structure of the Lecture Material

- This course has three main topics

Section	Topics
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 three main topics

Section	Topics
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 <code>java.util.concurrent</code>

Structure of the Lecture Material

- This course has three main topics

Section	Topics
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 <code>java.util.concurrent</code>
Software Patterns	• Concurrency patterns

Structure of the Lecture Material

- This course has three main topics

Section	Topics
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 <code>java.util.concurrent</code>
Software Patterns	• Concurrency patterns



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

Structure of the Lecture Material

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



Structure of the Lecture Material

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



Structure of the Lecture Material

- This course has three main topics
 - Each topic 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/cs254#lectures

Structure of the Lecture Material

- This course has three 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

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



Structure of the Lecture Material

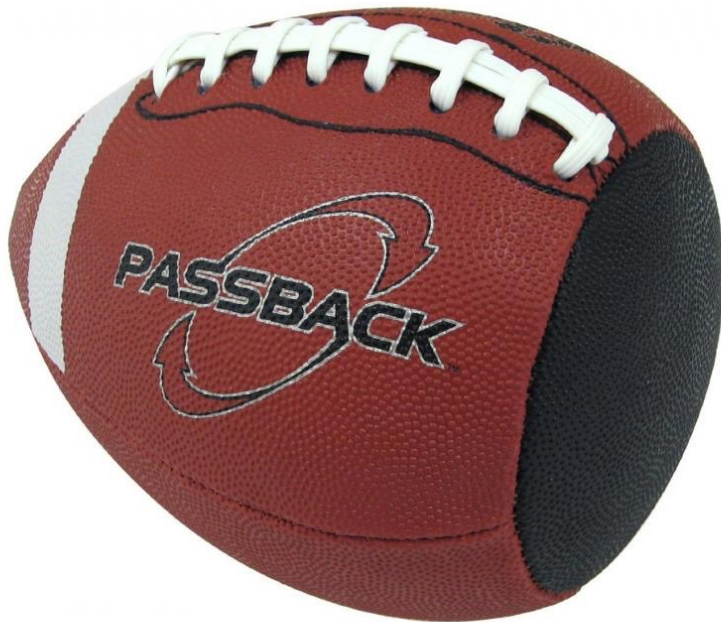
- There will be periodic tests on material covered in the lectures
- All tests (including the final) are “closed book,” “closed Internet,” “closed electronics,” etc.



1st test quiz will be on Wednesday, January 26th 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 electronics,” 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 electronics,” etc.
- We’ll try to grade & review the tests by the next class



I recommend that you study for tests by reviewing slides & watching screencasts available at www.dre.vanderbilt.edu/~schmidt/cs254#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 electronics,” etc.
- We’ll try to grade & review the tests by the 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/cs254/work-summary.html#quizzes

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 electronics,” etc.
- We’ll try to grade & review the tests by the 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 test & don’t attend class you’ll be penalized 50%



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

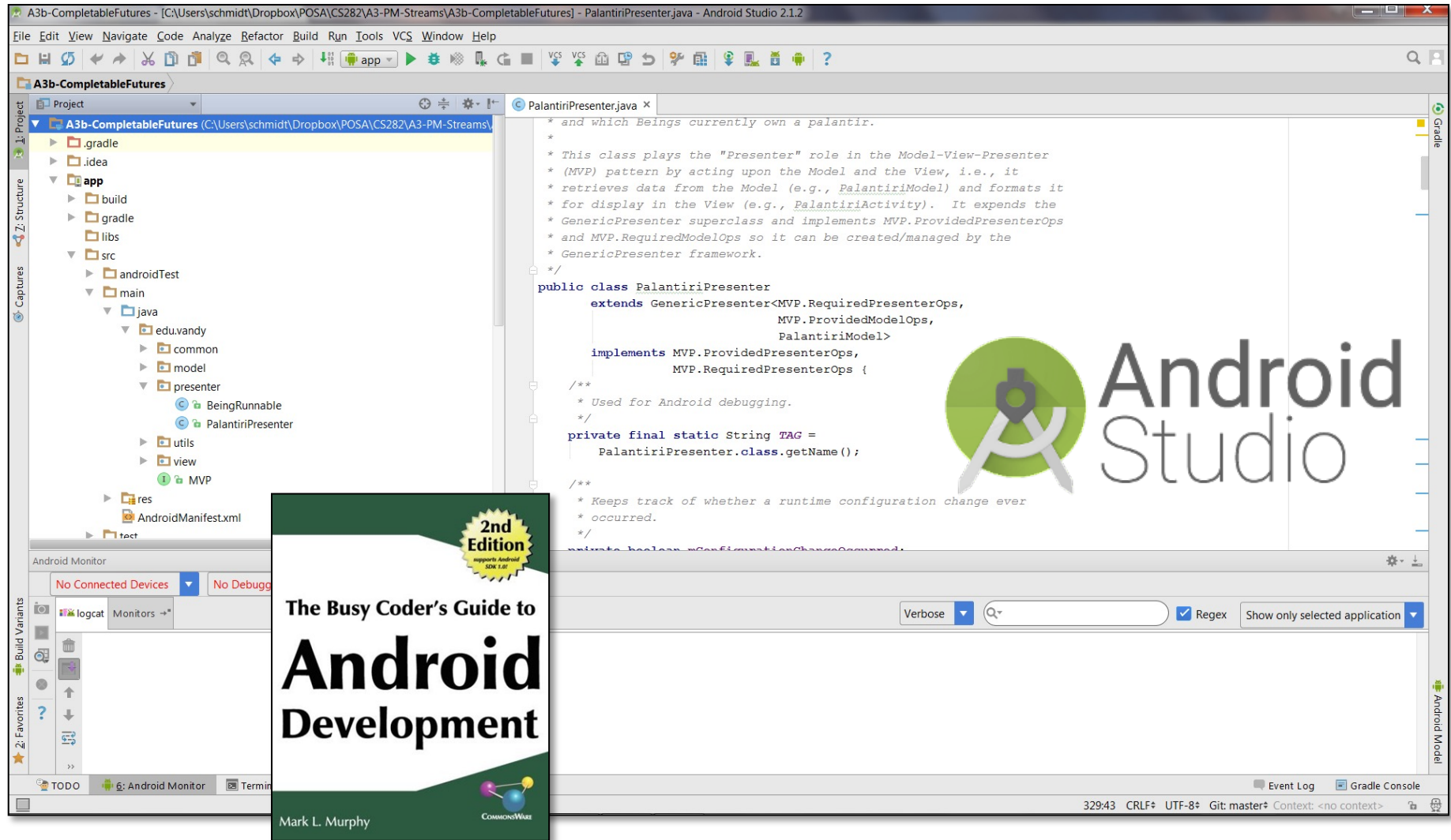


The final exam is noon to 3pm, Monday, May 2nd via Brightspace

Overview of the Assignments & Assessments

Overview of Assignments & Assessments

- Programming assignments should be written in Java 11 using Android Studio



You can use any IDE, but your final submission *must* build/run with the latest Android Studio & Android 11 "R" (API level 30)

Overview of Assignments & Assessments

- Programming assignments should be written in Java 11 using Android Studio
- The Java 11 runtime environment (JRE) comes pre-installed with Android now



You can use cool Java 11 features starting with Android Studio "Arctic Fox"

Overview of Assignments & Assessments

- Android programming assignments must be submitted using Android Studio

- A wizard for creating new apps
- A visual editor for creating GUIs
- An editor for manipulating Android XML descriptors needed for your app
- An emulator for testing your apps on your PC
- A debugger for finding errors in the emulator or on a device



See developer.android.com/sdk

Overview of Assignments & Assessments

- Android programming assignments must be submitted using Android Studio
 - Please install Android 11 "R" (API level 30)



See en.wikipedia.org/wiki/Android_11

Overview of Assignments & Assessments

- All source code for assignments & examples available at GitHub

The screenshot shows the GitHub repository page for `douglasraigschmidt / CS254`. At the top, there are buttons for 'Unwatch', 'Star' (0), and 'Fork' (0). Below this is a navigation bar with links for 'Code', 'Issues', 'Pull requests', 'Actions', 'Projects', 'Wiki', 'Security', 'Insights', and 'Settings'. The 'Code' tab is selected. Below the navigation bar, there are buttons for 'master' (1 branch, 0 tags), 'Go to file', 'Add file', and 'Code'. The main content area shows a list of files: 'assignment1a' (updated 9 minutes ago) and 'README.md' (updated 7 hours ago). Below this is a preview of the 'README.md' file, which contains the text: 'CS254' and 'Contains examples and assignments for my CS 254 course at Vanderbilt University, which can be accessed via <http://www.dre.vanderbilt.edu/~schmidt/cs254>'. On the right side, there is an 'About' section with a description: 'Contains examples and assignments for my CS 254 course at Vanderbilt University, which can be accessed via <http://www.dre.vanderbilt.edu/~schmidt/cs254>'. Below this are sections for 'Releases' (No releases published) and 'Packages' (No packages published).

`douglasraigschmidt / CS254` Unwatch 1 Star 0 Fork 0

<> Code ! Issues 🔗 Pull requests ▶ Actions 📁 Projects 📖 Wiki ! Security 📈 Insights ⚙ Settings

🔑 master 1 branch 0 tags Go to file Add file Code

douglasraigschmidt updates ae6cf26 9 minutes ago ⌚ 2 commits

File	Update	Time
assignment1a	updates	9 minutes ago
README.md	updates	7 hours ago

README.md

CS254

Contains examples and assignments for my CS 254 course at Vanderbilt University, which can be accessed via <http://www.dre.vanderbilt.edu/~schmidt/cs254>

About

Contains examples and assignments for my CS 254 course at Vanderbilt University, which can be accessed via <http://www.dre.vanderbilt.edu/~schmidt/cs254>

Releases

No releases published
[Create a new release](#)

Packages

No packages published
[Publish your first package](#)

Go to GitHub at github.com/douglasraigschmidt/CS254

Overview of Assignments & Assessments

- All source code for assignments & examples available at GitHub
- You will need to learn how to use GitLab et al.

A screenshot of the GitLab website's landing page. The background is a solid purple color. In the top left corner is the GitLab logo (a cat face) and the text "GitLab". In the top right corner is a white hamburger menu icon. The main heading is "Open source software to collaborate on code" in a large, white, sans-serif font. Below this is a paragraph of white text describing GitLab's features: "GitLab offers git repository management, code reviews, issue tracking, activity feeds and wikis. Enterprises install GitLab on-premise and connect it with LDAP and Active Directory servers for secure authentication and authorization. A single GitLab server can handle more than 25,000 users but it is also possible to create a high availability setup with multiple active servers." Below the paragraph is another paragraph: "Do you want more from your GitLab installation? A subscription bundles the Enterprise Edition with support from the GitLab team. The Enterprise Edition allows you to sync LDAP groups, control pushes via git hooks, integrate better with Jenkins and Jira, and to run MySQL and forward logs when using our Omnibus package. Our service engineers will help you keep your server running smoothly." At the bottom are two buttons: a light blue button labeled "GitLab Community Edition" and a green button labeled "Get a subscription".

GitLab

Open source software to collaborate on code

GitLab offers git repository management, code reviews, issue tracking, activity feeds and wikis. Enterprises install GitLab on-premise and connect it with LDAP and Active Directory servers for secure authentication and authorization. A single GitLab server can handle more than 25,000 users but it is also possible to create a high availability setup with multiple active servers.

Do you want more from your GitLab installation? A subscription bundles the Enterprise Edition with support from the GitLab team. The Enterprise Edition allows you to sync LDAP groups, control pushes via git hooks, integrate better with Jenkins and Jira, and to run MySQL and forward logs when using our Omnibus package. Our service engineers will help you keep your server running smoothly.

GitLab Community Edition

Get a subscription

See item #13 at github.com/douglasraigschmidt/CS254/wiki/CS-254-FAQ

Overview of Assignments & Assessments

- All source code for assignments & examples available at GitHub
 - You will need to learn how to use GitLab et al.
- Be prepared to update your repositories multiple times
 - i.e., you need to understand Git!



"If you don't like change, you're going to like irrelevance even less."

Overview of Assignments & Assessments

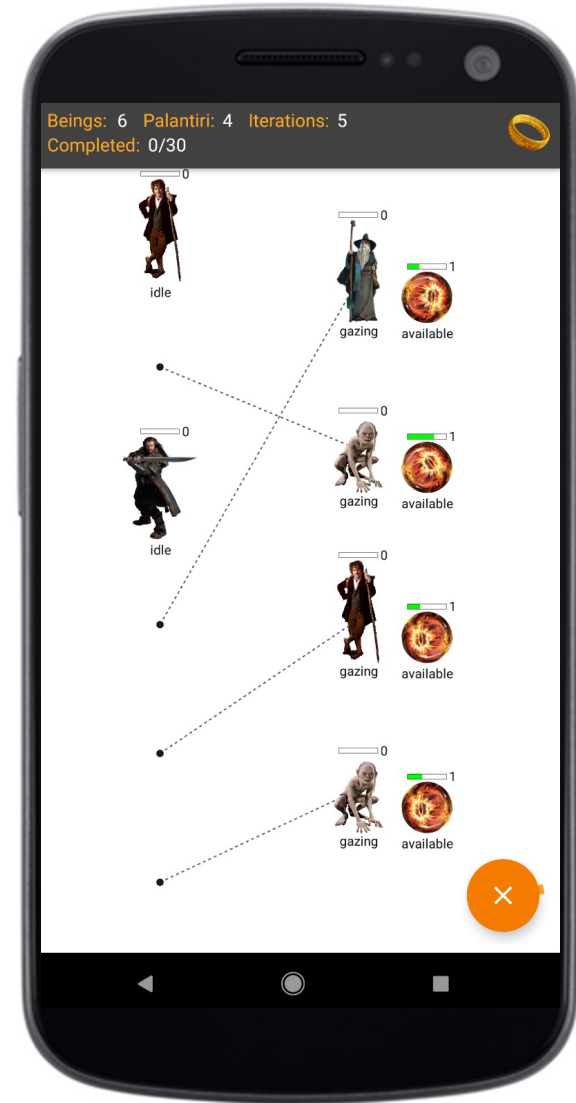
- Assignments will provide a range of experience with modern Java & Android concurrent programs



See github.com/douglasraigschmidt/CS254

Overview of Assignments & Assessments

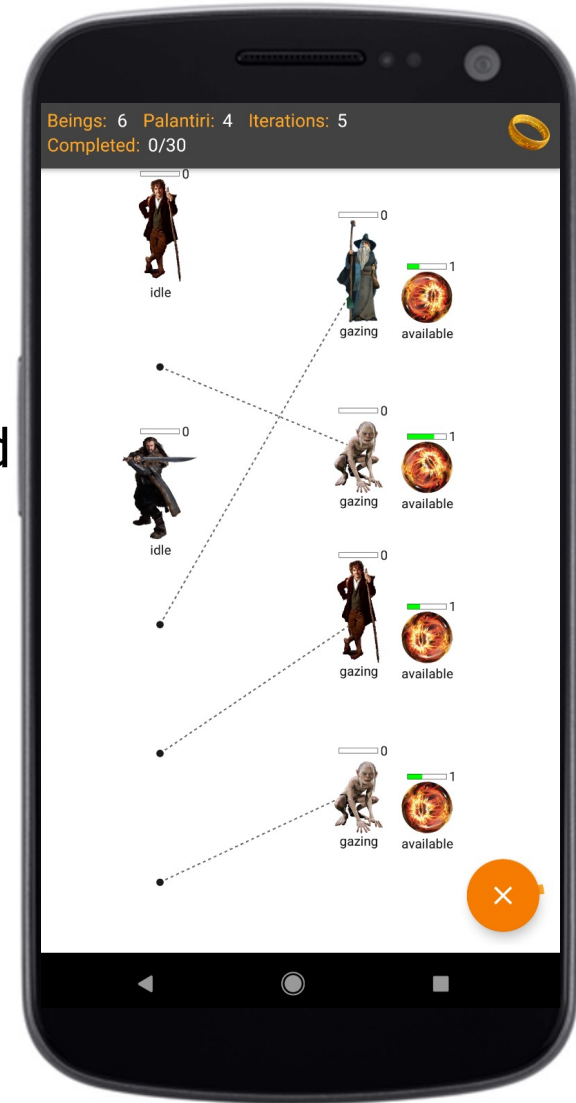
- In particular, you'll implement multiple variants of a Java concurrent resource manager & an associated Android app



See en.wikipedia.org/wiki/Palantir

Overview of Assignments & Assessments

- In particular, you'll implement multiple variants of a Java concurrent resource manager & an associated Android app, e.g.
 - Java Thread & Runnable
 - Java Executor framework (e.g., `ExecutorService`, `ExecutorCompletionService`, & `ForkJoinPool`)
 - Java Semaphore, built-in monitor objects, `StampedLock`, `ReentrantLock`, `ConcurrentHashMap`, etc.



The topics covered by the assignments generalize to more than Android!

Overview of Assignments & Assessments

- Assignment assessments will be done via reviews by course staff



Overview of Assignments & Assessments

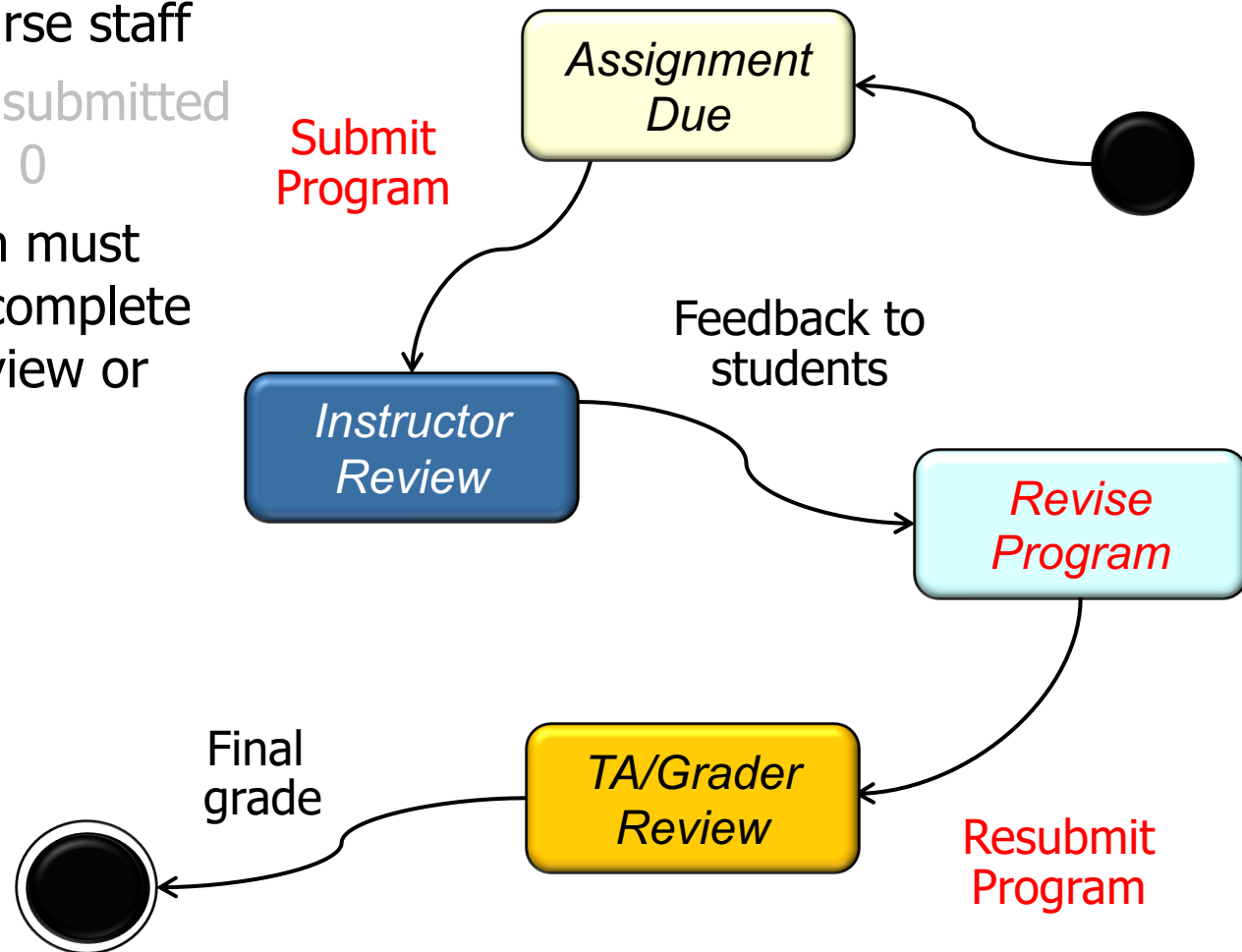
- Assignment assessments will be done via reviews by course staff
- Assignments *must* be submitted on time or you'll get a 0



See item #4 at github.com/douglasraigschmidt/CS254/wiki/CS-254-FAQ

Overview of Assignments & Assessments

- Assignment assessments will be done via reviews by course staff
 - Assignments *must* be submitted on time or you'll get a 0
- Your initial submission must compile & be largely complete or you won't get a review or a final grade



Overview of Assignments & Assessments

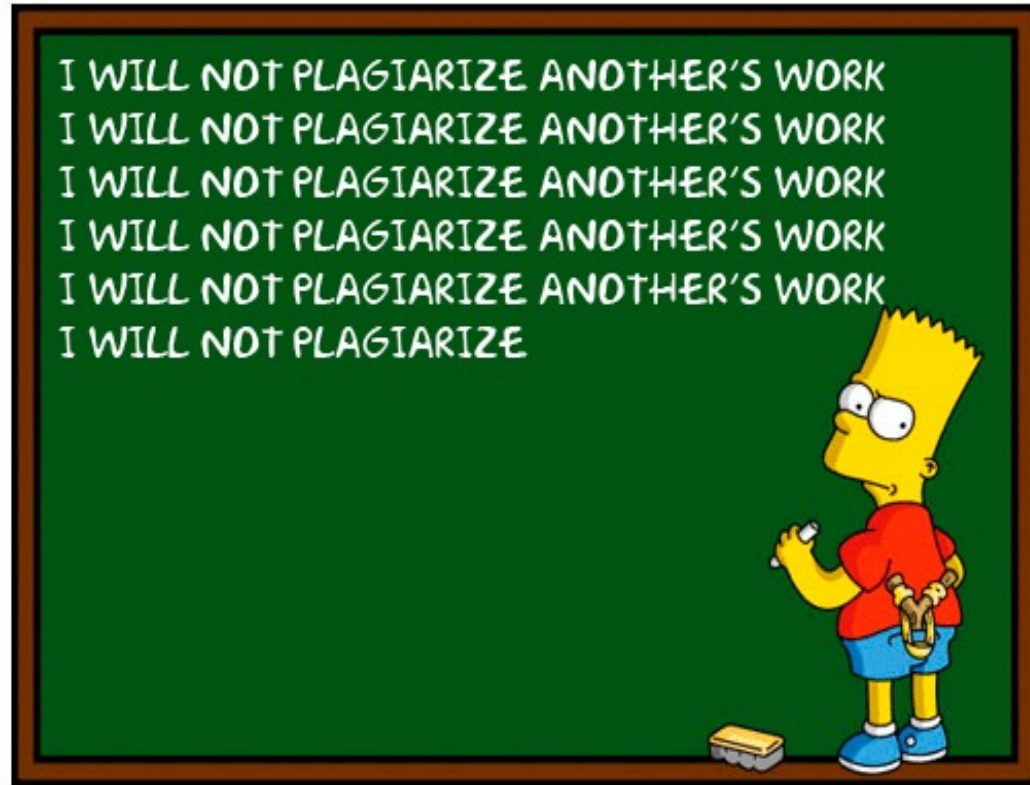
- Assignment assessments will be done via reviews by course staff
 - Assignments *must* be submitted on time or you'll get a 0
 - Your initial submission must compile & be largely complete or you won't get a review or a final grade
- You will not receive a grade for assignments if you do not attend class regularly



See www.dre.vanderbilt.edu/~schmidt/cs254/assignments.html

Overview of Assignments & Assessments

- Assignment assessments will be done via reviews by course staff
 - Assignments *must* be submitted on time or you'll get a 0
 - Your initial submission must compile & be largely complete or you won't get a review or a final grade
 - You will not receive a grade for assignments if you do not attend class regularly
- Work *must* be your own
 - This applies for quizzes & programming assignments



Overview of Assignments & Assessments

- The bulk of your grade is based on the results of the automated unit tests

assignment4 - Android Studio

Project Files

Run: All in assignment4.image-crawler

Tests failed: 33, passed: 88, ignored: 48 of 169 tests - 11 s 586 ms

Test ignored.

Test ignored.

Test ignored.

Test ignored.

Test ignored.

Test ignored.

java.lang.AssertionError: Verification failed: call 1 of 1: class java.util.concurrent.CompletableFuture.supplyAsync(any

Calls to same mock:

- 1) class java.util.concurrent.CompletableFuture.completedFuture(Page(mockPage#11))
- 2) class java.util.concurrent.CompletableFuture.reportGet(Page(mockPage#11))

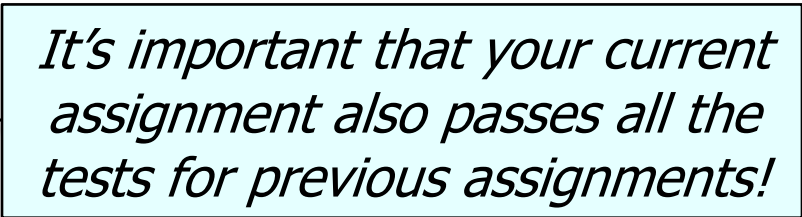
at io.mockk.impl.recording.states.VerifyingState.failIfNotPassed(VerifyingState.kt:66)
at io.mockk.impl.recording.states.VerifyingState.recordingDone(VerifyingState.kt:42)
at io.mockk.impl.recording.CommonCallRecorder.done(CommonCallRecorder.kt:47)
at io.mockk.impl.eval.RecordedBlockEvaluator.record(RecordedBlockEvaluator.kt:60)
at io.mockk.impl.eval.VerifyBlockEvaluator.verify(VerifyBlockEvaluator.kt:30)
at io.mockk.MockKDsl.internalVerify(API.kt:118)
at io.mockk.MockKKt.verify(MockK.kt:146)
at io.mockk.MockKKt.verify\$default(MockK.kt:143)
at edu.vanderbilt.imagecrawler.crawlers.CompletableFuturesCrawlerTests.getPageAsyncWhiteBox(CompletableFuturesCrawlerTests.kt:143)
at org.mockito.internal.junit.JUnitRule\$1.evaluateSafely(JUnitRule.java:52)

Build | TODO | Git | Terminal | Run

Tests failed: 33, passed: 88, ignored: 48 (moments ago)

See www.dre.vanderbilt.edu/~schmidt/cs254/assignments.html

- The bulk of your grade is based on the results of the automated unit tests



See item #16 at github.com/douglascraigshmidt/CS254/wiki/CS-254-FAQ

Overview of Assignments & Assessments

- The bulk of your grade is based on the results of the automated unit tests

Tests failed: 33, passed: 88, ignored: 48 of 169 tests – 11 s 586 ms

"C:\Program Files\Java\jdk1.8.0_201\bin\java.exe" ...

Test ignored.

Test ignored.

Test ignored.

Test ignored.

Test ignored.

Test ignored.

java.lang.AssertionError: Verification failed: call 1 of 1: class java.util.concurrent.CompletableFuture.supplyAsync(any)

Calls to same mock:

- 1) class java.util.concurrent.CompletableFuture.completedFuture(Page(mockPage#11))
- 2) class java.util.concurrent.CompletableFuture.reportGet(Page(mockPage#11))

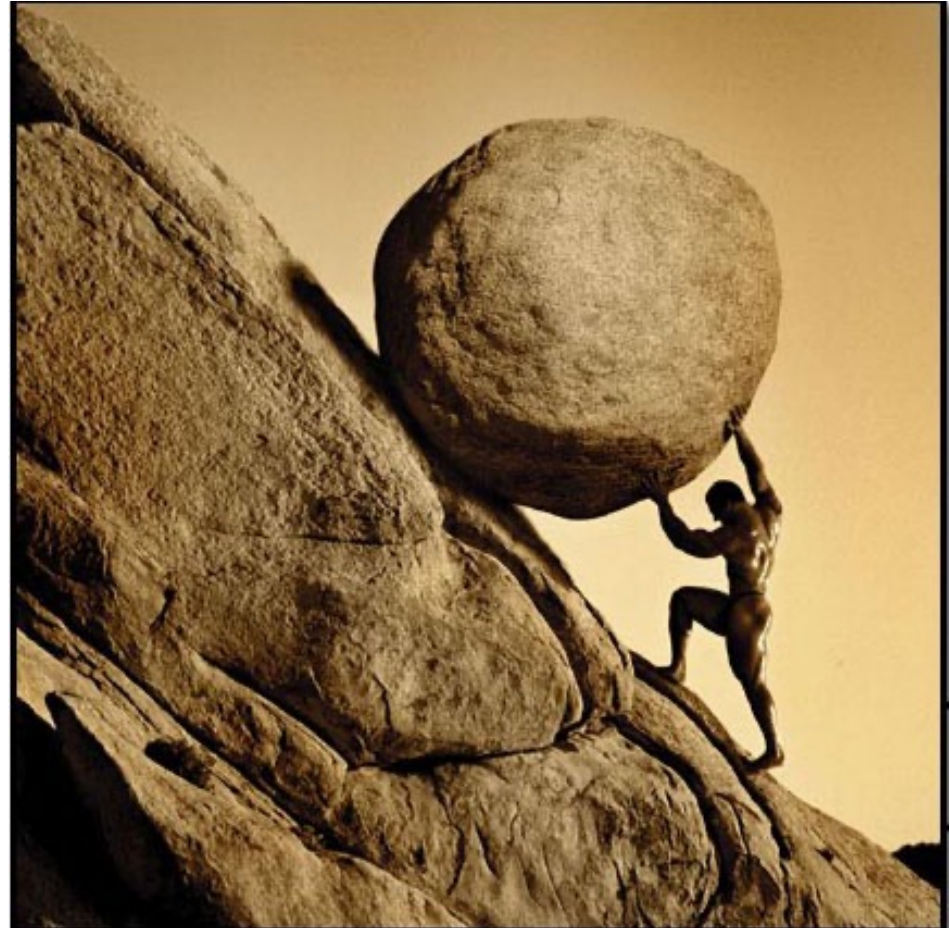
at io.mockk.impl.eval.VerifyBlockEvaluator.verify(VerifyBlockEvaluator.kt:30)
at io.mockk.MockKDsl.internalVerify(API.kt:118)
at io.mockk.MockKkt.verify(MockK.kt:146)
at io.mockk.MockKkt.verify\$default(MockK.kt:143)
at edu.vanderbilt.imagecrawler.crawlers.CompletableFutureCrawlerTests.getPageAsyncWhiteBox(CompletableFutureCrawlerTests.kt:143)
at org.mockito.internal.junit.JUnitRule\$1.evaluateSafely(JUnitRule.java:52)

Please don't upgrade to a different (i.e., newer) version of Gradle!

See item #17 at github.com/douglasraigschmidt/CS254/wiki/CS-254-FAQ

Overview of Assignments & Assessments

- The relative weighting of each portion of the course is:
 - 45% Quizzes
 - 40% Programming projects
 - 10% Final exam
 - 05% Participation



These weightings may change, depending on various factors

Overview of Assignments & Assessments

- The relative weighting of each portion of the course is:
 - 45% Quizzes
 - 40% Programming projects
 - 10% Final exam
 - **05% Participation**
 - Participation includes attendance, involvement, & “following directions”

IMPORTANT

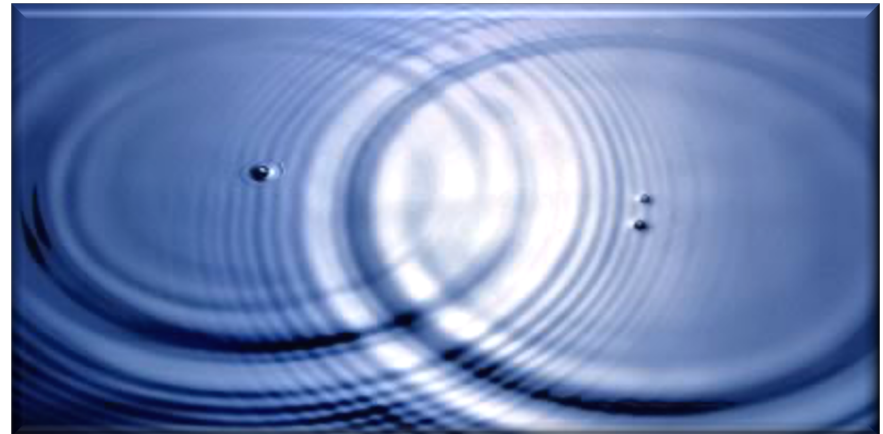


Overview of Assignments & Assessments

- The relative weighting of each portion of the course is:
 - 45% Quizzes
 - 40% Programming projects
 - 10% Final exam
 - **05% Participation**
 - Participation includes attendance, involvement, & “following directions”

Attendance also affects other aspects of your quiz & assignment grades

IMPORTANT



See www.dre.vanderbilt.edu/~schmidt/cs254/work-summary.html#quizzes
& www.dre.vanderbilt.edu/~schmidt/cs254/assignments.html

Overview of Assignments & Assessments

- The relative weighting of each portion of the course is:
 - 45% Quizzes
 - 40% Programming projects
 - 10% Final exam
 - **05% Participation**
 - Participation includes attendance, involvement, & “following directions”

IMPORTANT



Don't expect to get an A in this class if you do not actively participate!!!!

Setting Up the Android & Java IDE on Android Studio

Installing Java/Android Developer Tools

- To use Android, you need to install the latest release of Android Studio



Android Studio provides the fastest tools for building apps on every type of Android device.

Download Android Studio

2020.3.1 for Windows 64-bit (912 MiB)

Download options

Release notes

See developer.android.com/studio

Installing Java/Android Developer Tools

- Installation steps



Installing Java/Android Developer Tools

- Installation steps
 - Download & install the latest version of Android Studio

Android Developers > Android Studio > Preview Was this helpful?  

New features in Android Studio Preview

Table of contents

- Android Studio Bumblebee | 2021.1.1
- Android Testing

 **Warning:** The JCenter repository became read-only on March 31st, 2021. For more information, see [JCenter service update](#).

Android Studio Arctic Fox | 2020.3.1 has been released to the stable channel. [Download it here](#).

Android Studio Bumblebee | 2021.1.1 is currently in the Canary and Dev channels.

Android Gradle plugin (AGP) 7.0 has been released to the stable channel. For more information, see the [AGP release notes](#).

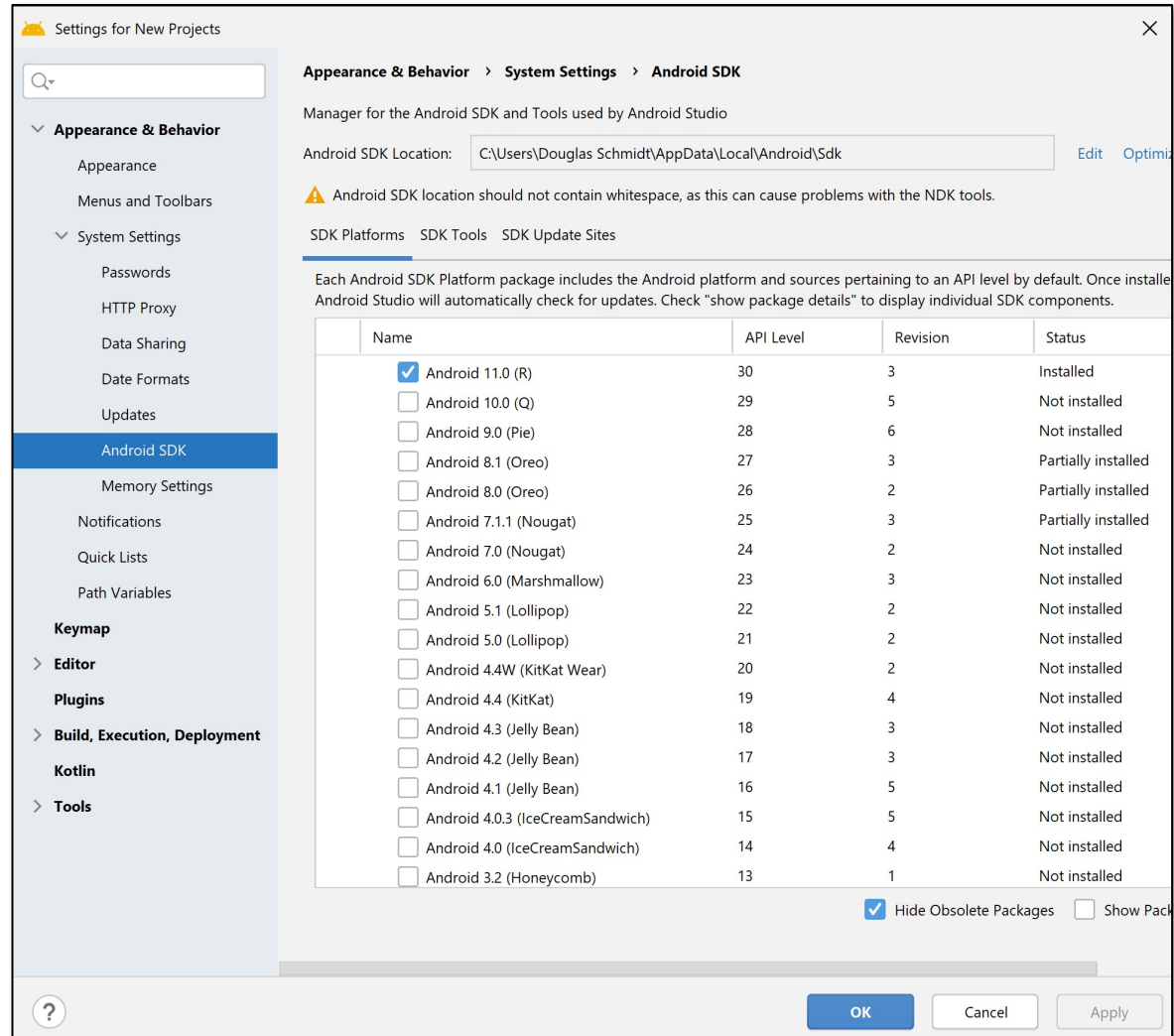
For the latest news on releases, including a list of notable fixes in each release, also see the [Release updates](#).

If you encounter any problems using a preview version of Android Studio, please [let us know](#). Your bug reports help to make Android Studio better.

See developer.android.com/studio

Add Components to the SDK

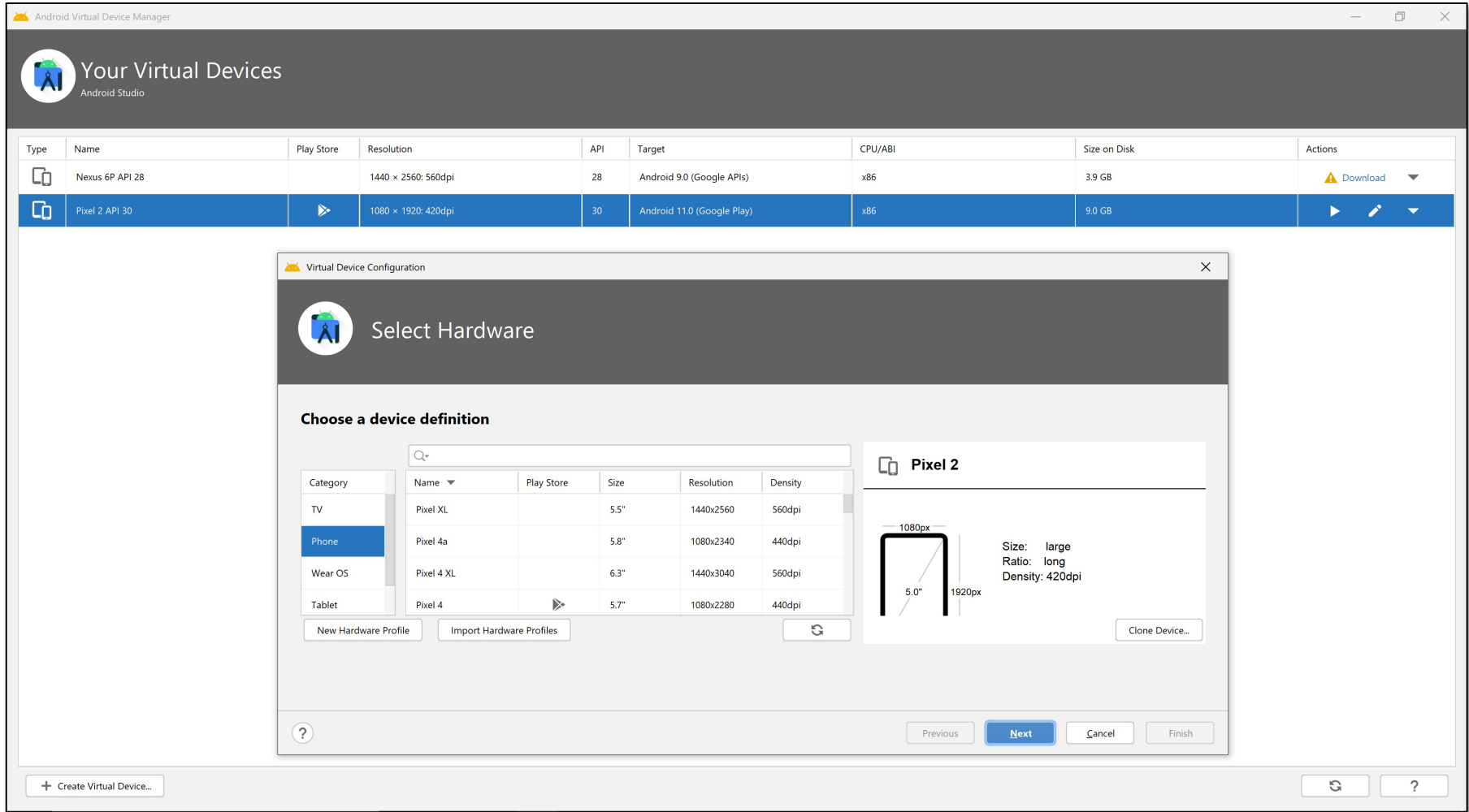
- Launch the Android Studio SDK Manager
 - Select "R" version of Android (11, API 30)



See developer.android.com/studio/intro/update.html

Add Components to the SDK

- Launch the Android Studio Virtual Device Manager
 - Create an Android API 30 emulator



developer.android.com/tools/devices/managing-avds.html

Intel HAXM Virtualization Driver

- **Requirements**

- Intel virtualization extensions (VT, VT-x, vmx)
- AMD virtualization extensions (AMD-v, SVM) [only supported on Linux]
- Download an x86 emulator image

- **Windows & Mac OSX**

- <sdk>/extras/intel/Hardware_Accelerated_Execution_Manager/IntelHAXM.exe/dmg

- **Linux**

- Install KVM & pass “-enable-kvm” flag to emulator when starting



developer.android.com/studio/run/emulator-acceleration

Setting Up GitLab et al.

Setting Up GitLab et al.

1. Create Your GitLab Repo

`www.gitlab.com`

```
git clone git@gitlab.com:your-name/CS-254-spring-2022.git
```



**Working Folder
(Student's)**

**GitLab Repo
(Student's)**

```
cd CS-254-spring-2022
```

3. Change Director into Your Working Folder

See docs.gitlab.com/ee/ssh for info on setting up an SSH key for GitLab et al.

Setting Up GitLab et al.

1. Update from Read-Only GitHub Repo

```
git remote add skeletons
```

```
git@github.com:douglasraigschmidt/CS254.git
```

2. Get Current Version

```
git pull skeletons master
```

Working Folder (Student's)

Assignment1a
...

3. Do work!

Local Repo (Student's)

Assignment1a
...

GitLab Repo (Student's)

Assignment1a
...

GitHub Repo (Instructor's)

Assignment1a
Assignment1b
Assignment2a
Assignment2b
...

```
git commit
```

4. Commit Changes

```
git push origin master
```

5. Send Changes to GitLab Repo

See item #13 at github.com/douglasraigschmidt/CS254/wiki/CS-254-FAQ

Setting Up GitLab et al.

The screenshot shows the GitLab web interface. At the top, there's a navigation bar with the GitLab logo, a search bar, and various icons. Below the navigation bar, a banner for the 'Free Trial of GitLab.com Gold' is visible. The main content area is titled 'Welcome to GitLab' with the subtitle 'Code, test, and deploy together'. There are four cards in a 2x2 grid:

- Create a project** (highlighted with a red box): Projects are where you store your code, access issues, wiki and other features of GitLab.
- Create a group**: Groups are the best way to manage projects and members.
- Explore public projects**: There are 1,385,678 public projects on this server. Public projects are an easy way to allow everyone to have read-only access.
- Learn more about GitLab**: Take a look at the documentation to discover all of GitLab's capabilities.

Setting Up GitLab et al.

New Project · GitLab

gitlab.com/projects/new

Apps Nashville Symphony Knowledge Check...

GitLab Projects Groups Activity Milestones Snippets

Search or jump to...

New project

A project is where you house your files (repository), plan your work (issues), and publish your documentation (wiki), [among other things](#).

All features are enabled for blank projects, from templates, or when importing, but you can disable them afterward in the project settings.

To only use CI/CD features for an external repository, choose **CI/CD for external repo**.

Information about additional Pages templates and how to install them can be found in our [Pages getting started guide](#).

Tip: You can also create a project from the command line. [Show command](#)

Blank project

Create from template

Import project

CI/CD for external repo

Project name

CS-254-spring-2021

Project URL

https://gitlab.com/lclfox/

Project slug

CS-254-spring-2021

Want to house several dependent projects under the same namespace? [Create a group](#).

Project description (optional)

Description format

Visibility Level

☒ Private

Project access must be granted explicitly to each user.

☐ Public

The project can be accessed without any authentication.

☐ Initialize repository with a README

Allows you to immediately clone this project's repository. Skip this if you plan to push up an existing repository.

Create project

Cancel

Setting Up GitLab et al.

The screenshot shows the GitLab web interface in a browser. The address bar shows the URL `gitlab.com/lclfox/cs-891-fall-2019`. The left sidebar contains navigation links: Project, Details, Activity, Cycle Analytics, Issues (0), Merge Requests (0), CI / CD, Operations, Packages, Wiki, Snippets, and Settings. The main content area shows a notification banner at the top: "You won't be able to pull or push project code via SSH until you add an SSH key to your profile". Below this, a blue banner states: "Project 'CS-254-spring-2021' was successfully created." The project details for "CS-254-spring-2021" (Project ID: 13944228) are shown, including a "Clone" button. The main message says "The repository for this project is empty" and provides options to "New file", "Add README", "Add CHANGELOG", or "Add CONTRIBUTING". Under "Command line instructions", it says "You can also upload existing files from your computer using the instructions below." and provides "Git global setup" instructions: `git config --global user.name "Lindsey Fox"` and `git config --global user.email "lindsey.fox@vanderbilt.edu"`. Finally, under "Create a new repository", it provides the following commands: `git clone https://gitlab.com/lclfox/cs-891-fall-2019.git`, `cd cs-891-fall-2019`, `touch README.md`, and `git add README.md`.

Lindsey Fox / cs-891-fall-2019 · C x

gitlab.com/lclfox/cs-891-fall-2019

GitLab Projects Groups Activity Milestones Snippets

Search or jump to...

cs-891-fall-2019

You won't be able to pull or push project code via SSH until you add an SSH key to your profile Don't show again | Remind later

Lindsey Fox > cs-891-fall-2019 > Details

Project 'CS-254-spring-2021' was successfully created.

CS-254-spring-2021 Project ID: 13944228

Add license

The repository for this project is empty

You can create files directly in GitLab using one of the following options.

New file Add README Add CHANGELOG Add CONTRIBUTING

Command line instructions

You can also upload existing files from your computer using the instructions below.

Git global setup

```
git config --global user.name "Lindsey Fox"
git config --global user.email "lindsey.fox@vanderbilt.edu"
```

Create a new repository

```
git clone https://gitlab.com/lclfox/cs-891-fall-2019.git
cd cs-891-fall-2019
touch README.md
git add README.md
```

Setting Up GitLab et al.

Members · Lindsey Fox / cs-891-fall-2019 / project_members

GitLab Projects Groups Activity Milestones Snippets

Search or jump to...

cs-891-fall-2019

Project

Issues 0

Merge Requests 0

CI / CD

Operations

Packages

Wiki

Snippets

Settings

General

Members

Integrations

Repository

CI / CD

Operations

Collapse sidebar

Lindsey Fox > cs-891-fall-2019 > Members

Project members

You can invite a new member to **CS-254-spring-2021** or invite another group.

Invite member Invite group

GitLab member or Email address

Douglas Craig Schmidt

Choose a role permission

Maintainer

[Read more about role permissions](#)

Access expiration date

Expiration date

Add to project Import

Existing members and groups

Members of **CS-254-spring-2021** 1

Find existing members by name Sort by Name, ascending

Lindsey Fox @lclfox **It's you** Maintainer

Given access 2 minutes ago

Setting Up GitLab et al.

The screenshot shows the GitLab web interface in a browser. The address bar displays `gitlab.com/lclfox/cs-891-fall-2019/edit`. The left sidebar contains a navigation menu with options: Project, Issues (0), Merge Requests (0), CI / CD, Operations, Packages, Wiki, Snippets, Settings (selected), General (selected), Members, Integrations, Repository, CI / CD, Operations, Pages, and Audit Events. The main content area is titled 'Lindsey Fox > cs-891-fall-2019 > General Settings'. It features a section 'Naming, topics, avatar' with a 'Collapse' button. This section includes fields for 'Project name' (CS-254-spring-2021) and 'Project ID' (13944228), a 'Topics' field, and a 'Project description (optional)' field. Below these is the 'Project avatar' section, which shows a circular placeholder with the letter 'C' and a 'Choose file...' button. A note states 'The maximum file size allowed is 200KB.' and a 'Save changes' button is at the bottom. The next section is 'Visibility, project features, permissions' with a 'Collapse' button. It includes a 'Project visibility' dropdown menu set to 'Private' and a note: 'The project is accessible only by members of the project. Access must be granted explicitly to each user.' The bottom of the page shows the start of an 'Issues' section.

Accessing Java & Android Source Code

Accessing Java & Android Source Code

- Android source code is available
 - For browsing
android.googlesource.com

android Git repositories

To clone one of these repositories, install [git](#), and run:

```
| git clone https://android.googlesource.com/name
```

Name

[accessories/manifest](#)
[device/asus/deb](#)
[device/asus/flo](#)
[device/asus/flo-kernel](#)
[device/asus/grouper](#)
[device/asus/tilapia](#)
[device/common](#)
[device/generic/armv7-a](#)
[device/generic/armv7-a-neon](#)
[device/generic/art](#)
[device/generic/common](#)
[device/generic/goldfish](#)
[device/generic/mini-emulator-armv7-a-neon](#)
[device/generic/mini-emulator-mips](#)
[device/generic/mini-emulator-x86](#)

Accessing Java & Android Source Code

- Android source code is available
 - For browsing android.googlesource.com
 - For downloading source.android.com

The Android Source Code

Android is an open-source software stack created for a wide array of devices with different form factors. The primary purposes of Android are to create an open software platform available for carriers, OEMs, and developers to make their innovative ideas a reality and to introduce a successful, real-world product that improves the mobile experience for users. We also wanted to make sure there was no central point of failure, where one industry player could restrict or control the innovations of any other. The result is a full, production-quality consumer product with source code open for customization and porting.

Governance Philosophy

Android was originated by a group of companies known as the Open Handset Alliance, led by Google. Today, many companies – both original members of the OHA and others – have invested heavily in Android. These companies have allocated significant engineering resources to improve Android and bring Android devices to market.

The companies that have invested in Android have done so on its merits because we believe an open platform is necessary. Android is intentionally and explicitly an open-source – as opposed to a free software – effort; a group of organizations with shared needs has pooled resources to collaborate on a single implementation of a shared product. The Android philosophy is pragmatic, first and foremost. The objective is a shared product that each contributor can tailor and customize.

Uncontrolled customization can, of course, lead to incompatible implementations. To prevent this, the Android Open Source Project also maintains the [Android Compatibility Program](#), which spells out what it means to be "Android compatible" and what is required of device builders to achieve that status. Anyone can (and will!) use the Android source code for any purpose, and we welcome all legitimate uses. However, in order to take part in the shared ecosystem of applications we are building around Android, device builders must participate in the Android Compatibility Program.

The Android Open Source Project is led by Google, who maintains and further develops Android. Although Android consists of multiple subprojects, this is strictly a project management technique. We view and manage Android as a single, holistic software product, not a "distribution", specification, or collection of replaceable parts. Our intent is that device builders port Android to a device; they don't implement a specification or curate a distribution.

Accessing Java & Android Source Code

- Java 11 source code is available
 - For browsing zgrepcode.com



The screenshot shows the Java.net website. At the top left is the Java logo and the text "java.net The Source for Java Technology Collaboration". At the top right are links for "Login", "Register", and "Help". On the left side, there is a "JDK 8" menu with options: "Downloads", "Feedback Forum", "OpenJDK", and "Planet JDK". The main content area is titled "JDK 8 Project" with the subtitle "Building the next generation of the JDK platform". Below this, there are three sections: "JDK 8 snapshot builds" with a list of links including "Download 8u40 early access snapshot builds", "Source code (instructions)", "Official Java SE 8 Reference Implementations", "Early Access Build Test Results (instructions)", "Feedback" with a paragraph about using the Project Feedback forum, and "We Want Contributions!" with a paragraph about contributing to the platform.

java.net The Source for Java Technology Collaboration

Login | Register | Help

JDK 8

- Downloads
- Feedback Forum
- OpenJDK
- Planet JDK

JDK 8 Project

Building the next generation of the JDK platform

JDK 8 snapshot builds

- [Download 8u40 early access snapshot builds](#)
- [Source code \(instructions\)](#)
- [Official Java SE 8 Reference Implementations](#)
- [Early Access Build Test Results \(instructions\)](#)

Feedback

Please use the [Project Feedback](#) forum if you have suggestions for or encounter issues using JDK 8.

If you find bugs in a release, please submit them using the usual [Java SE bug reporting channels](#), not with the Issue tracker accompanying this project. Be sure to include complete version information from the output of the `java -version` command.

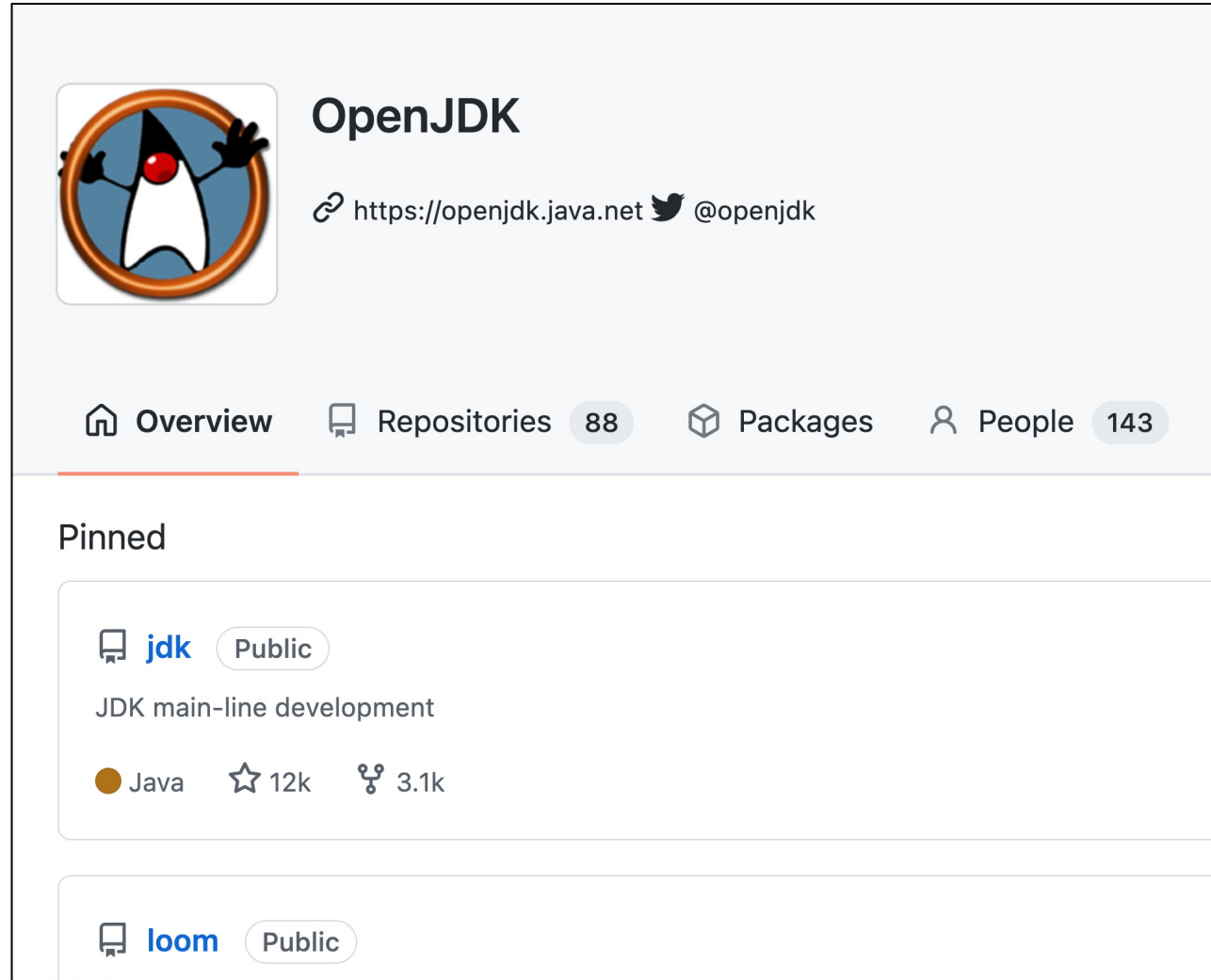
We Want Contributions!

Frustrated with a bug that never got fixed? Have a great idea for improving the Java SE platform? See [how to contribute](#) for information on making contributions to the platform.

Accessing Java & Android Source Code

- Java 11 source code is available

- For browsing zgrepcode.com
- For downloading github.com/openjdk



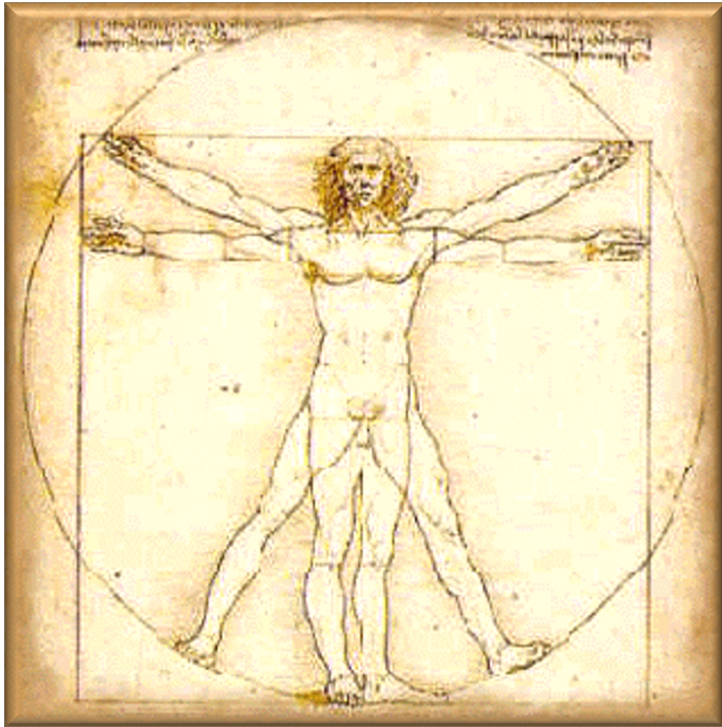
The screenshot displays the GitHub profile for OpenJDK. At the top left is the OpenJDK logo, a cartoon rocket ship with a red nose and black gloves, inside a blue circle with an orange border. To the right of the logo is the name "OpenJDK" in bold black text. Below the name is a link icon followed by the URL "https://openjdk.java.net" and a Twitter icon followed by the handle "@openjdk". Below this information is a navigation bar with four tabs: "Overview" (selected with an orange underline), "Repositories" (with a computer icon), "Packages" (with a cube icon), and "People" (with a person icon). To the right of the "Repositories" and "People" tabs are counts: "88" and "143" respectively. Below the navigation bar is a section titled "Pinned". Under "Pinned", there are two repository cards. The first card is for the "jdk" repository, labeled "Public", with the description "JDK main-line development". It shows a "Java" language tag, 12k stars, and 3.1k forks. The second card is for the "loom" repository, also labeled "Public".

Summary



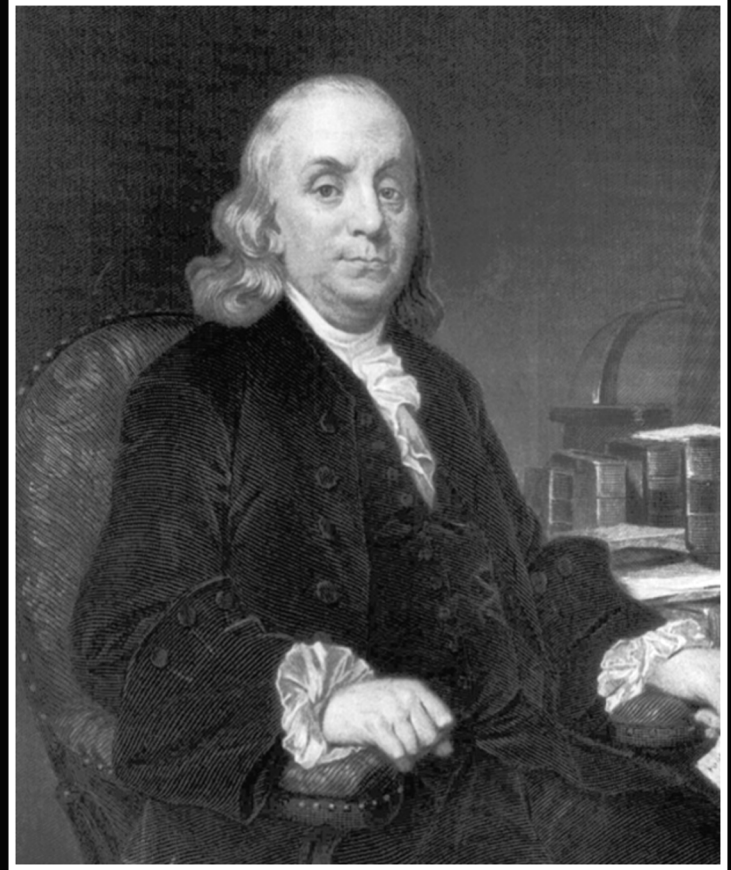
Summary

- You will get out of this course what you put into it



Summary

- You will get out of this course what you put into it
- Be prepared to work hard



HARD WORK

“Human Felicity is produc'd not so much by great Pieces of good Fortune that seldom happen, as by little Advantages that occur every Day” - Benjamin Franklin

Summary

- You will get out of this course what you put into it
 - Be prepared to work hard
 - Do *not* miss deadlines...



See github.com/douglasraigschmidt/CS254/wiki/Assignment-Deadlines

Summary

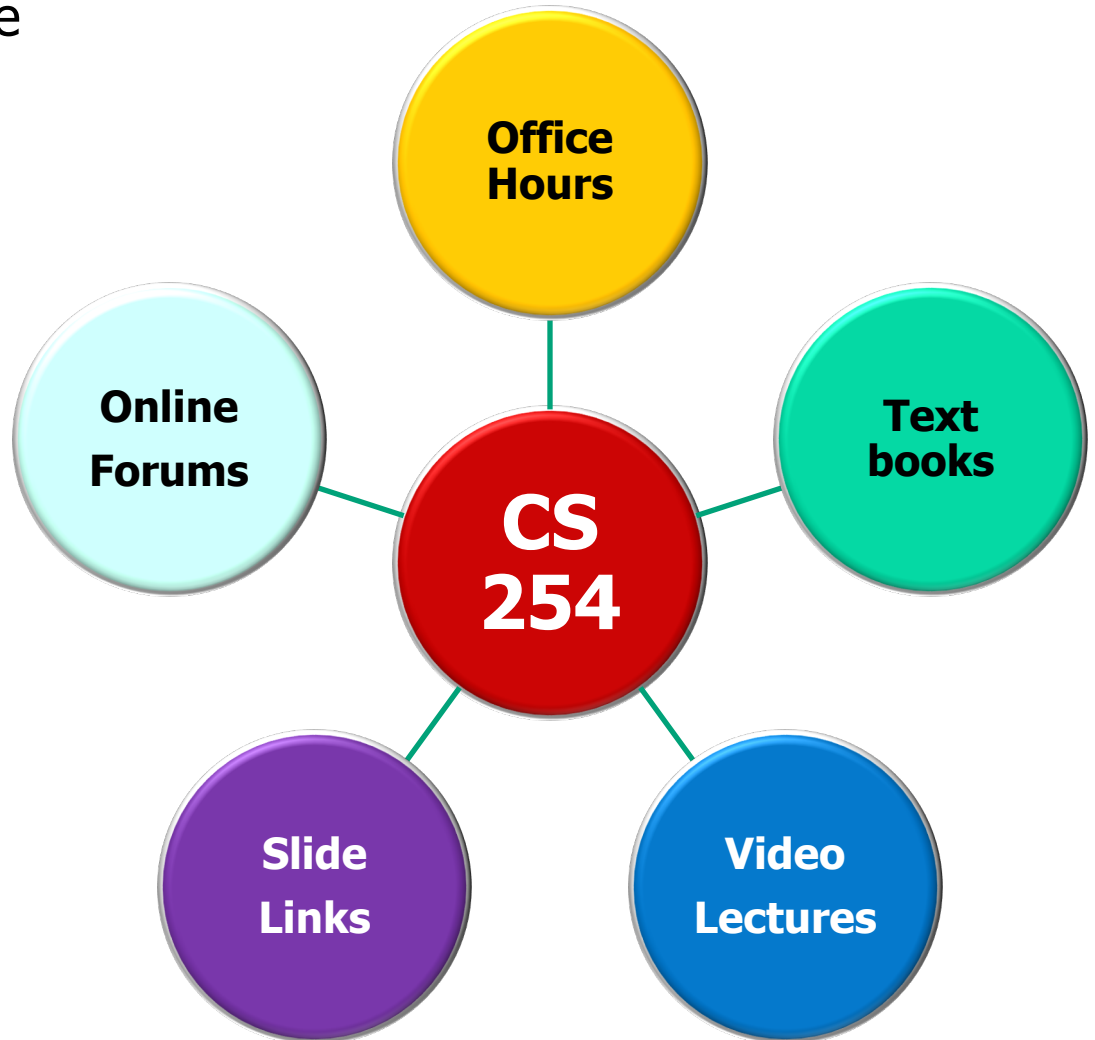
- You will get out of this course what you put into it
 - Be prepared to work hard
 - Do *not* miss deadlines...
- Participate in discussions in class & on piazza



See piazza.com/vanderbilt/spring2022/cs254

Summary

- You will get out of this course what you put into it
 - Be prepared to work hard
 - Do *not* miss deadlines...
 - Participate in discussions in class & on piazza
- Avail yourself of available resources



See www.dre.vanderbilt.edu/~schmidt/cs254

Summary

- You will get out of this course what you put into it
 - Be prepared to work hard
 - Do *not* miss deadlines...
 - Participate in discussions in class & on piazza
- Avail yourself of available resources



Please resist the urge to email me directly unless it's a confidential matter or you'd like to schedule a meeting!

Summary

- There are abundant opportunities!

BROAD CATEGORY	2020 SALARY PROJECTION	2019 SALARY PROJECTION	% CHANGE
Engineering	\$69,961	\$69,188	1.1%
Computer Science	\$67,411	\$67,539	-0.2%
Math & Sciences	\$62,488	\$62,177	0.5%
Business	\$57,939	\$57,657	0.5%
Social Sciences	\$57,425	\$57,310	0.2%
Communications	\$56,484	\$52,056	8.5%
Humanities	\$53,617	\$56,651	-5.4%
Agriculture & Natural Resources	\$53,504	\$55,750	-4.0%



See www.nacweb.org/job-market/compensation/starting-salary-projections-for-top-earning-degrees-level

Summary

- If there's an emergency, pay attention to the escape route!
- See engineering.vanderbilt.edu/about/evacuationplans.php

