CS 253: Parallel Functional Programming w/ Java & Android: Overview & Logistics

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

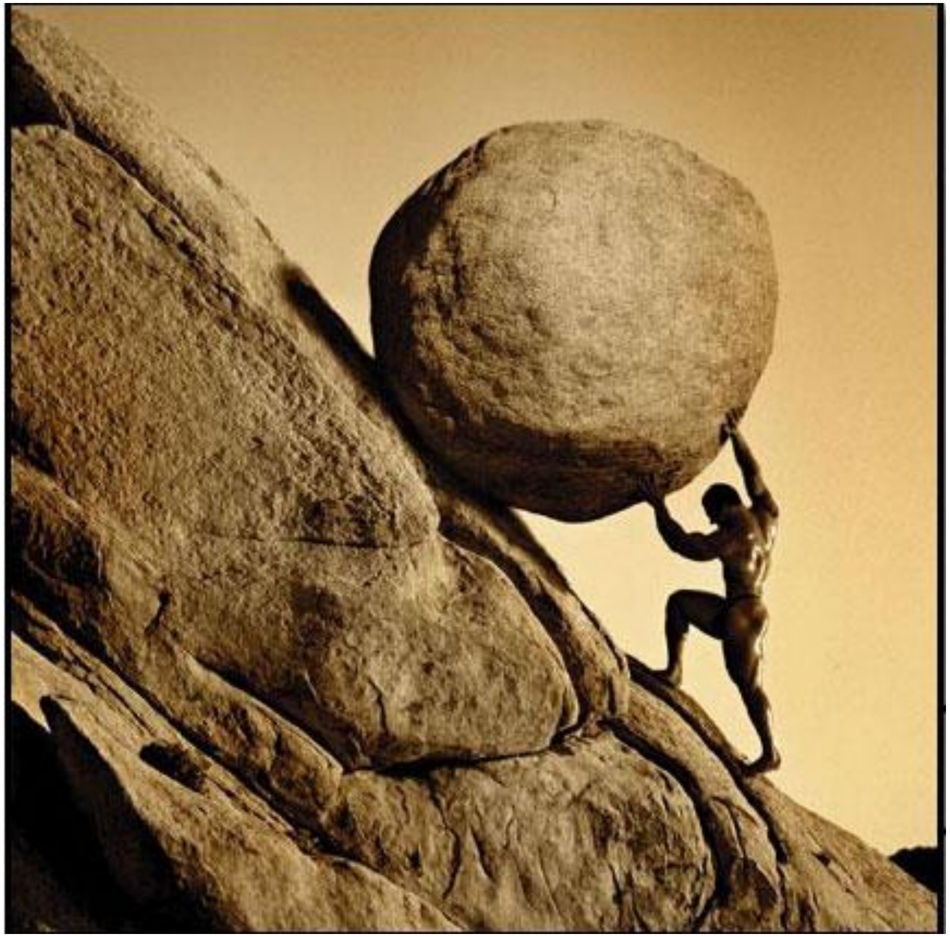
Professor of Computer Science

Institute for Software
Integrated Systems

Vanderbilt University
Nashville, Tennessee, USA

Learning Objectives in this Lesson

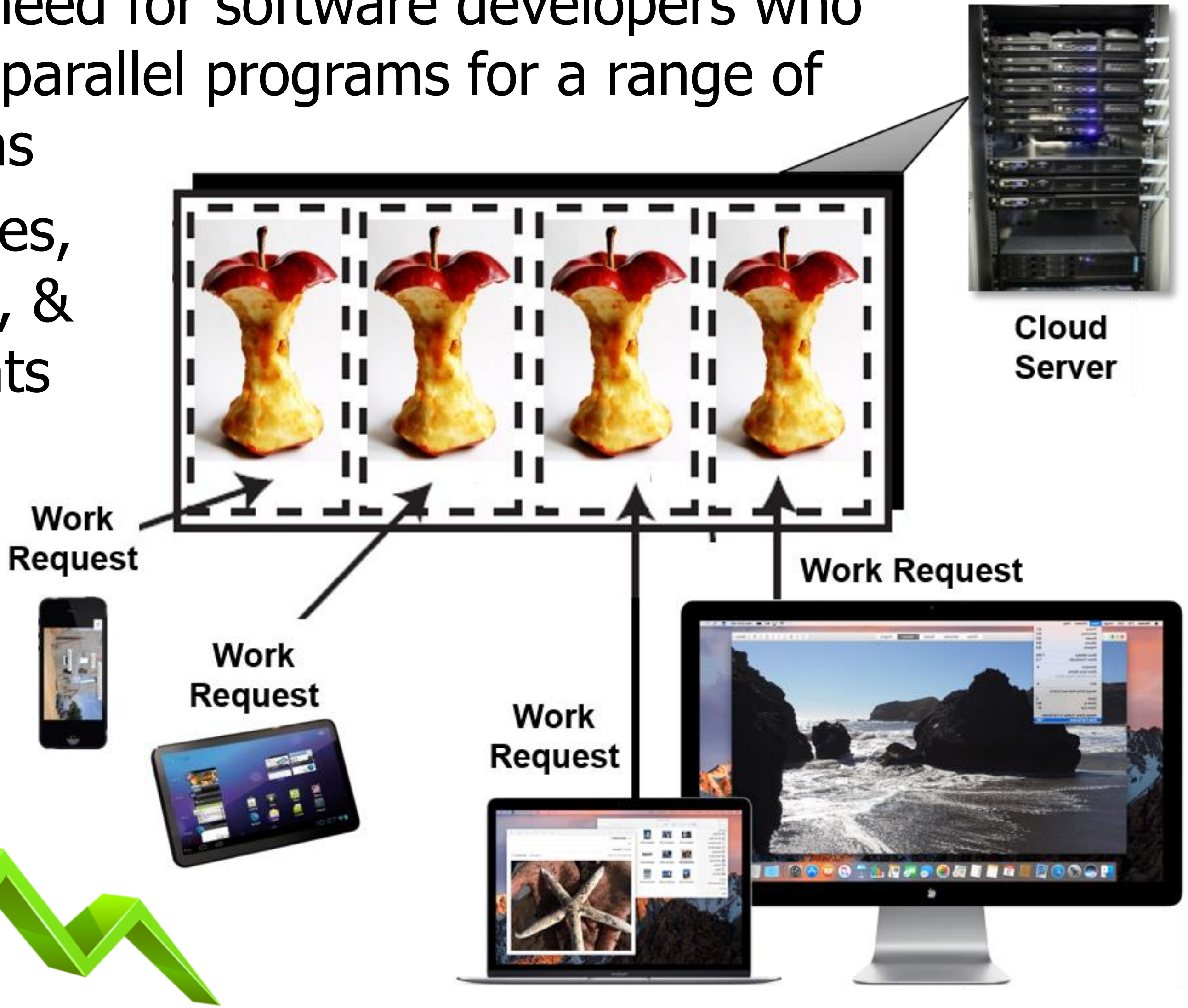
- Understand the course topics & logistics
  - Course philosophy
  - Course contents
  - Structure of the lecture material
  - Overview of the assignments & assessments
  - Setting up the Java & Android IDE on Android Studio
  - Accessing Android & Java source code



Course Philosophy

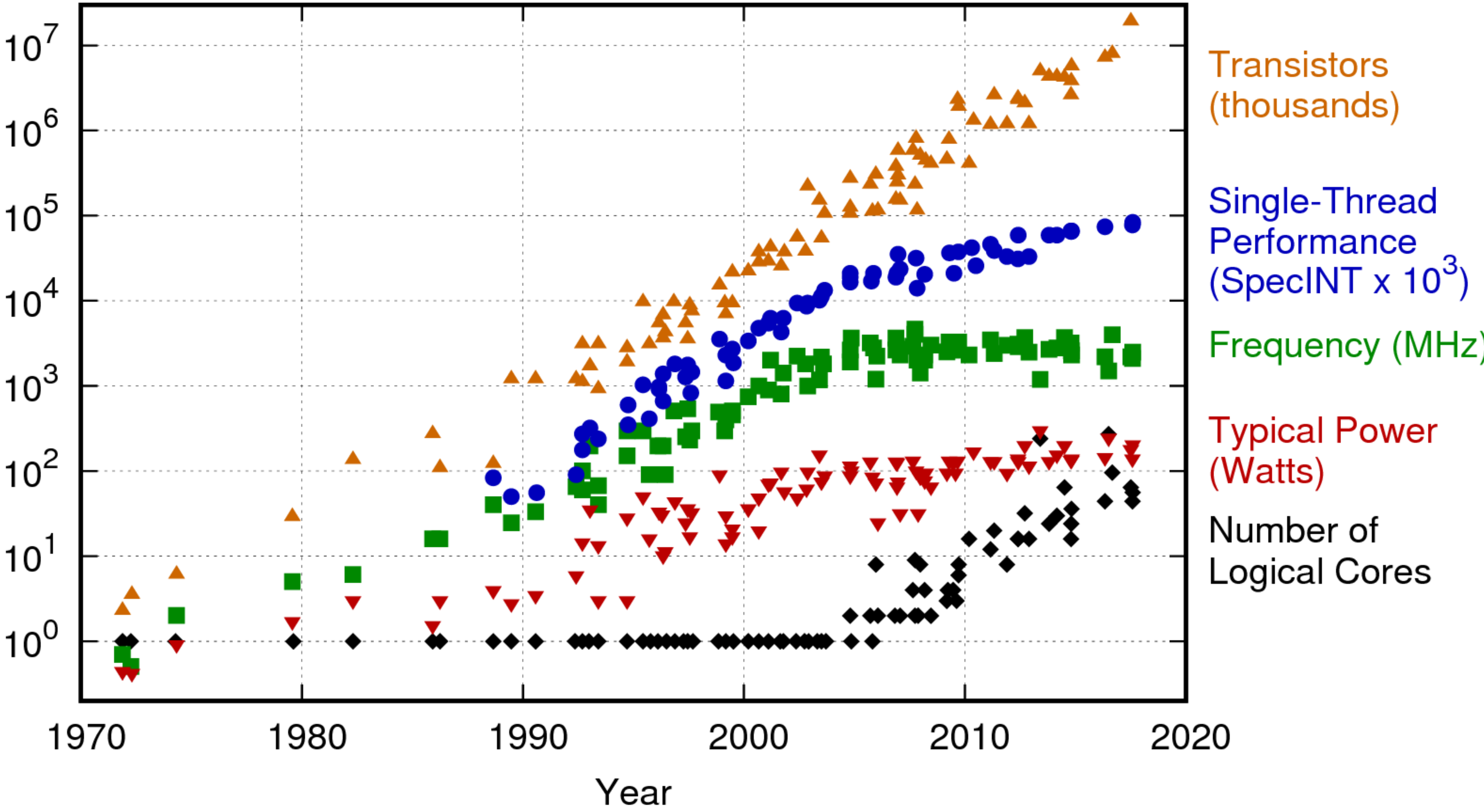
Course Philosophy

- 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

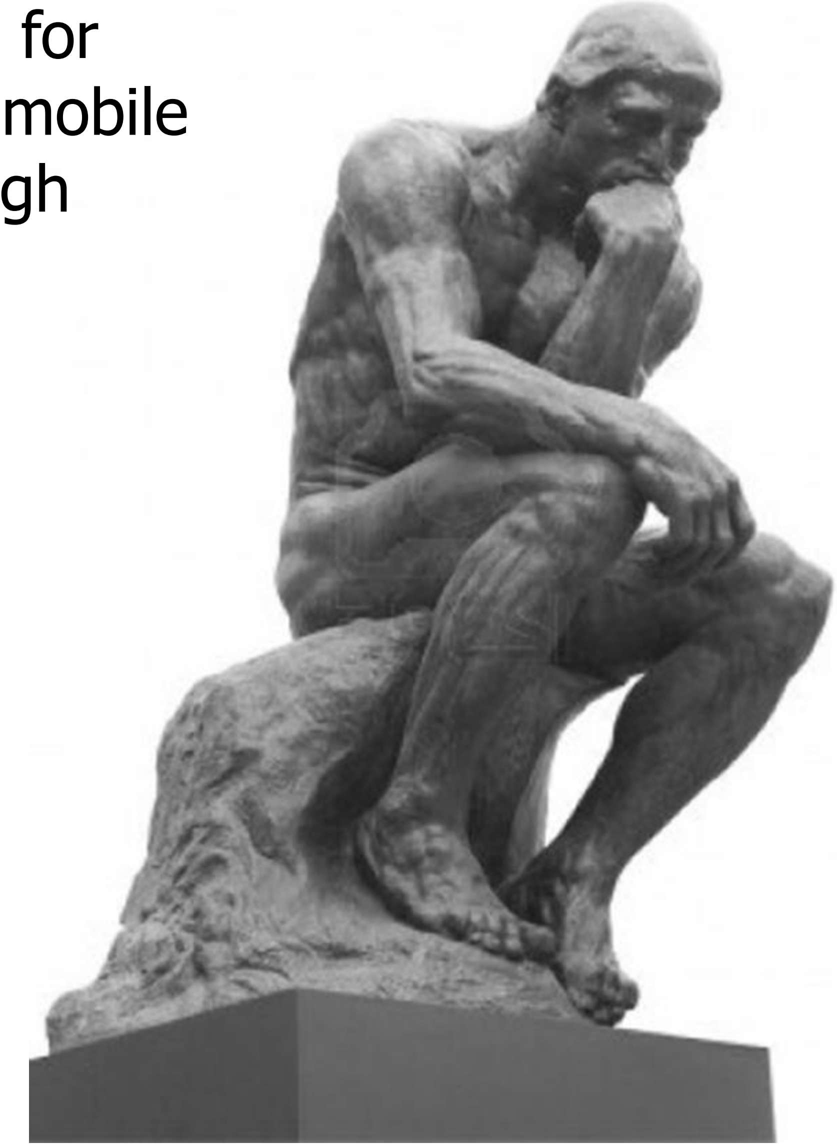
- Demand is driven by software/hardware infrastructure advances



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 & resilient

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

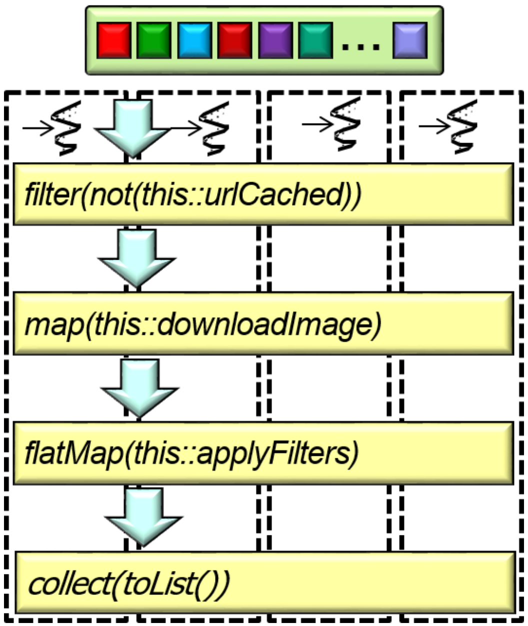
This course involves lots of hands-on software development & testing!

Summary of the Course Contents

Summary of Course Contents

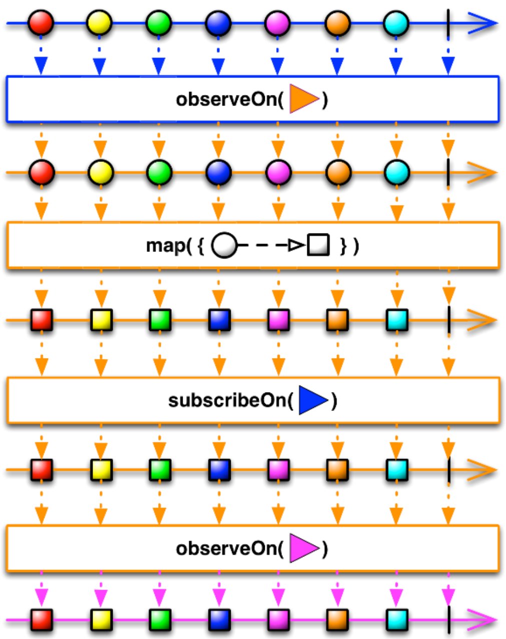
- Key Java parallelism frameworks

Parallel Streams

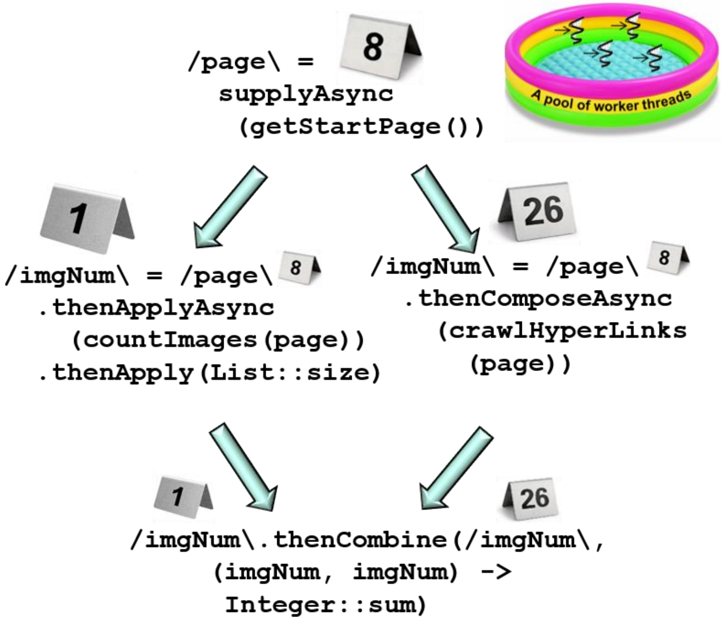


```
filter(not(this::urlCached))
map(this::downloadImage)
flatMap(this::applyFilters)
collect(toList())
```

Reactive Streams



Completable Futures



```
/page\ =
  supplyAsync
    (getStartPage())
```

```
/imgNum\ = /page\
  .thenApplyAsync
    (countImages(page))
  .thenApply(List::size)
```

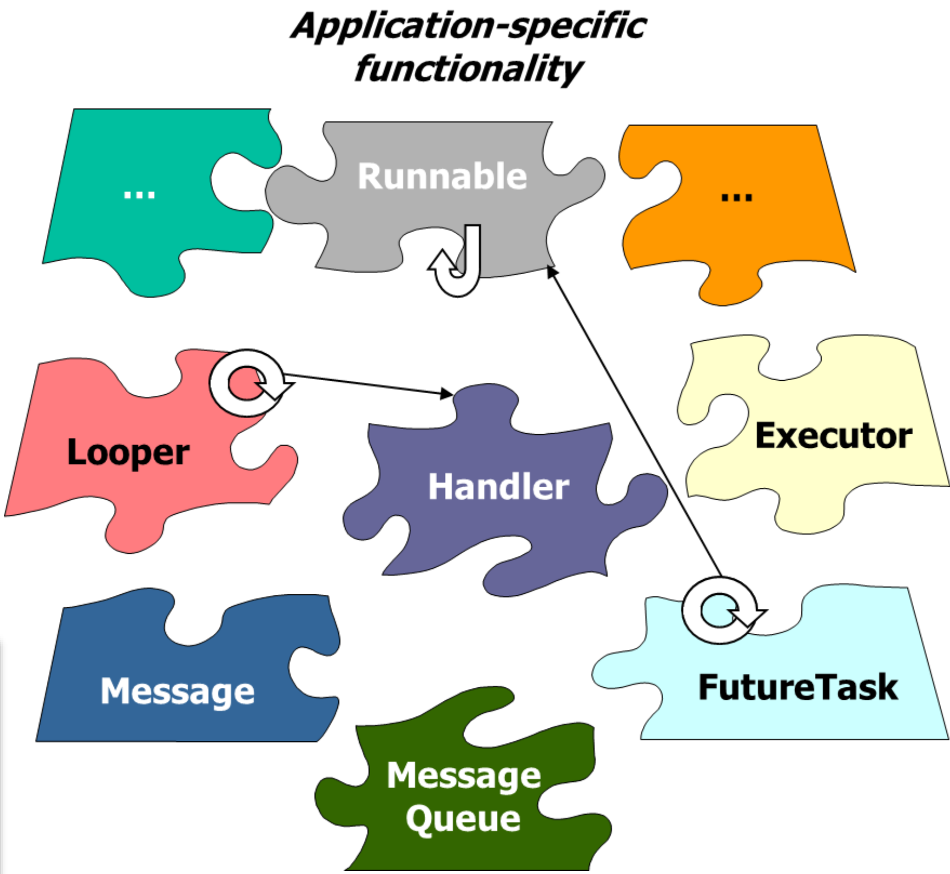
```
/imgNum\ = /page\
  .thenComposeAsync
    (crawlHyperLinks
      (page))
```

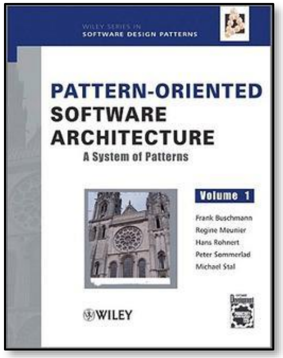
```
/imgNum\.thenCombine(/imgNum\,
  (imgNum, imgNum) ->
    Integer::sum)
```

Also covers Java object-oriented & functional programming language features

Summary of Course Contents

- Key Java parallelism frameworks
- Patterns for parallel 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

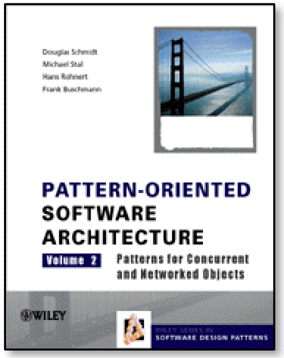
WILEY SERIES IN SOFTWARE DESIGN PATTERNS

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 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) Java, Android, & Git

See www.coursera.org/specializations/android-app-development

Structure of the Lecture Material

Structure of the Lecture Material

- This course has three main modules

Section	Topics
Java functional programming features	• Coverage of Java functional programming features, such as lambda expressions, method references, & functional interfaces • Assume you know Java's support for abstraction, inheritance, & polymorphism

Structure of the Lecture Material

- This course has three main modules

Section	Topics
Java functional programming features	• Coverage of Java functional programming features, such as lambda expressions, method references, & functional interfaces • Assume you know Java's support for abstraction, inheritance, & polymorphism
Java Parallelism	• Coverage of Java 8 parallelism frameworks, e.g. • 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 functional programming features	• Coverage of Java functional programming features, such as lambda expressions, method references, & functional interfaces • Assume you know Java's support for abstraction, inheritance, & polymorphism
Java Parallelism	• Coverage of Java 8 parallelism frameworks, e.g. • Java sequential & parallel streams • Java completable futures • Reactive streams (e.g., RxJava & Project Reactor)
Software Patterns	• 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

We'll leverage YouTube's "progress bar" feature to demarcate segments!

Structure of the Lecture Material

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

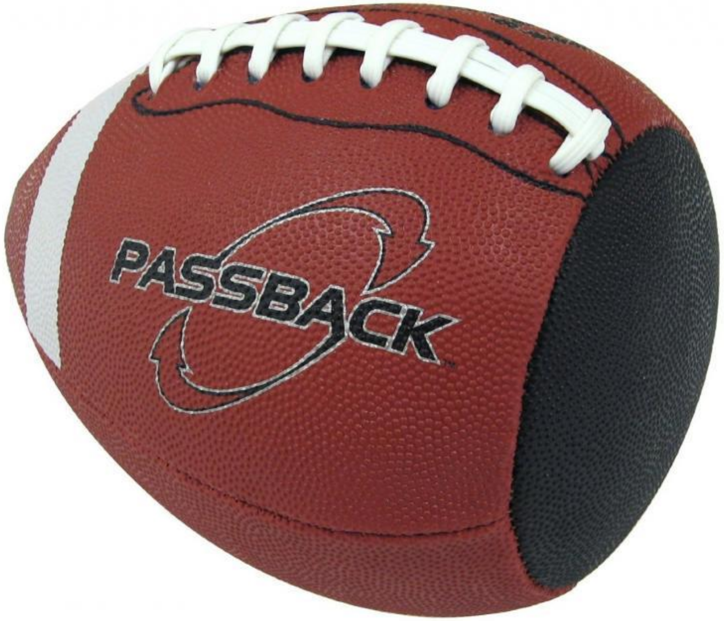
Structure of the Lecture Material

- There will be bi-weekly quizzes on material covered in the lectures
  - 1st quiz will be on Wednesday, September 2nd

All quizzes are "closed book" & 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, September 2nd
  - 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, September 2nd
  - 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

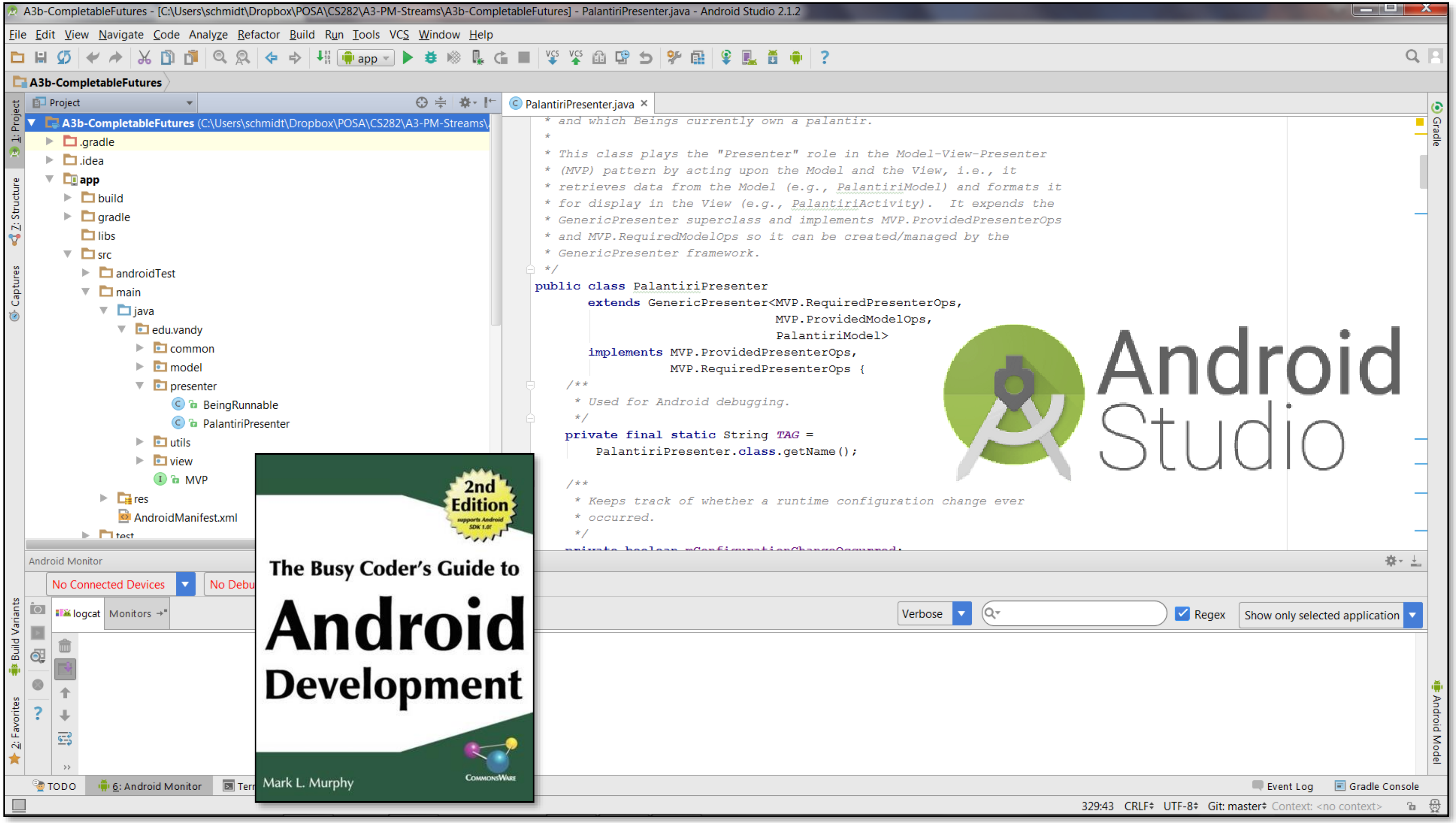
- There *may* be a cumulative final exam that covers all the lectures
  - The focus will be on the last weeks of the semester

The final exam is 2 to 5pm, Tuesday, December 10th via Brightspace

Overview of Assignments & Assessments

Overview of Assignments & Assessments

- Programming assignments should be written in Java 8 using Android Studio



```java
 * and which Beings currently own a palantir.
 *
 * This class plays the "Presenter" role in the Model-View-Presenter
 * (MVP) pattern by acting upon the Model and the View, i.e., it
 * retrieves data from the Model (e.g., PalantiriModel) and formats it
 * for display in the View (e.g., PalantiriActivity).  It expends the
 * GenericPresenter superclass and implements MVP.ProvidedPresenterOps
 * and MVP.RequiredModelOps so it can be created/managed by the
 * GenericPresenter framework.
 */
public class PalantiriPresenter
       extends GenericPresenter<MVP.RequiredPresenterOps,
                                MVP.ProvidedModelOps,
                                PalantiriModel>
       implements MVP.ProvidedPresenterOps,
                  MVP.RequiredPresenterOps {
    /**
     * Used for Android debugging.
     */
    private final static String TAG =
        PalantiriPresenter.class.getName();

    /**
     * Keeps track of whether a runtime configuration change ever
     * occurred.
     */
    private boolean mConfigurationChangeOccurred;
```

You can use any IDE, but your final submission *must* build & run with the latest Android Studio & Android 11/10+ (API 30)

Overview of Assignments & Assessments

- Programming assignments should be written in Java 8 using Android Studio
  - The Java 8 runtime environment (JRE) is pre-installed with Android

See github.com/douglascraigschmidt/CS253/wiki/Installing-Software

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

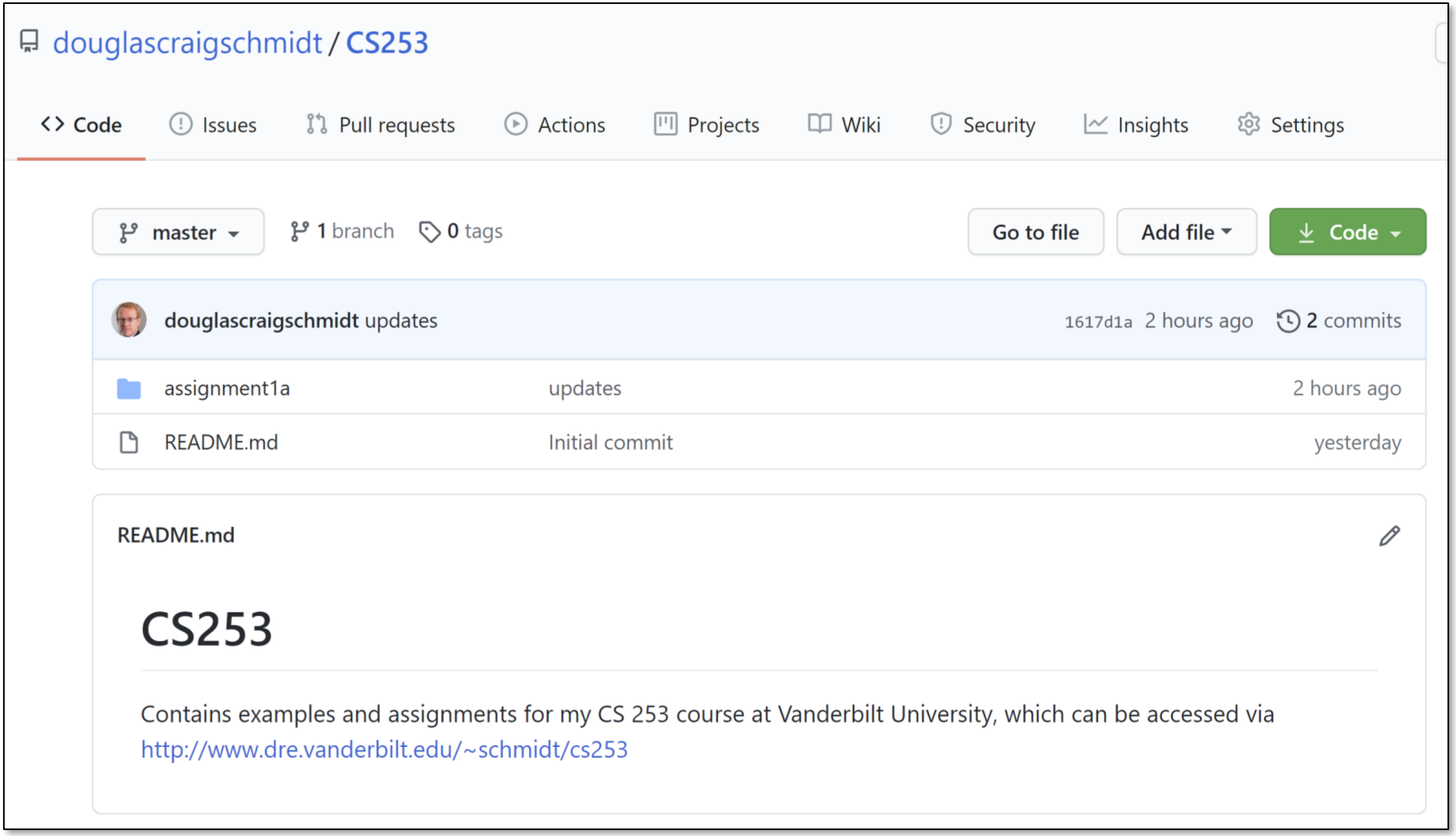
Overview of Assignments & Assessments

- Android programming assignments must be submitted using Android Studio
  - Please install Android 11/10+ (API level 30)

See en.wikipedia.org/wiki/Android_11

Overview of Assignments & Assessments

- All source code for assignments & examples available at GitHub



douglascraigschmidt / CS253

douglascraigschmidt updates 1617d1a 2 hours ago 2 commits

assignment1a	updates	2 hours ago
README.md	Initial commit	yesterday

README.md

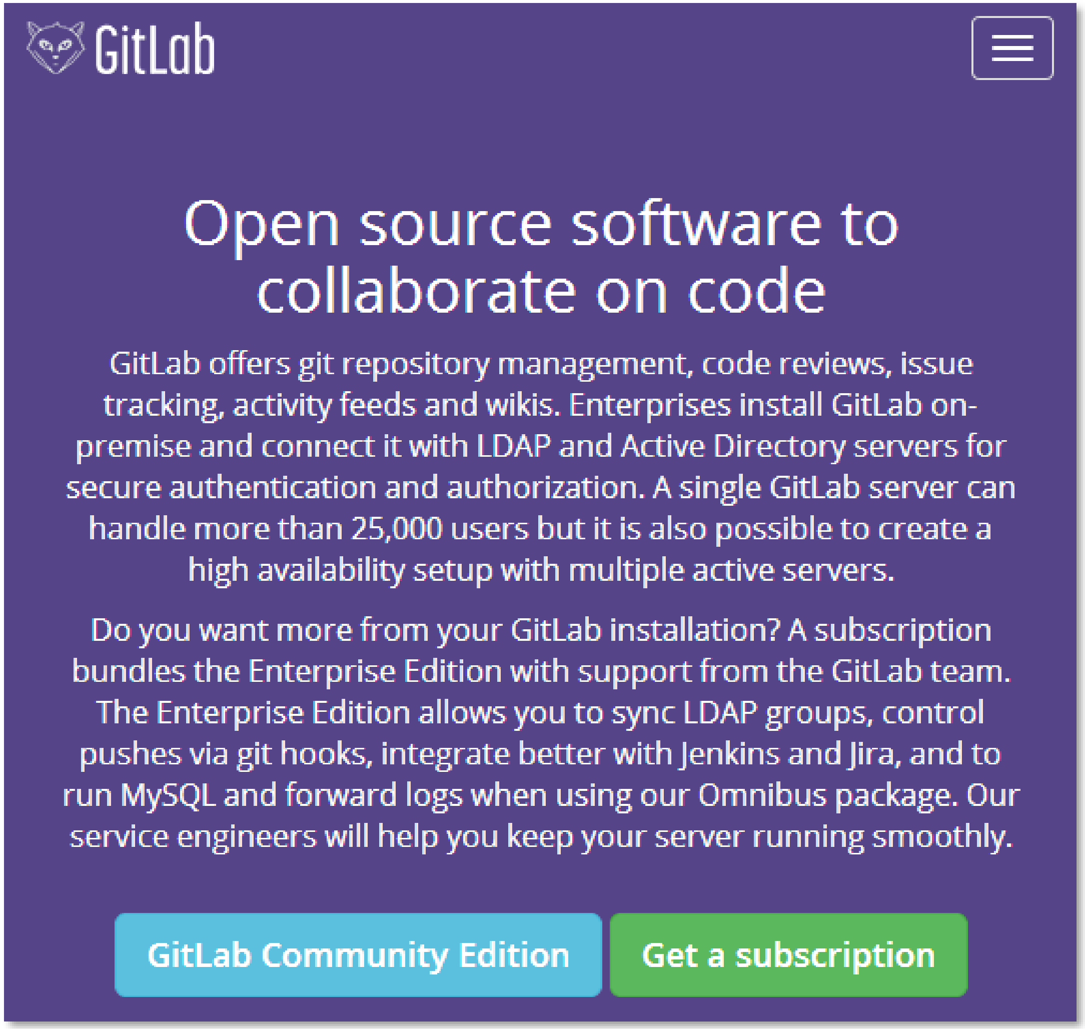
CS253

Contains examples and assignments for my CS 253 course at Vanderbilt University, which can be accessed via http://www.dre.vanderbilt.edu/~schmidt/cs253

Go to GitHub at github.com/douglascraigschmidt/CS253

Overview of Assignments & Assessments

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



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.

Overview of Assignments & Assessments

- All source code for assignments & exam
  - You will need to learn how to use GitLab et al.
  - Be prepared to update your repositories occasionally

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

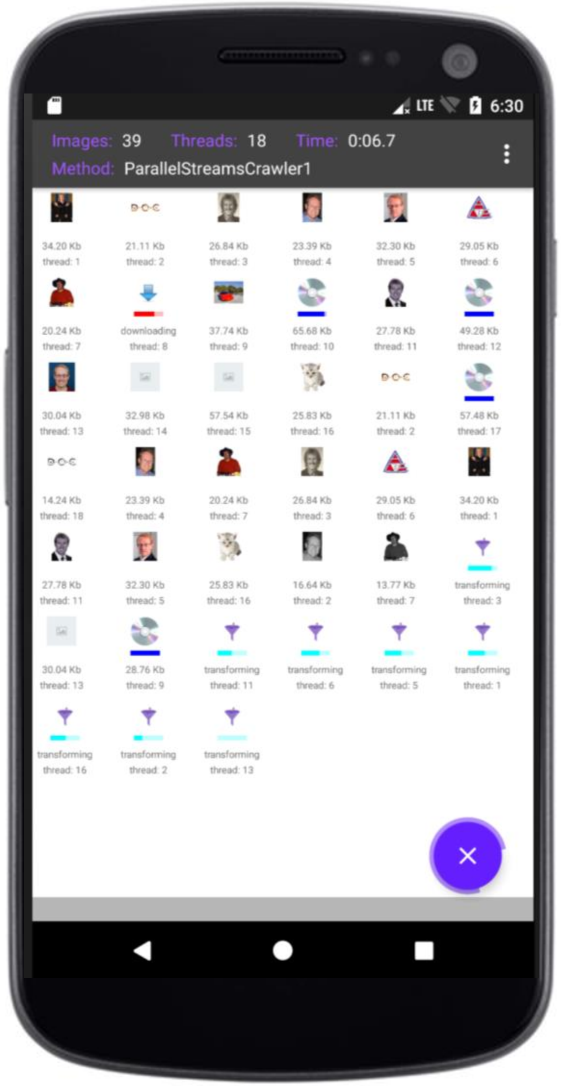
Overview of Assignments & Assessments

- Assignments will provide a range of experience with Java 8 & Android parallel programs

Go to GitHub at github.com/douglascraigschmidt/CS253

Overview of Assignments & Assessments

- Assignments will provide a range of experience with Java 8 & Android parallel programs
  - Implement an image crawler app on Android using various Java 8 features, e.g.
    - Java lambda expressions, method references, & functional interfaces
    - Java sequential streams
    - Java parallel streams
    - Java completable futures
    - Java reactive streams



Images: 39 Threads: 18 Time: 0:06.7
Method: ParallelStreamsCrawler1

34.20 Kb thread: 1	21.11 Kb thread: 2	26.84 Kb thread: 3	23.39 Kb thread: 4	32.30 Kb thread: 5	29.05 Kb thread: 6
20.24 Kb thread: 7	downloading thread: 8	37.74 Kb thread: 9	65.68 Kb thread: 10	27.78 Kb thread: 11	49.28 Kb thread: 12
30.04 Kb thread: 13	32.98 Kb thread: 14	57.54 Kb thread: 15	25.83 Kb thread: 16	21.11 Kb thread: 2	57.48 Kb thread: 17
14.24 Kb thread: 18	23.39 Kb thread: 4	20.24 Kb thread: 7	26.84 Kb thread: 3	29.05 Kb thread: 6	34.20 Kb thread: 1
27.78 Kb thread: 11	32.30 Kb thread: 5	25.83 Kb thread: 16	16.64 Kb thread: 2	13.77 Kb thread: 7	transforming thread: 3
30.04 Kb thread: 13	28.76 Kb thread: 9	transforming thread: 11	transforming thread: 6	transforming thread: 5	transforming thread: 1
transforming thread: 16	transforming thread: 2	transforming thread: 13			

The topics covered by the assignments may change during the semester

Overview of Assignments & Assessments

- Assignment assessments will be done via reviews by course staff

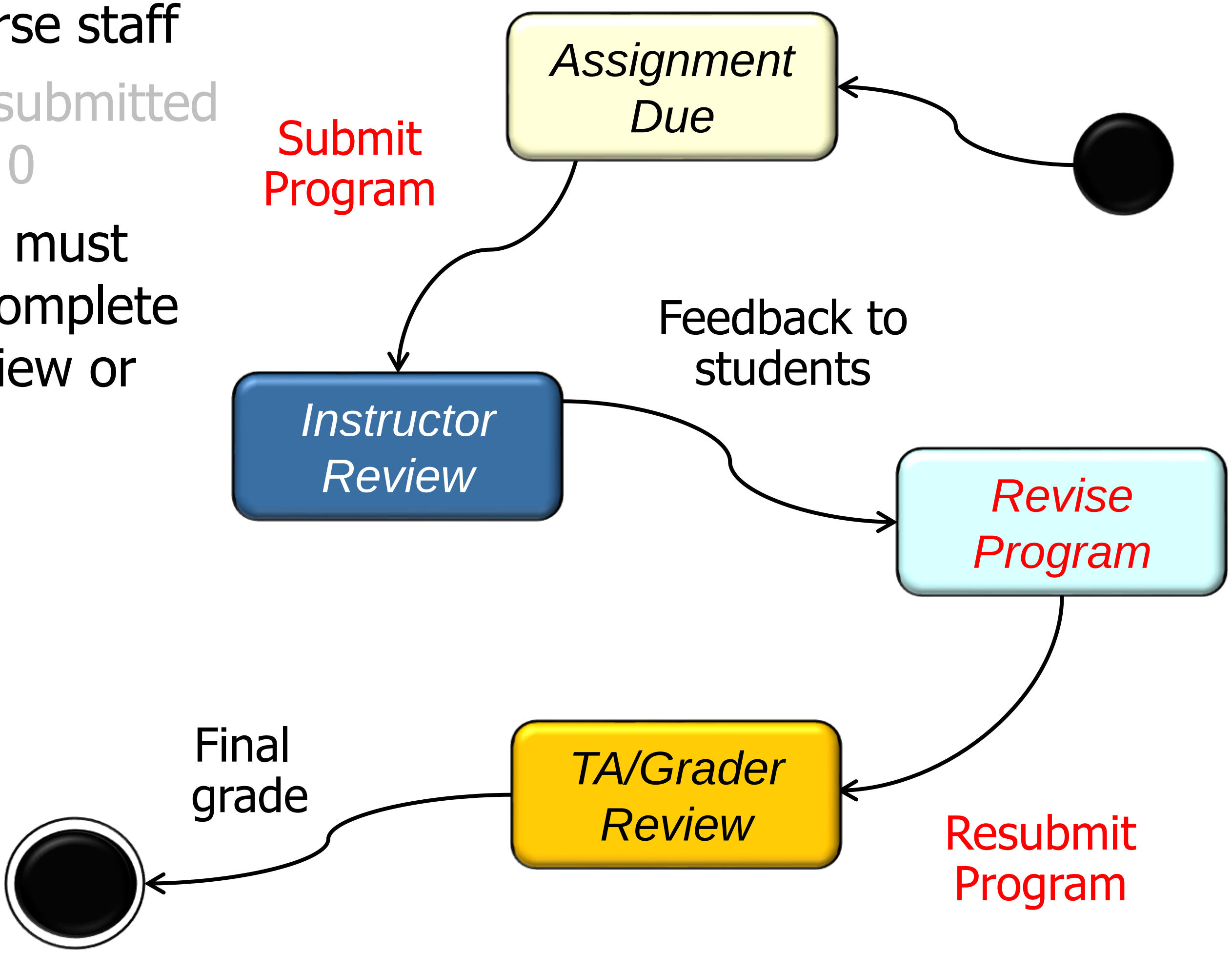
Overview of Assignments & Assessments

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

See github.com/douglascraigschmidt/CS253/wiki/CS-253-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



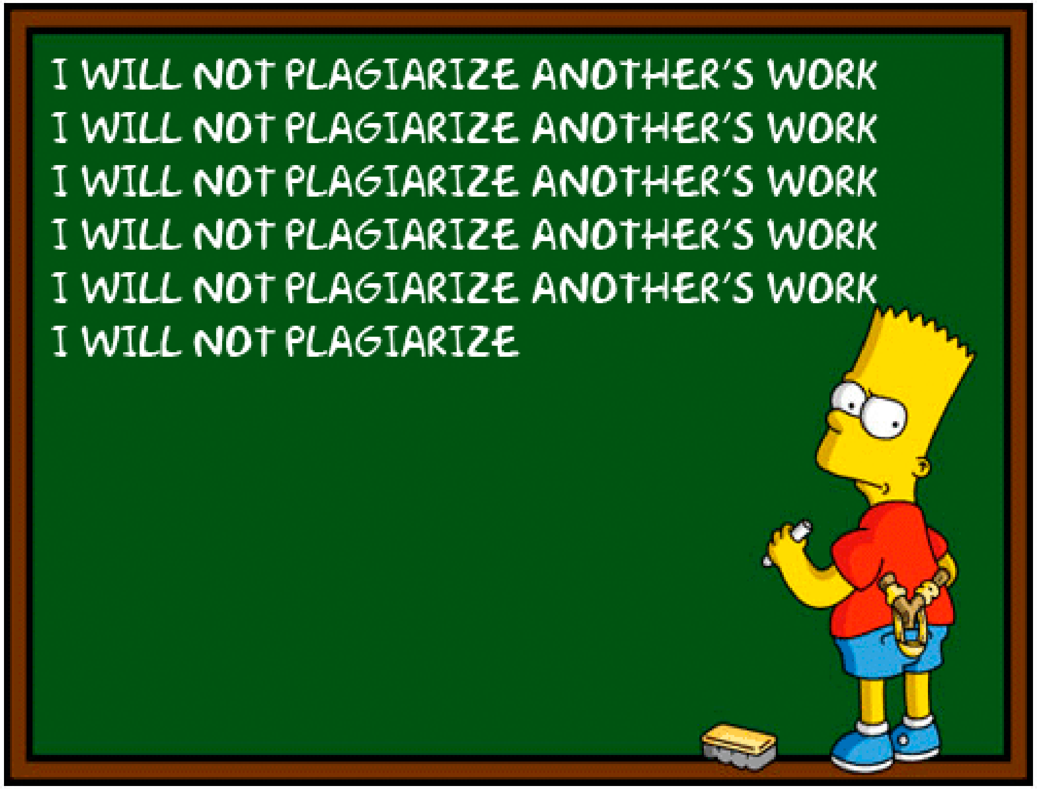
Overview of Assignments & Assessments

- 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/cs253/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 goes for quizzes & programming assignments


I WILL NOT PLAGIARIZE ANOTHER'S WORK
I WILL NOT PLAGIARIZE ANOTHER'S WORK
I WILL NOT PLAGIARIZE ANOTHER'S WORK
I WILL NOT PLAGIARIZE ANOTHER'S WORK
I WILL NOT PLAGIARIZE ANOTHER'S WORK
I WILL NOT PLAGIARIZE

www.vanderbilt.edu/student_handbook/the-honor-system#statement-of-the-honor-code

Overview of Assignments & Assessments

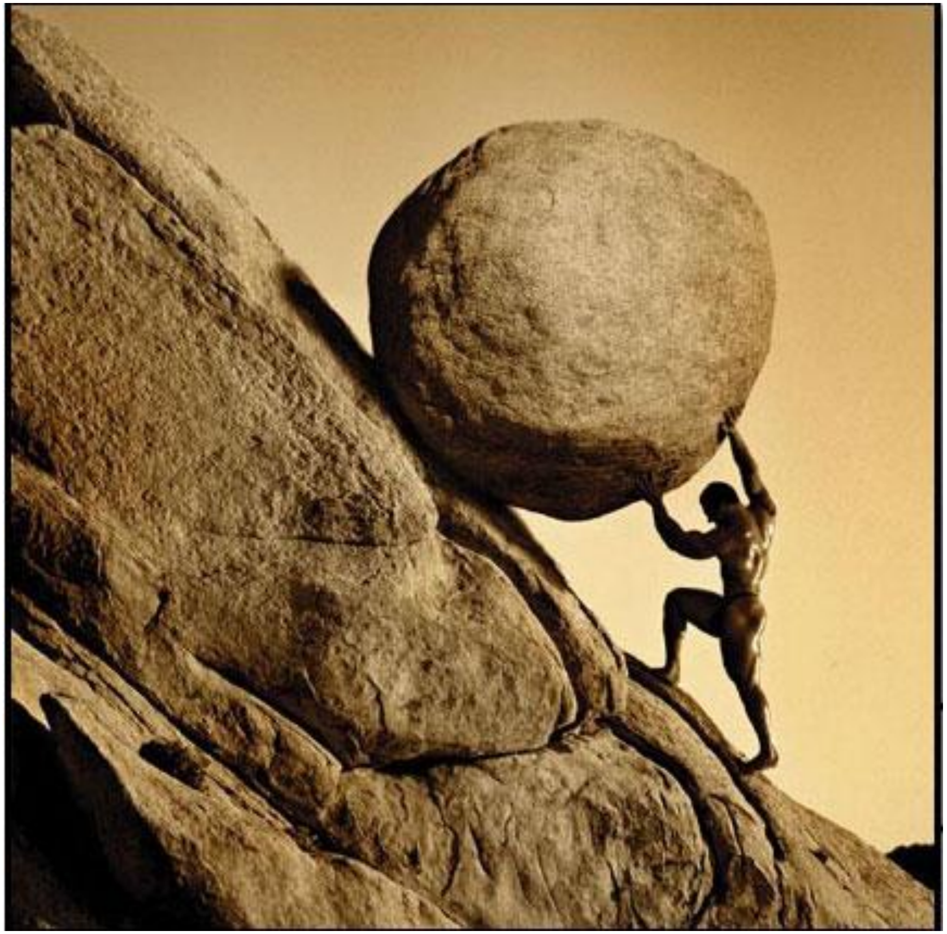
- Assessment criteria

Assessment Category	%
Execution correctness	40%
Structure (e.g., modularization, information hiding, etc.)	30%
Insightful programming (e.g., developing reusable class components, etc.)	10%
Consistent style (e.g., capitalization, indenting, etc.)	10%
Appropriate commenting style	10%

See www.dre.vanderbilt.edu/~schmidt/cs253/assignments.html

Overview of Assignments & Assessments

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



These weightings may change, depending on various factors

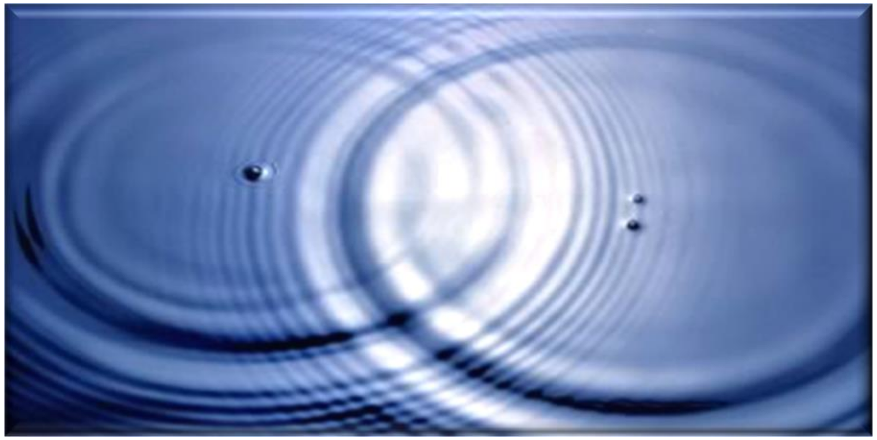
Overview of Assignments & Assessments

- The relative weighting of each portion of the course is:
  - 40% Quizzes
  - 40% Programming projects
  - 10% Final exam
  - 10% Participation
    - Participation is ~5% attendence & involvement & ~5% "following directions"

Overview of Assignments & Assessments

- The relative weighting of each portion of the course is:
  - 40% Quizzes
  - 40% Programming projects
  - 10% Final exam
  - 10% Participation
    - Participation is ~5% attendence & involvement & ~5% "following directions"

Attendance also affects other aspects of your quiz & assignment grades



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

Overview of Assignments & Assessments

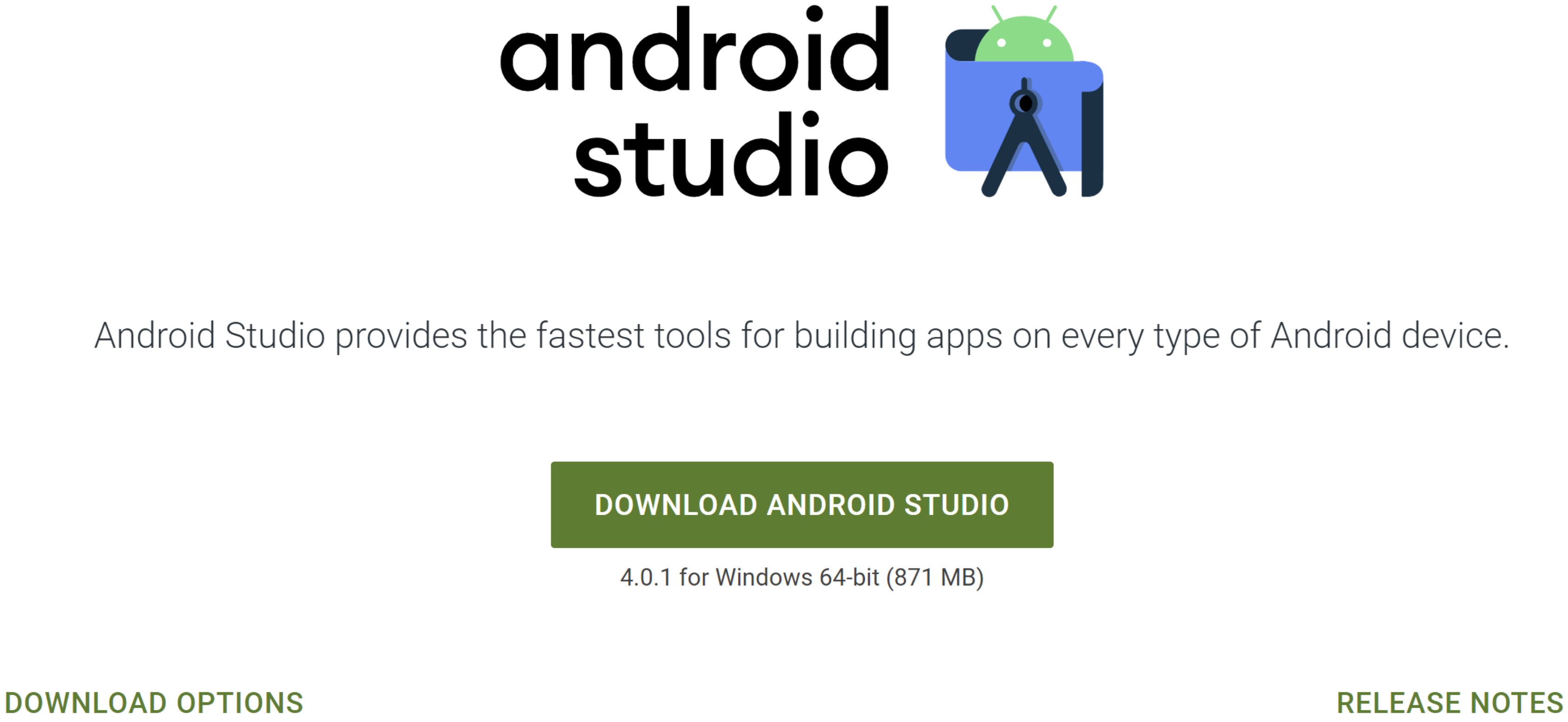
- The relative weighting of each portion of the course is:
  - 40% Quizzes
  - 40% Programming projects
  - 10% Final exam
  - 10% Participation
    - Participation is ~5% attendence & involvement & ~5% “following directions”

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 Eclipse 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.

4.0.1 for Windows 64-bit (871 MB)

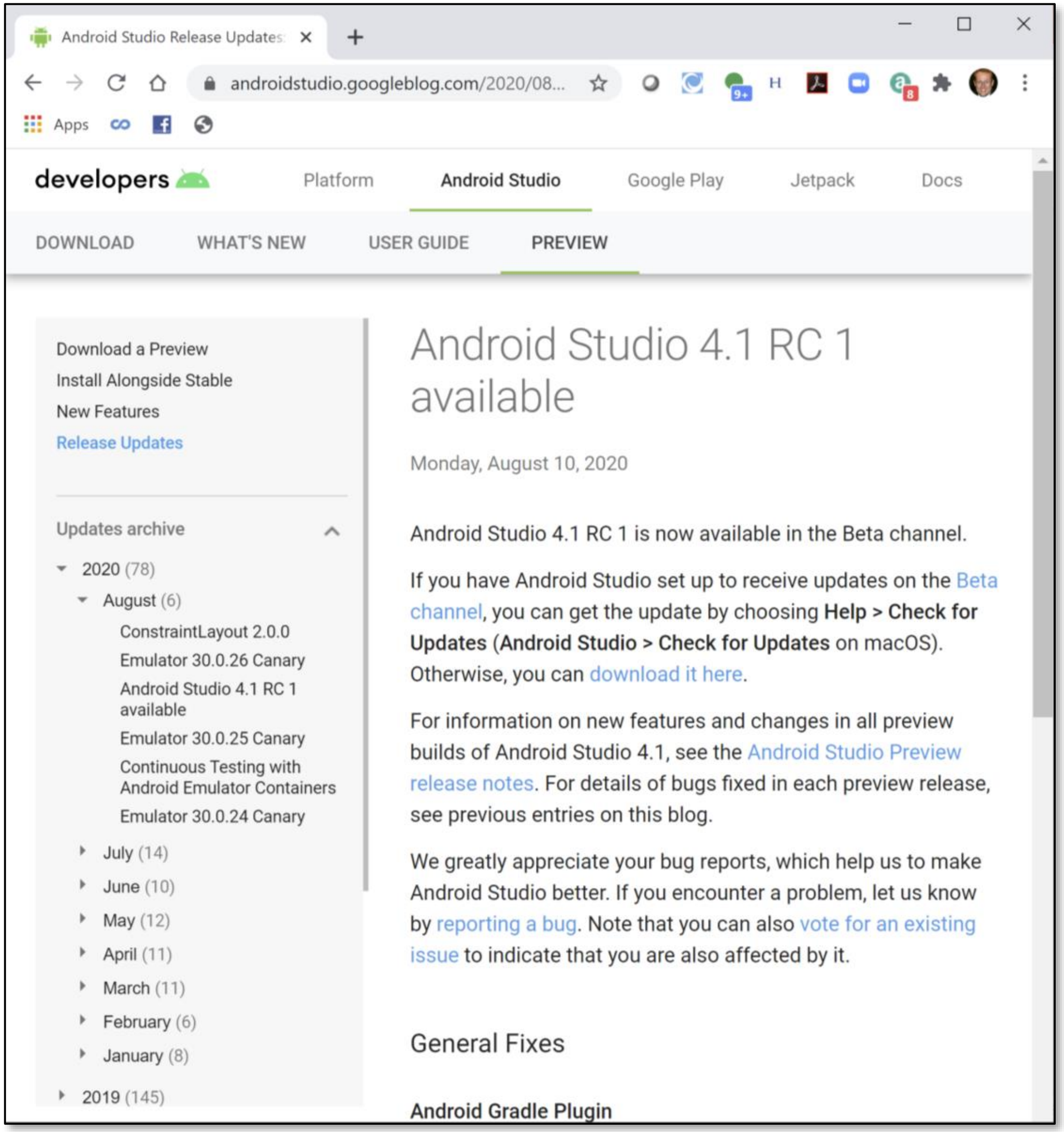
See developer.android.com/studio

Installing Eclipse Java/Android Developer Tools

- Installation steps

Installing Eclipse Java/Android Developer Tools

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



Android Studio 4.1 RC 1 available

Monday, August 10, 2020

Android Studio 4.1 RC 1 is now available in the Beta channel.

If you have Android Studio set up to receive updates on the Beta channel, you can get the update by choosing **Help > Check for Updates** (**Android Studio > Check for Updates** on macOS). Otherwise, you can download it here.

For information on new features and changes in all preview builds of Android Studio 4.1, see the Android Studio Preview release notes. For details of bugs fixed in each preview release, see previous entries on this blog.

We greatly appreciate your bug reports, which help us to make Android Studio better. If you encounter a problem, let us know by reporting a bug. Note that you can also vote for an existing issue to indicate that you are also affected by it.

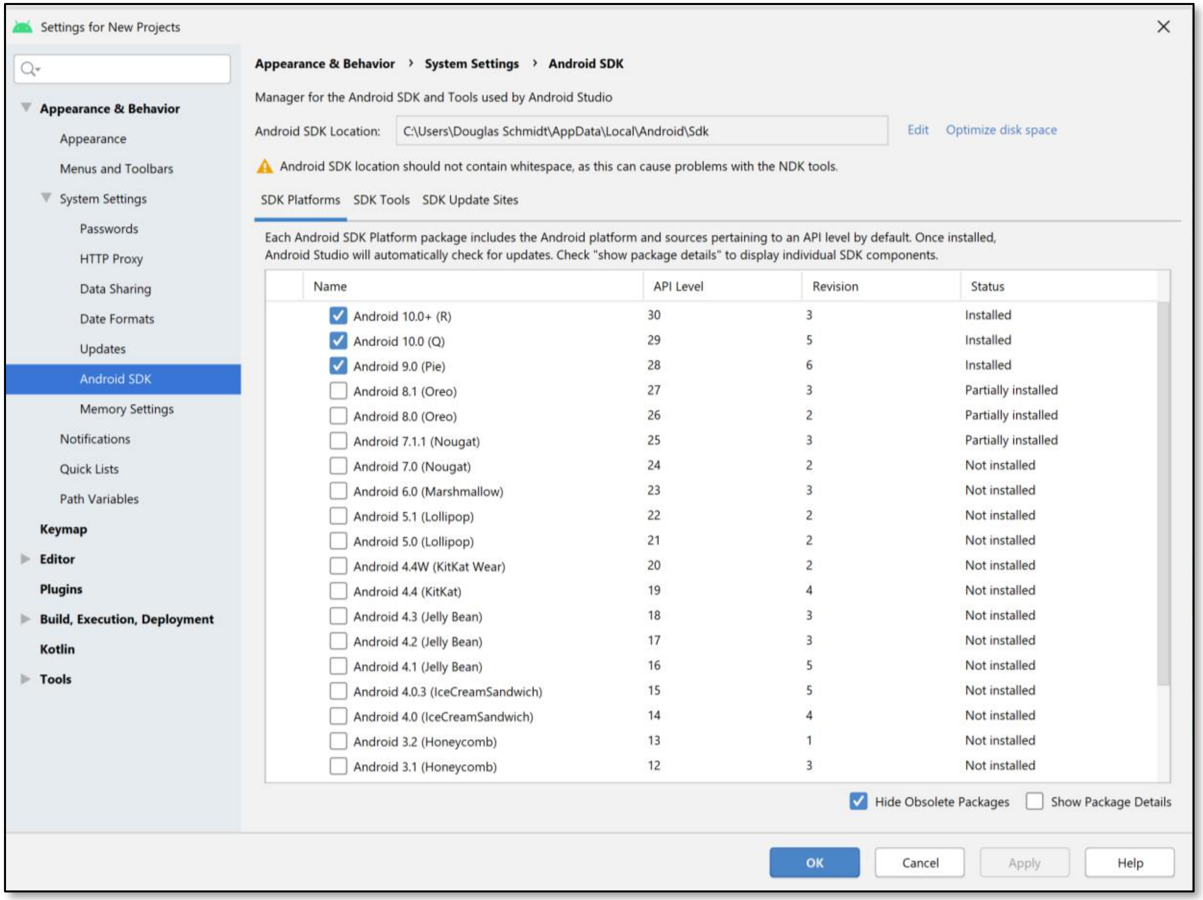
General Fixes

Android Gradle Plugin

See developer.android.com/studio

Add Components to the SDK

- Launch the Android Studio SDK Manager
  - Select the "R" version of Android (11/10+, API 30)



Appearance & Behavior › System Settings › Android SDK

Manager for the Android SDK and Tools used by Android Studio

Android SDK Location: C:\Users\Douglas Schmidt\AppData\Local\Android\Sdk

⚠ Android SDK location should not contain whitespace, as this can cause problems with the NDK tools.

Each Android SDK Platform package includes the Android platform and sources pertaining to an API level by default. Once installed, Android Studio will automatically check for updates. Check "show package details" to display individual SDK components.

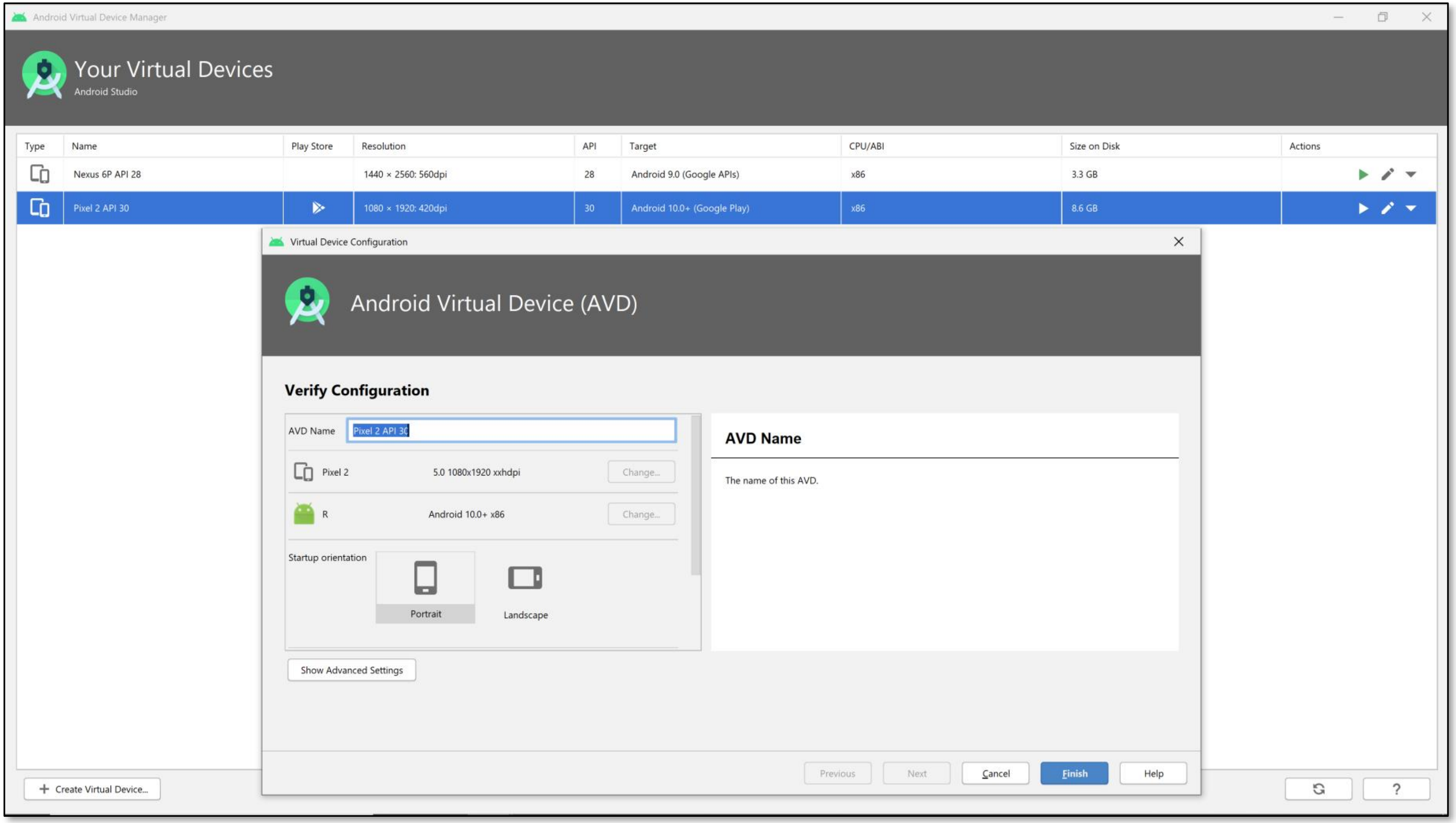
	Name	API Level	Revision	Status
☑	Android 10.0+ (R)	30	3	Installed
☑	Android 10.0 (Q)	29	5	Installed
☑	Android 9.0 (Pie)	28	6	Installed
□	Android 8.1 (Oreo)	27	3	Partially installed
□	Android 8.0 (Oreo)	26	2	Partially installed
□	Android 7.1.1 (Nougat)	25	3	Partially installed
□	Android 7.0 (Nougat)	24	2	Not installed
□	Android 6.0 (Marshmallow)	23	3	Not installed
□	Android 5.1 (Lollipop)	22	2	Not installed
□	Android 5.0 (Lollipop)	21	2	Not installed
□	Android 4.4W (KitKat Wear)	20	2	Not installed
□	Android 4.4 (KitKat)	19	4	Not installed
□	Android 4.3 (Jelly Bean)	18	3	Not installed
□	Android 4.2 (Jelly Bean)	17	3	Not installed
□	Android 4.1 (Jelly Bean)	16	5	Not installed
□	Android 4.0.3 (IceCreamSandwich)	15	5	Not installed
□	Android 4.0 (IceCreamSandwich)	14	4	Not installed
□	Android 3.2 (Honeycomb)	13	1	Not installed
□	Android 3.1 (Honeycomb)	12	3	Not installed

☑ Hide Obsolete Packages □ Show Package Details

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



Android Virtual Device Manager

Your Virtual Devices
Android Studio

Type	Name	Play Store	Resolution	API	Target	CPU/ABI	Size on Disk	Actions
	Nexus 6P API 28		1440 × 2560: 560dpi	28	Android 9.0 (Google APIs)	x86	3.3 GB	▶ ✎ ▼
	Pixel 2 API 30	▶	1080 × 1920: 420dpi	30	Android 10.0+ (Google Play)	x86	8.6 GB	▶ ✎ ▼

Virtual Device Configuration

Android Virtual Device (AVD)

Verify Configuration

AVD Name: Pixel 2 API 30

Pixel 2 — 5.0 1080x1920 xxhdpi — Change...

R — Android 10.0+ x86 — Change...

Startup orientation: Portrait / Landscape

Show Advanced Settings

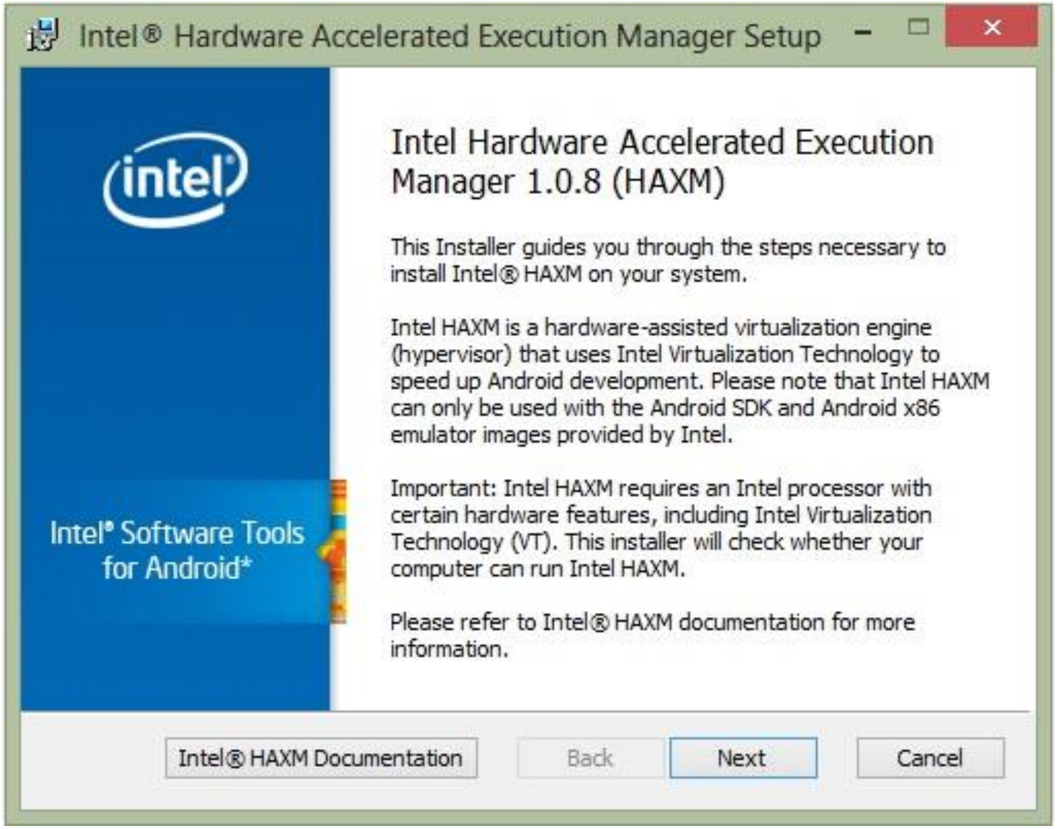
AVD Name

The name of this AVD.

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



Intel® Hardware Accelerated Execution Manager Setup

Intel Hardware Accelerated Execution Manager 1.0.8 (HAXM)

This Installer guides you through the steps necessary to install Intel® HAXM on your system.

Intel HAXM is a hardware-assisted virtualization engine (hypervisor) that uses Intel Virtualization Technology to speed up Android development. Please note that Intel HAXM can only be used with the Android SDK and Android x86 emulator images provided by Intel.

Important: Intel HAXM requires an Intel processor with certain hardware features, including Intel Virtualization Technology (VT). This installer will check whether your computer can run Intel HAXM.

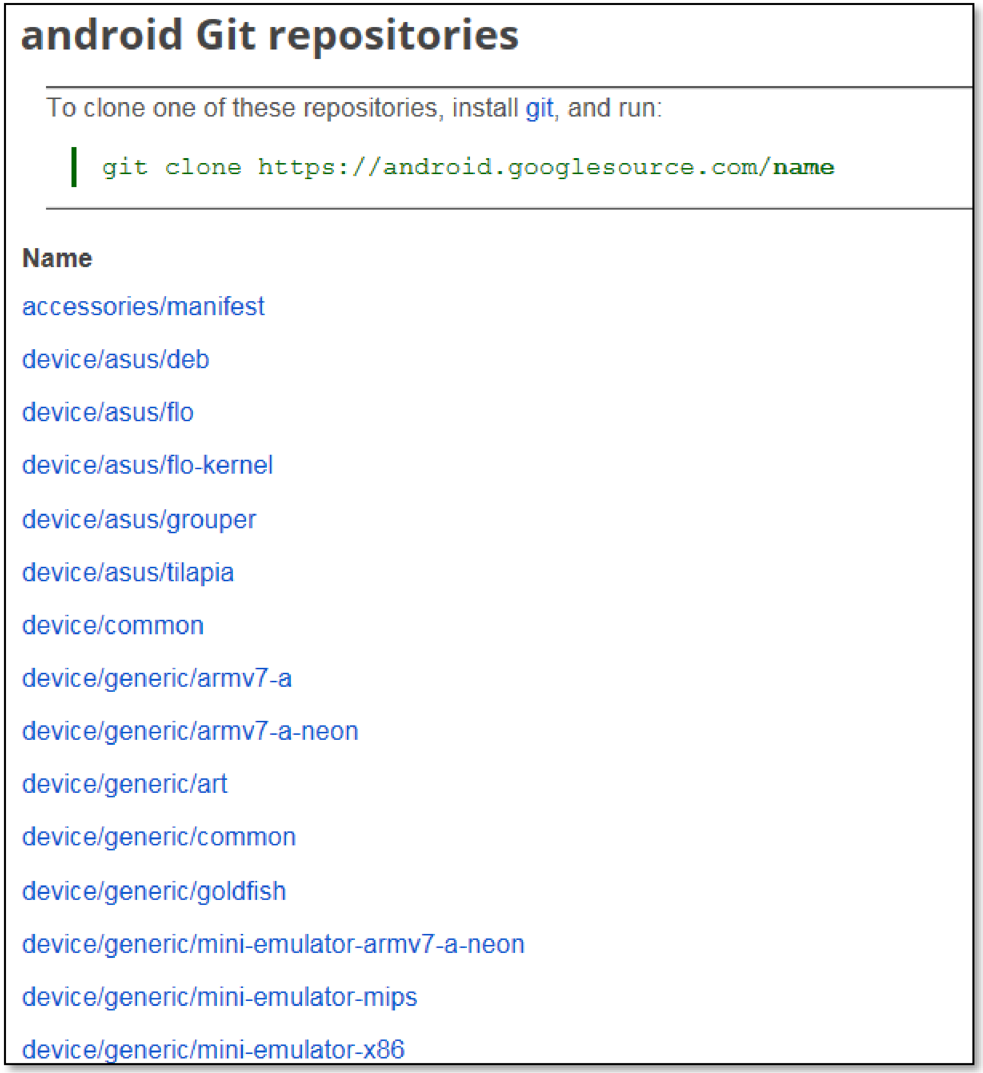
Please refer to Intel® HAXM documentation for more information.

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

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:

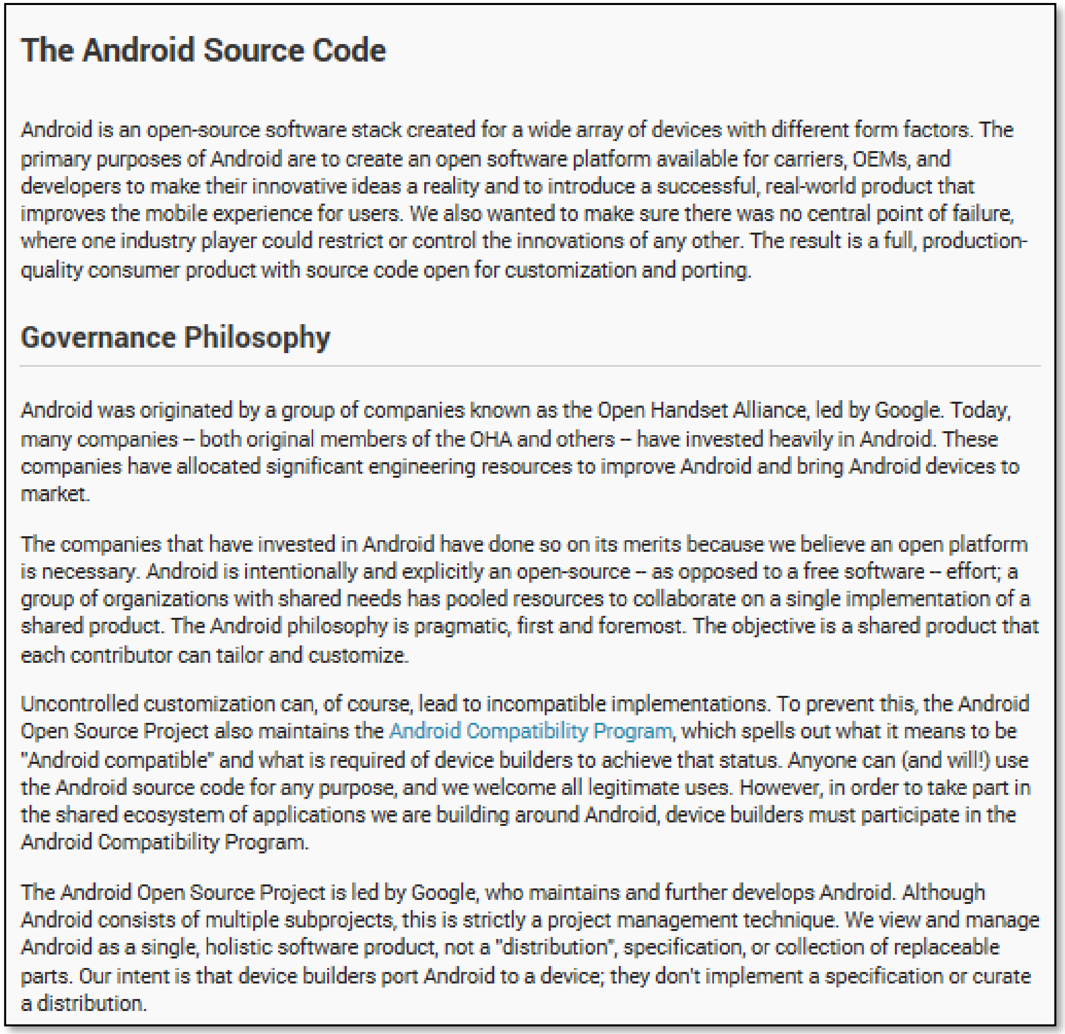
```
git clone https://android.googlesource.com/name
```

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.

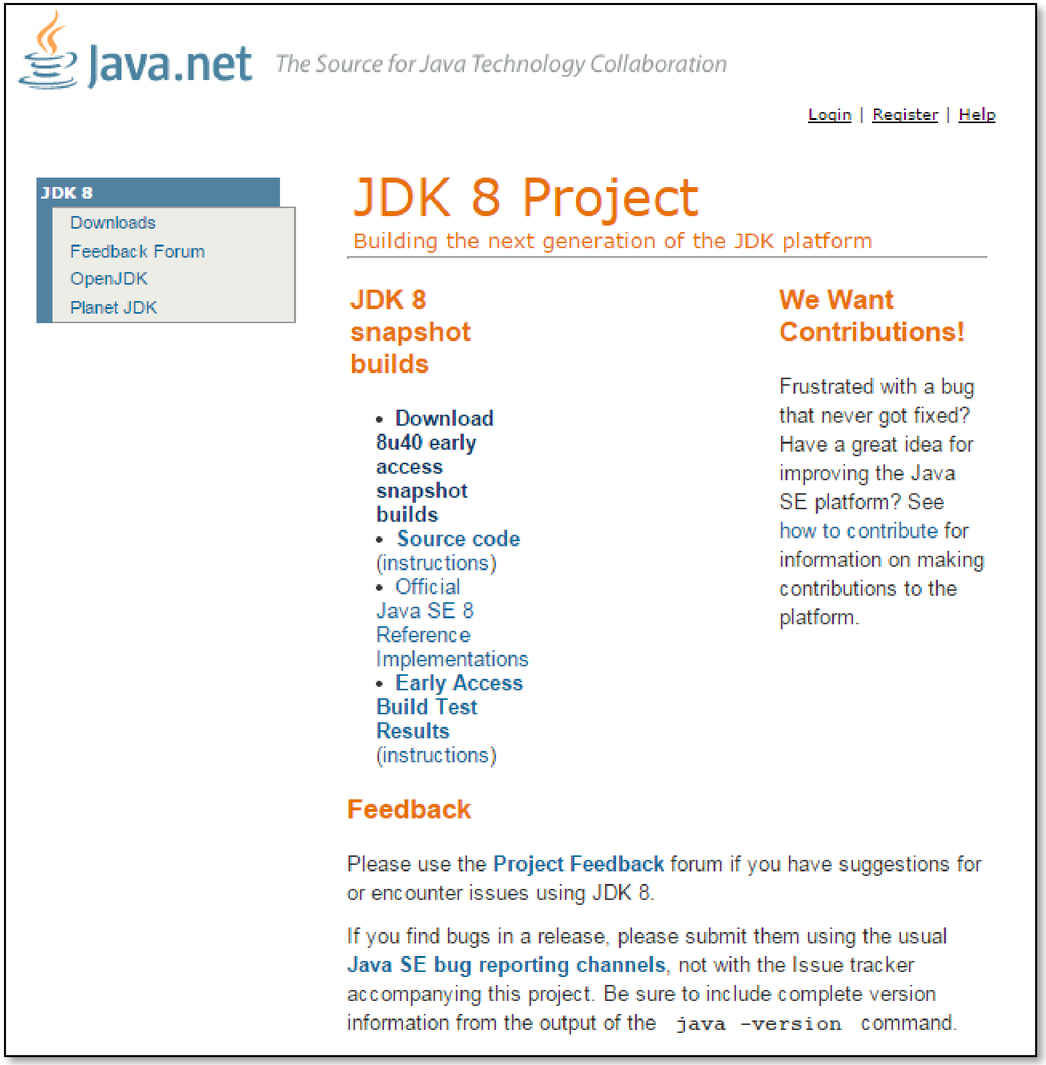
The companies that have invested in Android have done so on its merits because we believe an open platform is necessary. Android is intentionally and explicitly an open-source – as opposed to a free software – effort; a group of organizations with shared needs has pooled resources to collaborate on a 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 8 source code is available
  - For browsing zgrepcode.com



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)

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.

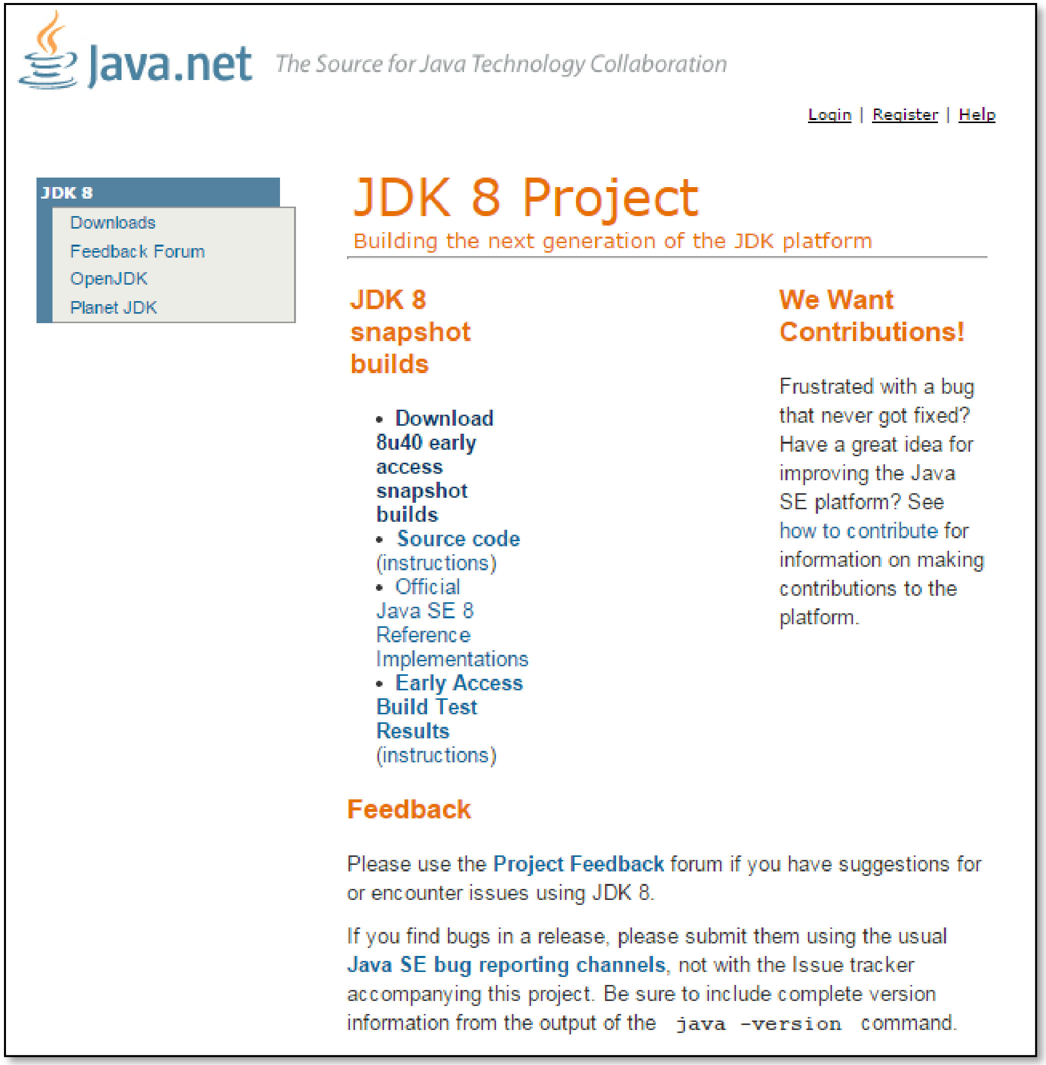
Feedback

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.

Accessing Java & Android Source Code

- Java 8 source code is available
  - For browsing zgrepcode.com
  - For downloading jdk8.java.net/download.html



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)

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.

Feedback

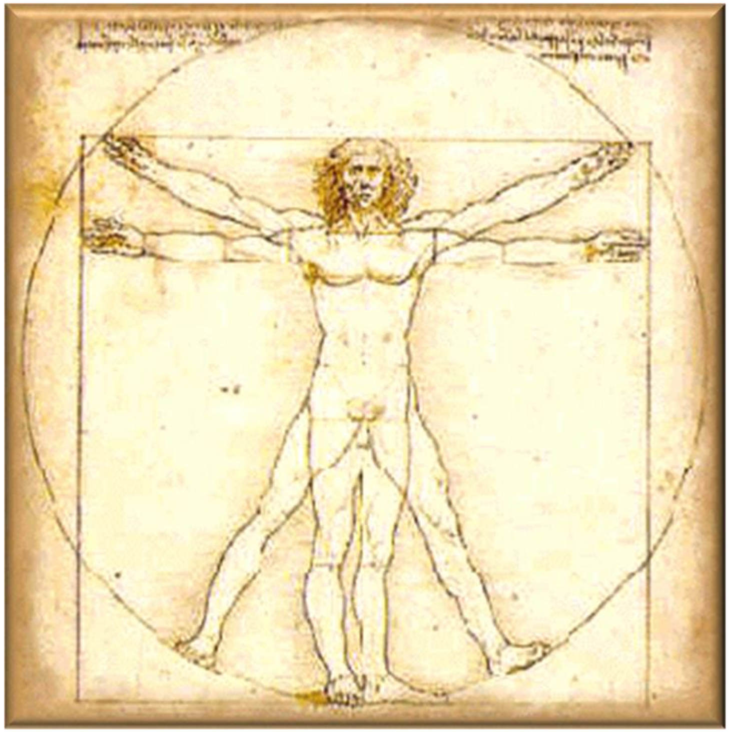
Please use the **Project Feedback** forum if you have suggestions for or encounter issues using JDK 8.

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.

Summary

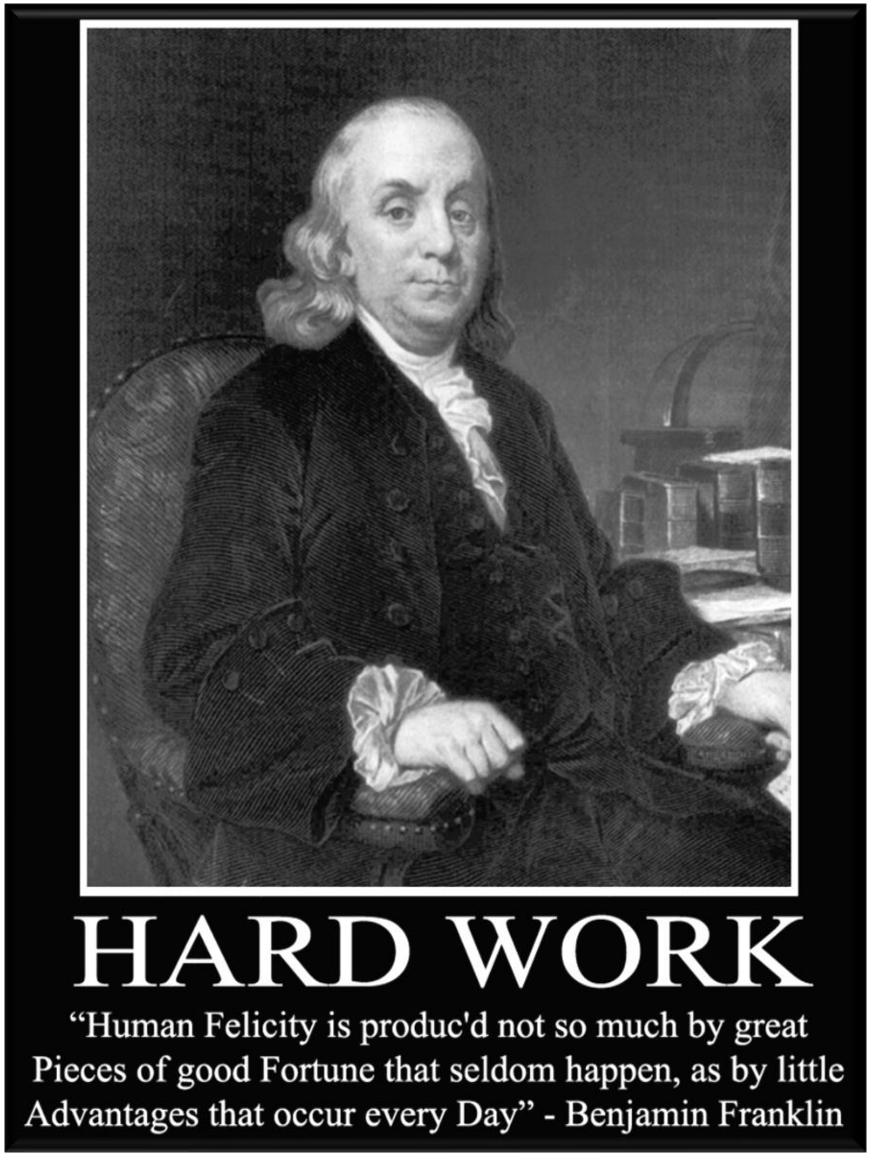
Summary

- 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…

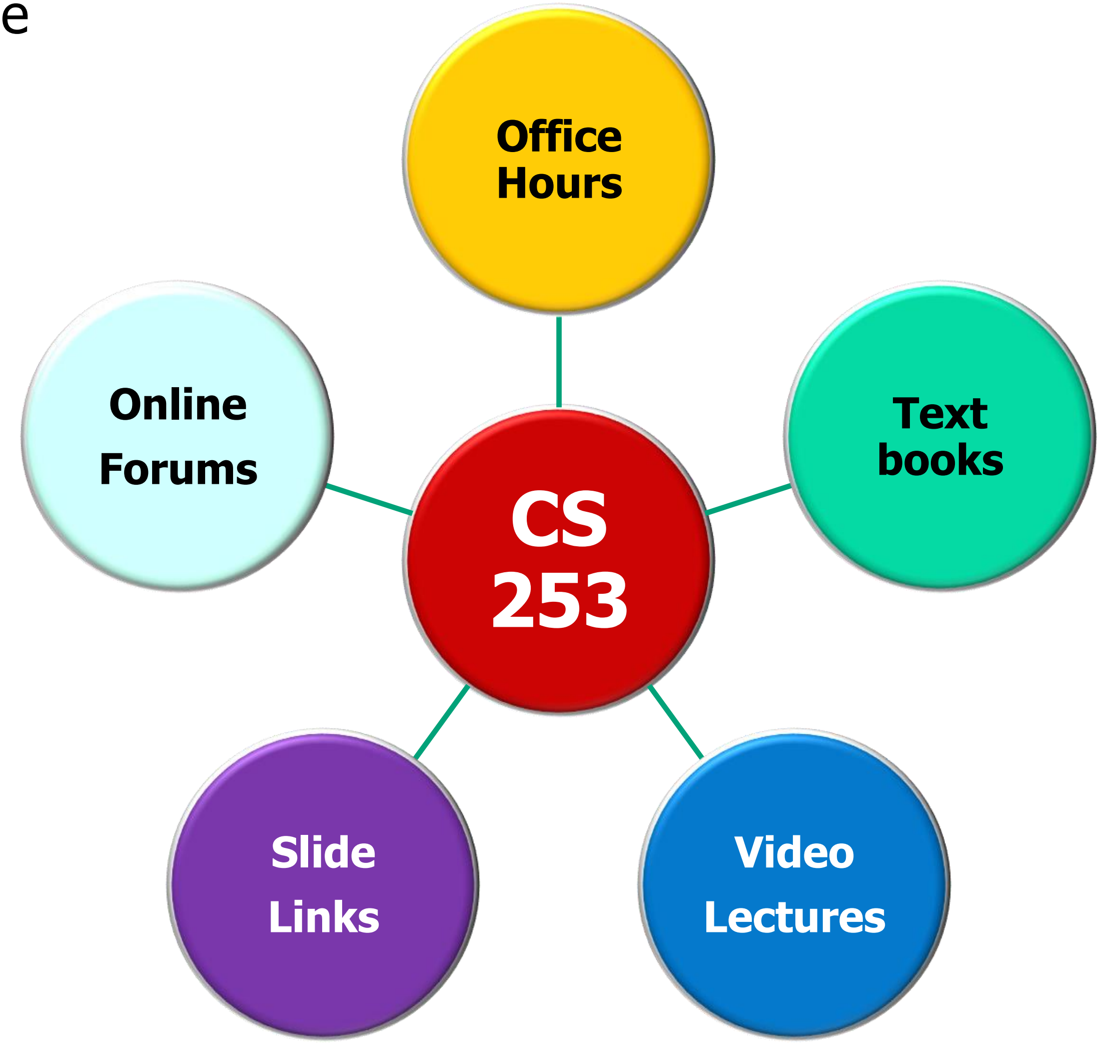
Summary

- 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/fall2020/cs253

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/cs253

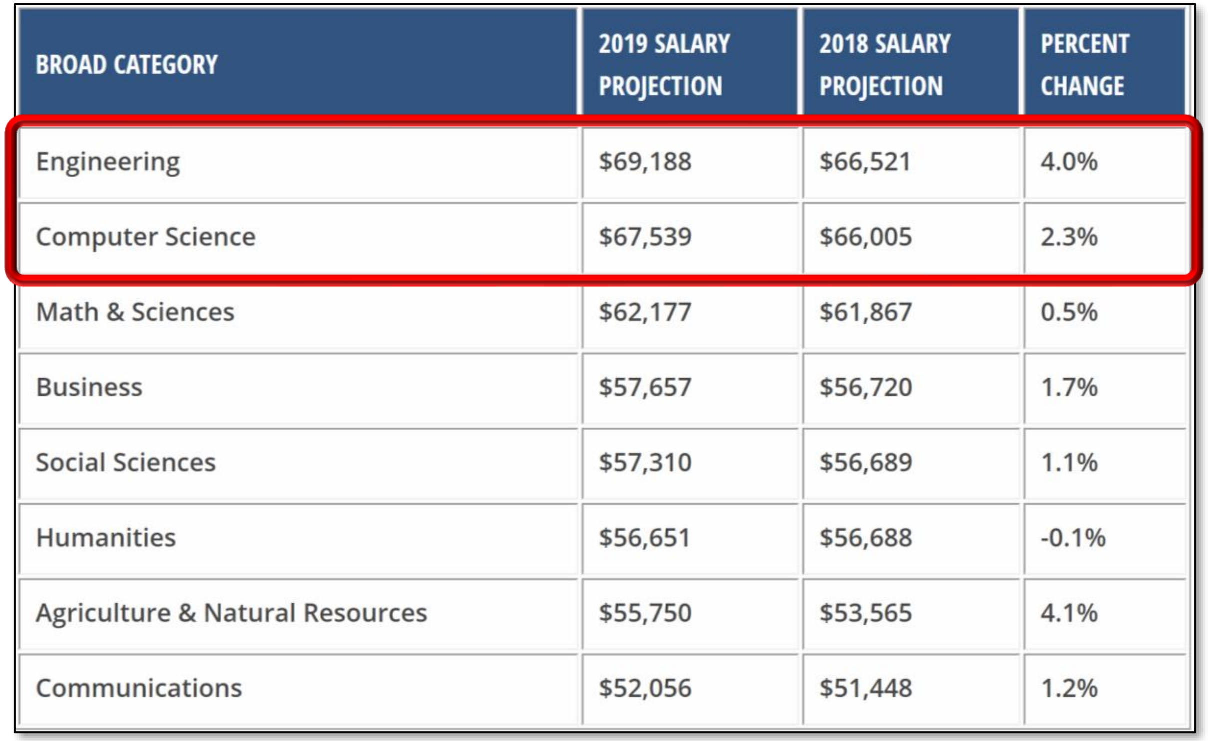
Summary

- 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	2019 SALARY PROJECTION	2018 SALARY PROJECTION	PERCENT CHANGE
Engineering	$69,188	$66,521	4.0%
Computer Science	$67,539	$66,005	2.3%
Math & Sciences	$62,177	$61,867	0.5%
Business	$57,657	$56,720	1.7%
Social Sciences	$57,310	$56,689	1.1%
Humanities	$56,651	$56,688	-0.1%
Agriculture & Natural Resources	$55,750	$53,565	4.1%
Communications	$52,056	$51,448	1.2%

See www.naceweb.org/job-market/compensation/stem-majors-projected-to-be-class-of-2019s-top-paid/