

# CS 891: Concurrent Java Programming in Android

## Overview & Logistics

Douglas C. Schmidt

[d.schmidt@vanderbilt.edu](mailto:d.schmidt@vanderbilt.edu)

[www.dre.vanderbilt.edu/~schmidt](http://www.dre.vanderbilt.edu/~schmidt)



Professor of Computer Science

Institute for Software  
Integrated Systems

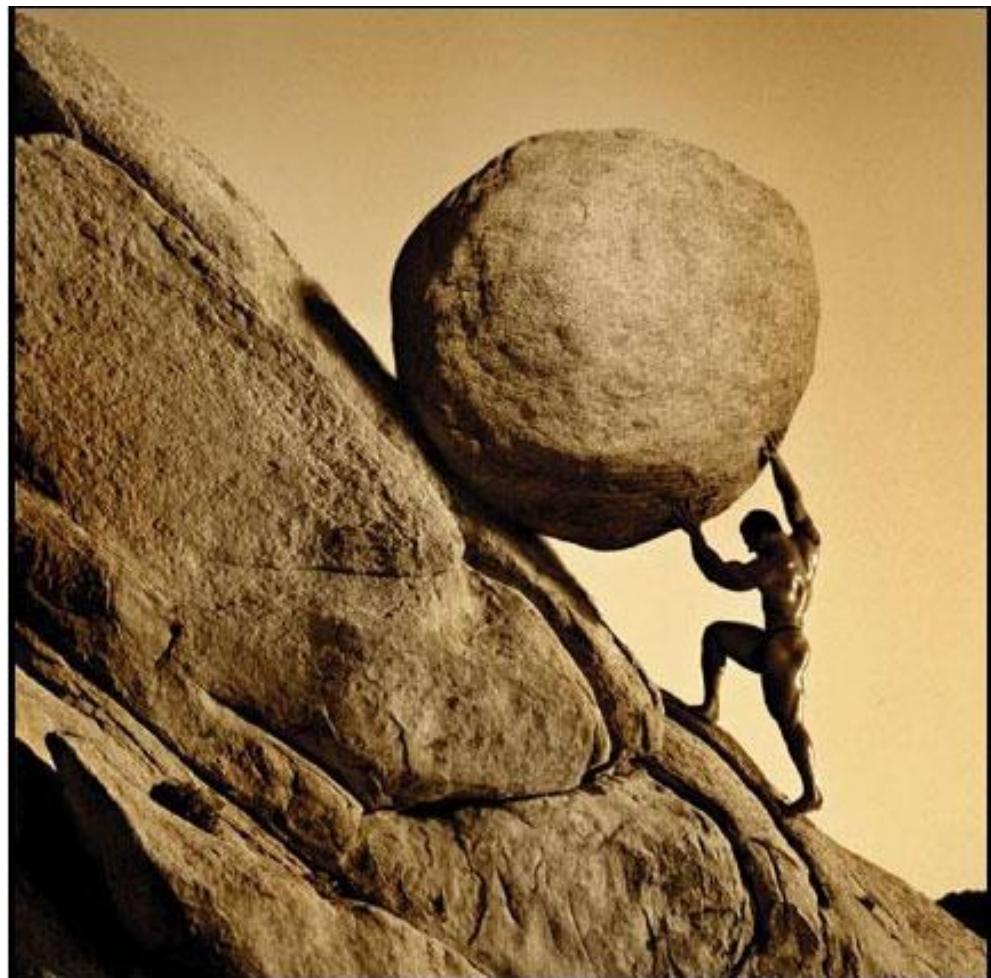
Vanderbilt University  
Nashville, Tennessee, USA



# 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

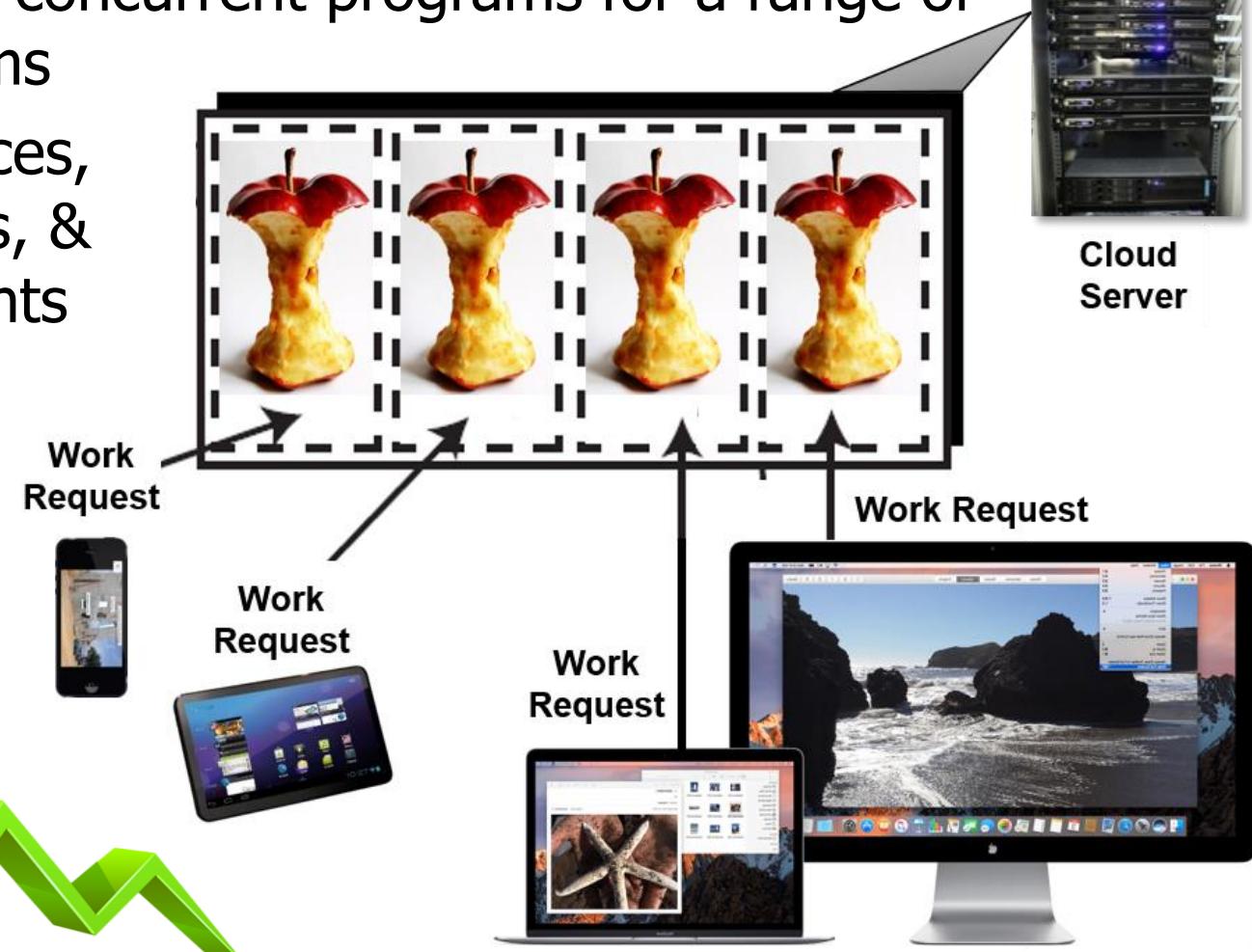


---

# Course Philosophy

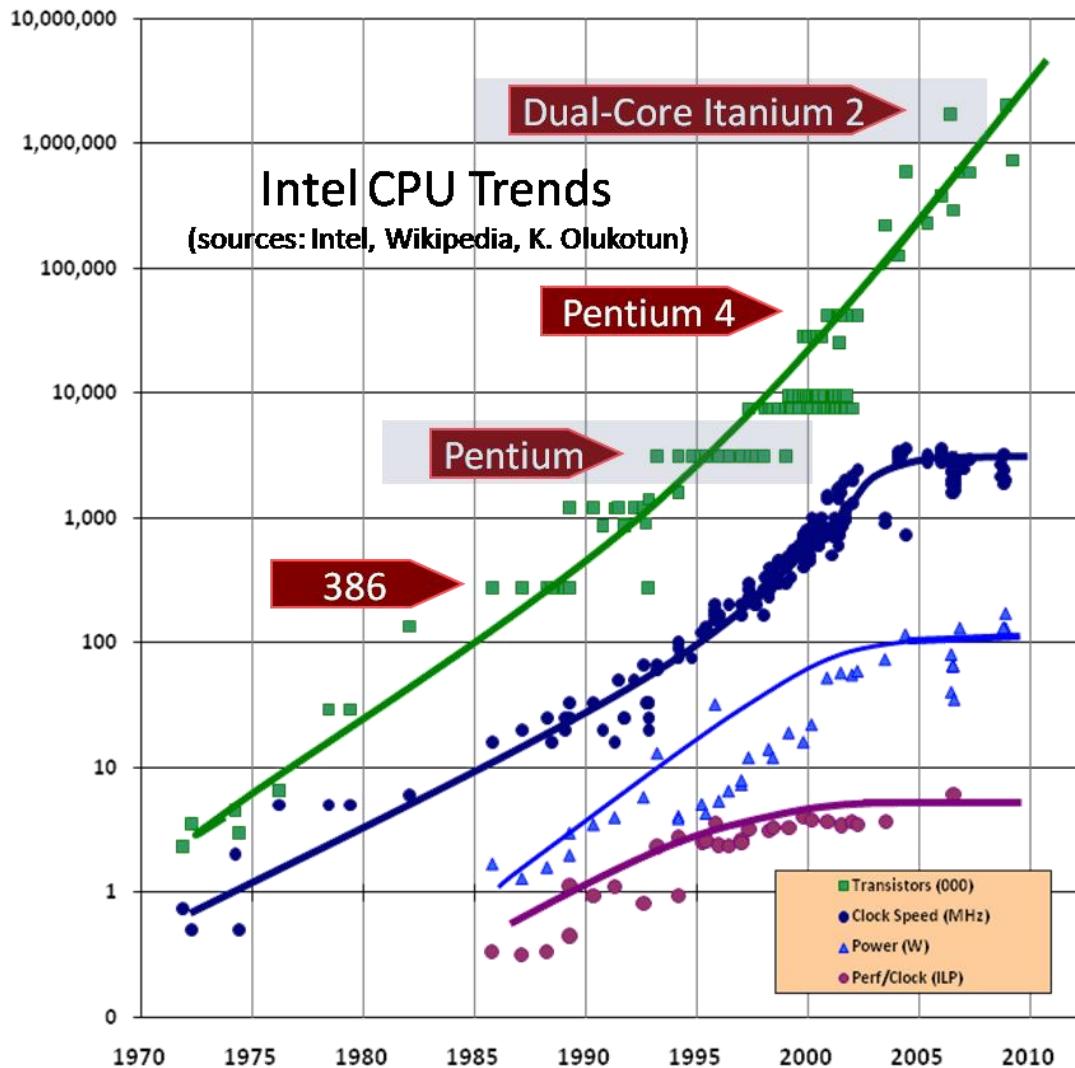
# Course Philosophy

- There's a growing need for software developers who know how to write concurrent 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
  - e.g., multi-core & many core processors, mass storage, ubiquitous network connectivity, & commodity hardware & software platforms

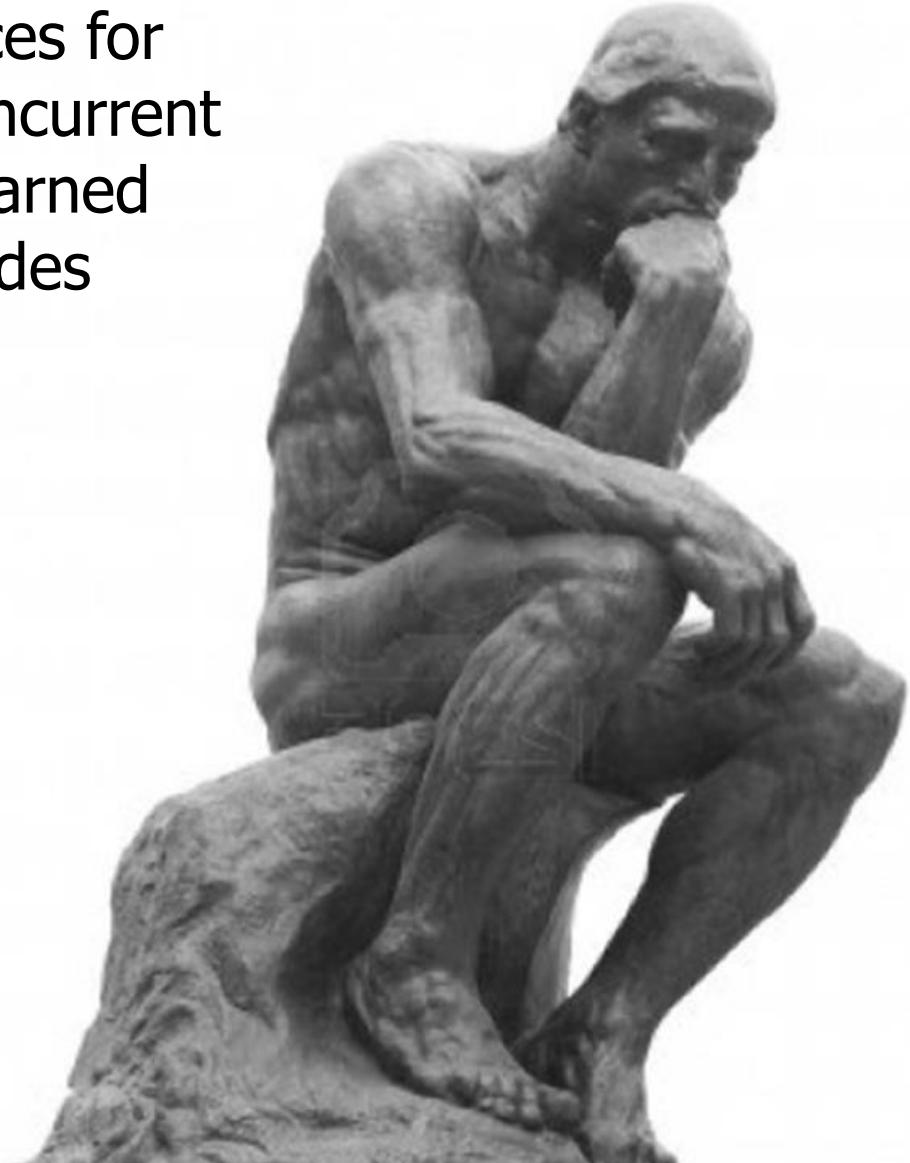


See [www.gotw.ca/publications/concurrency-ddj.htm](http://www.gotw.ca/publications/concurrency-ddj.htm)

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



---

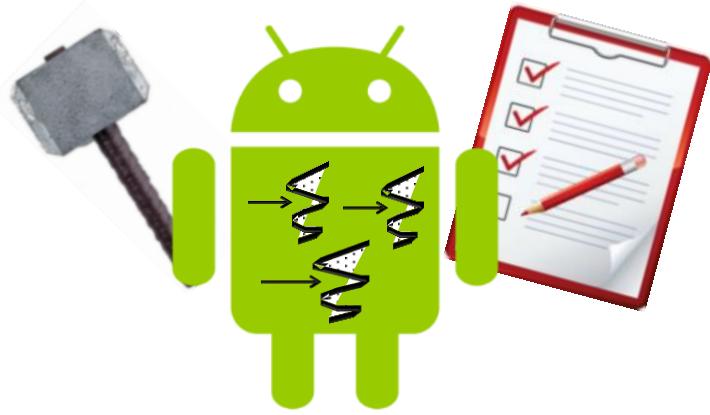
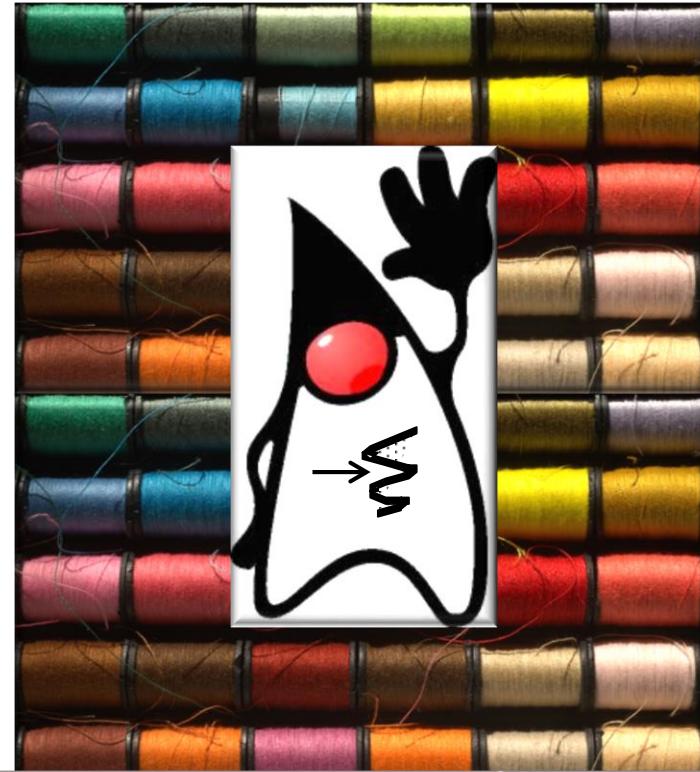
This course involves lots of hands-on software development & testing!

---

# Summary of the Course Contents

# Summary of Course Contents

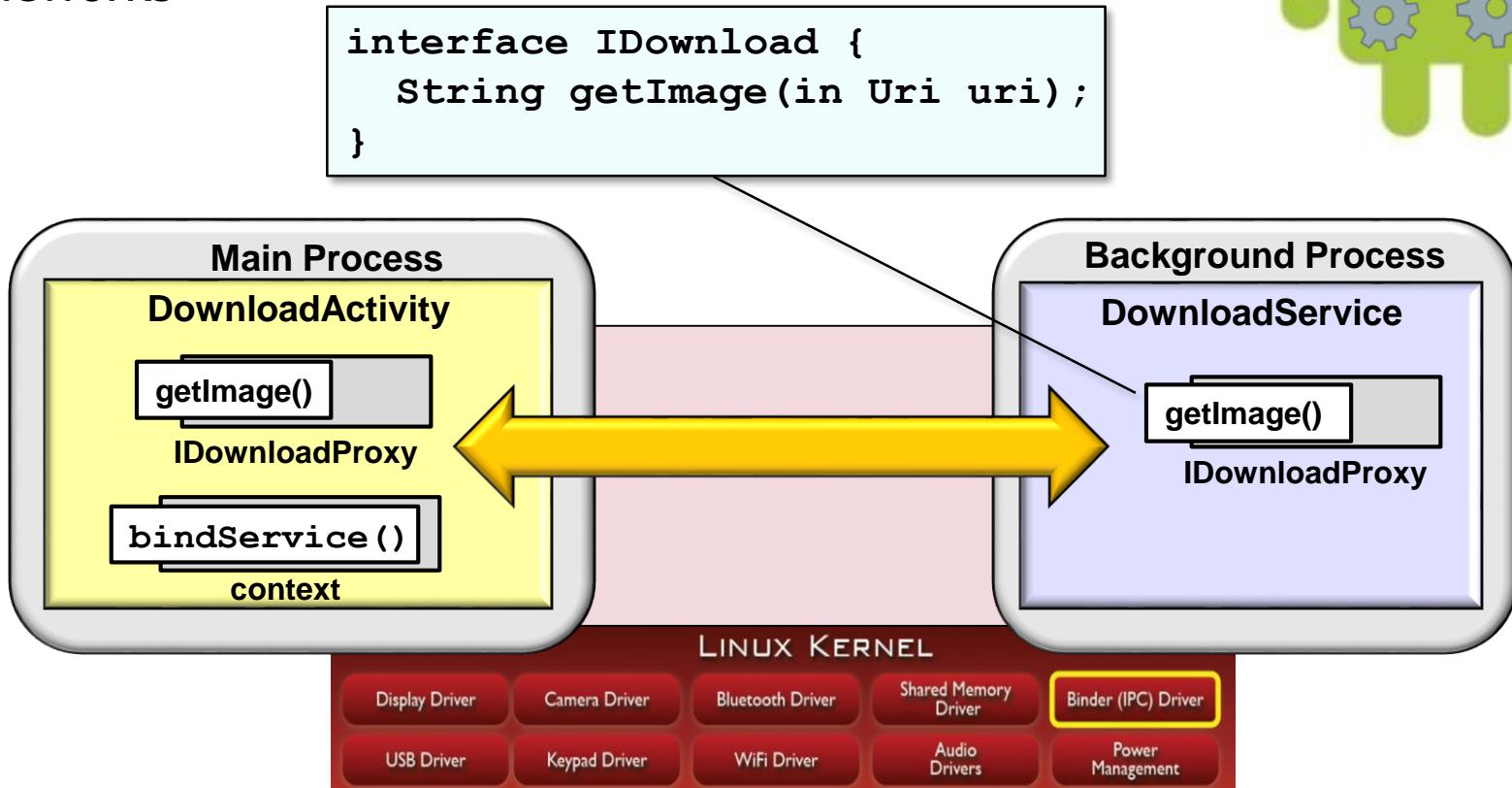
- Coverage of foundational Java & Android concurrency mechanisms



Including Java 8 programming language features & concurrency mechanisms

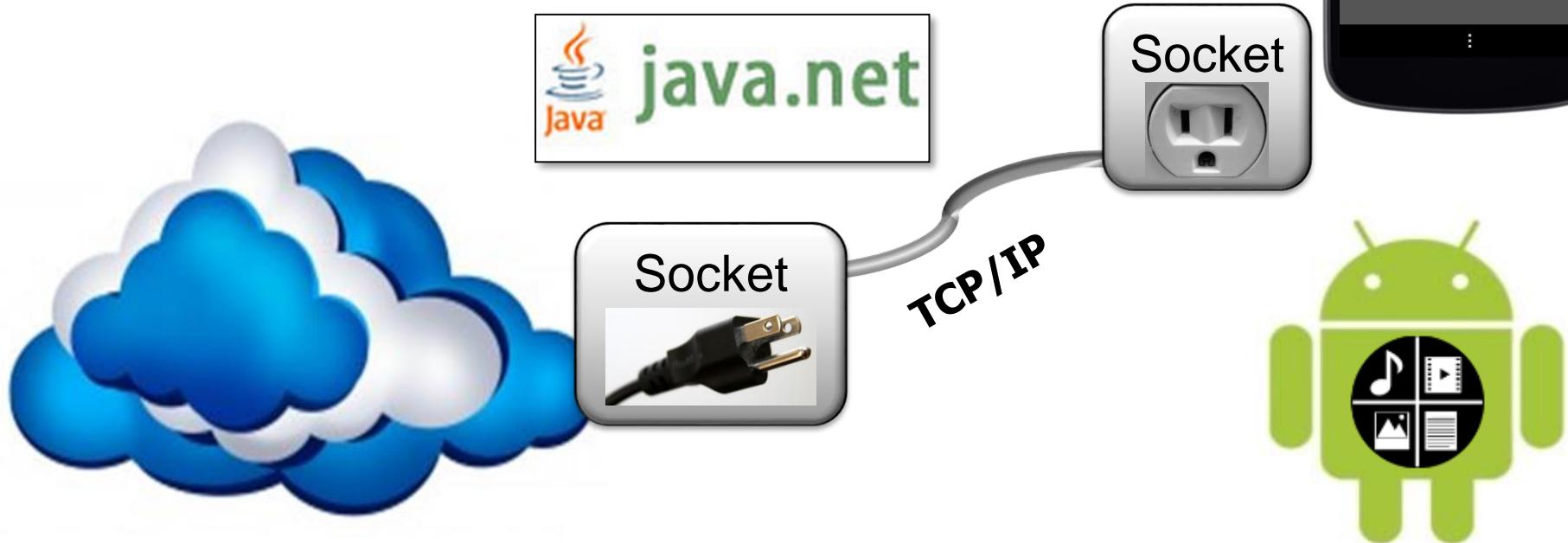
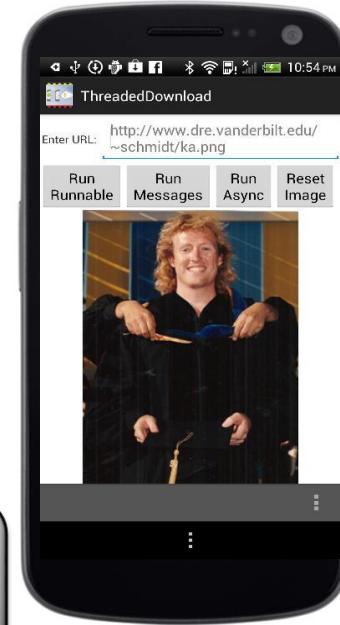
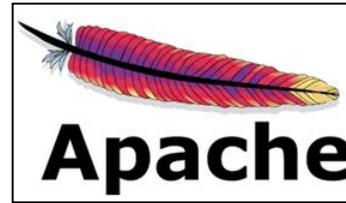
# Summary of Course Contents

- Coverage of foundational Java & Android concurrency mechanisms
- Coverage of Android bound services & inter-process communication (IPC) frameworks



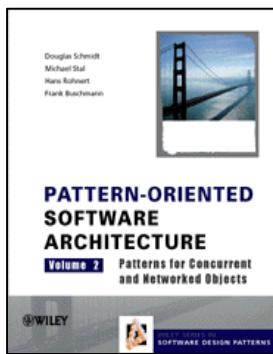
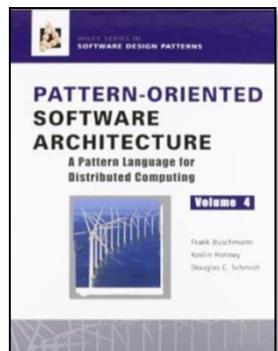
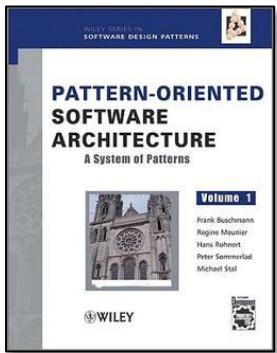
# Summary of Course Contents

- Coverage of foundational Java & Android concurrency mechanisms
- Coverage of Android bound services & inter-process communication (IPC) frameworks
- Mobile & Web communication & content providers (maybe)

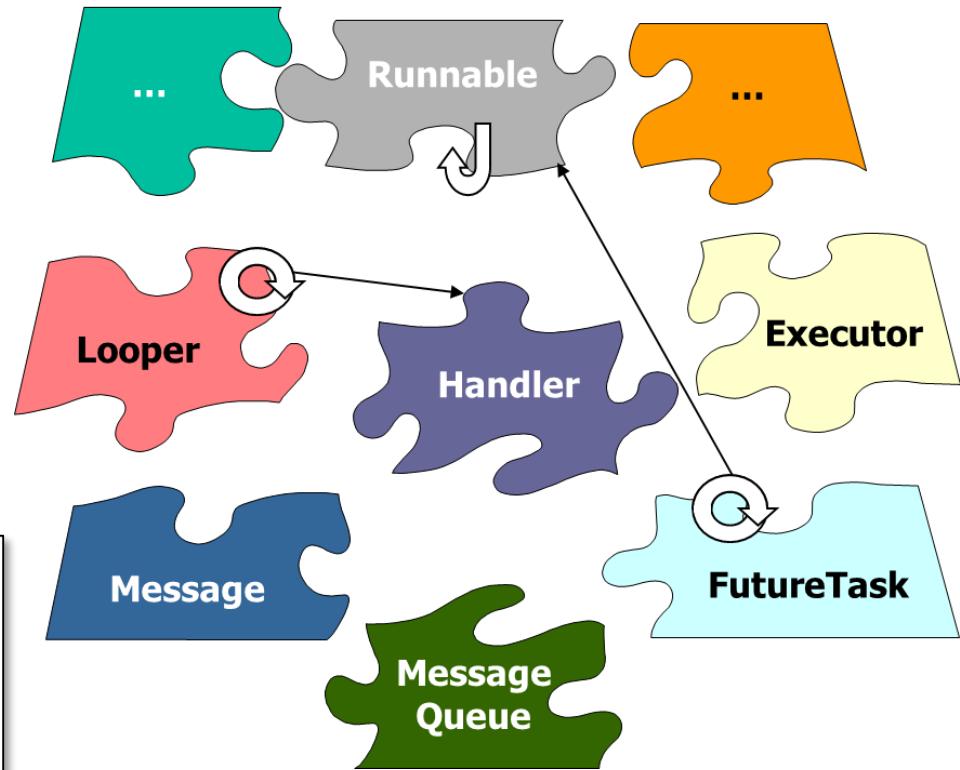


# Summary of Course Contents

- Coverage of foundational Java & Android concurrency mechanisms
- Coverage of Android bound services & inter-process communication (IPC) frameworks
- Mobile & Web communication & content providers (maybe)
- Patterns/frameworks for concurrent & networked programming (maybe)

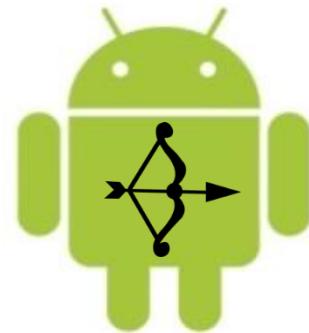


***Application-specific functionality***



# Summary of Course Contents

- Coverage of foundational Java & Android concurrency mechanisms
- Coverage of Android bound services & inter-process communication (IPC) frameworks
- Mobile & Web communication & content providers (maybe)
- Patterns/frameworks for concurrent & networked programming (maybe)
- We assume you know (or can quickly learn) Android, Java 8, & Git



See [www.coursera.org/specializations/android-app-development](http://www.coursera.org/specializations/android-app-development)

---

# Structure of the Lecture Material

# Structure of the Lecture Material

---

- This course has four main modules

Section	Topics
Java & Android Threading	<ul style="list-style-type: none"><li>• Coverage of basic &amp; advanced Java &amp; Android threading mechanisms, e.g.<ul style="list-style-type: none"><li>• Java Threads, Runnables, &amp; Executor framework</li><li>• Android HaMeR &amp; AsyncTask frameworks</li></ul></li></ul>

# Structure of the Lecture Material

---

- This course has four main modules

Section	Topics
Java Threading	<ul style="list-style-type: none"><li>• Coverage of basic &amp; advanced Java threading mechanisms, e.g.<ul style="list-style-type: none"><li>• Java Threads, Runnables, &amp; Executor framework</li><li>• Android HaMeR &amp; AsyncTask frameworks</li></ul></li></ul>
Java Synchronization	<ul style="list-style-type: none"><li>• Coverage of basic &amp; advanced Java synchronization mechanisms, e.g.,<ul style="list-style-type: none"><li>• Build-in monitor objects</li><li>• Myriad synchronizer classes in <code>java.util.concurrent</code></li></ul></li></ul>

# Structure of the Lecture Material

- This course has four main modules

Section	Topics
Java Threading	<ul style="list-style-type: none"><li>• Coverage of basic &amp; advanced Java threading mechanisms, e.g.<ul style="list-style-type: none"><li>• Java Threads, Runnables, &amp; Executor framework</li><li>• Android HaMeR &amp; AsyncTask frameworks</li></ul></li></ul>
Java Synchronization	<ul style="list-style-type: none"><li>• Coverage of basic &amp; advanced Java synchronization mechanisms, e.g.,<ul style="list-style-type: none"><li>• Build-in monitor objects</li><li>• Myriad synchronizer classes in <code>java.util.concurrent</code></li></ul></li></ul>
Mobile ⇄ Web Communication	<ul style="list-style-type: none"><li>• Android Bound Services &amp; AIDL, Android Content Providers</li></ul>

**OPTIONAL**

# Structure of the Lecture Material

- This course has four main modules

Section	Topics
Java Threading	<ul style="list-style-type: none"><li>• Coverage of basic &amp; advanced Java threading mechanisms, e.g.<ul style="list-style-type: none"><li>• Java Threads, Runnables, &amp; Executor framework</li><li>• Android HaMeR &amp; AsyncTask frameworks</li></ul></li></ul>
Java Synchronization	<ul style="list-style-type: none"><li>• Coverage of basic &amp; advanced Java synchronization mechanisms, e.g.,<ul style="list-style-type: none"><li>• Build-in monitor objects</li><li>• Myriad synchronizer classes in <code>java.util.concurrent</code></li></ul></li></ul>
Mobile ⇄ Web Communication	<ul style="list-style-type: none"><li>• Android Bound Services &amp; AIDL, Android Content Providers</li></ul>
Software Patterns	<ul style="list-style-type: none"><li>• Concurrency &amp; communication patterns</li></ul>

**OPTIONAL**

# Structure of the Lecture Material

- This course has four main modules

Section	Topics
Java Threading	<ul style="list-style-type: none"><li>• Coverage of basic &amp; advanced Java threading mechanisms, e.g.<ul style="list-style-type: none"><li>• Java Threads, Runnables, &amp; Executor framework</li><li>• Android HaMeR &amp; AsyncTask frameworks</li></ul></li></ul>
Java Synchronization	<ul style="list-style-type: none"><li>• Coverage of basic &amp; advanced Java synchronization mechanisms, e.g.,<ul style="list-style-type: none"><li>• Build-in monitor objects</li><li>• Myriad synchronizer classes in <code>java.util.concurrent</code></li></ul></li></ul>
Mobile ⇄ Web Communication	<ul style="list-style-type: none"><li>• Android Bound Services &amp; AIDL, Android Content Providers</li></ul>
Software Patterns	<ul style="list-style-type: none"><li>• Concurrency &amp; communication patterns</li></ul>



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

# Structure of the Lecture Material

---

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



# Structure of the Lecture Material

---

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



# Structure of the Lecture Material

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



Screencasts of each lesson “part” & PDF versions of the slides will be uploaded to [www.dre.vanderbilt.edu/~schmidt/cs891s#lectures](http://www.dre.vanderbilt.edu/~schmidt/cs891s#lectures)

# Structure of the Lecture Material

---

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



---

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

# Structure of the Lecture Material

---

- There will be monthly quizzes on material covered in the lectures



---

Typically held on the first Wednesday of each month

# Structure of the Lecture Material

- There will be monthly quizzes on material covered in the lectures
  - All quizzes (& the final) are “closed book”

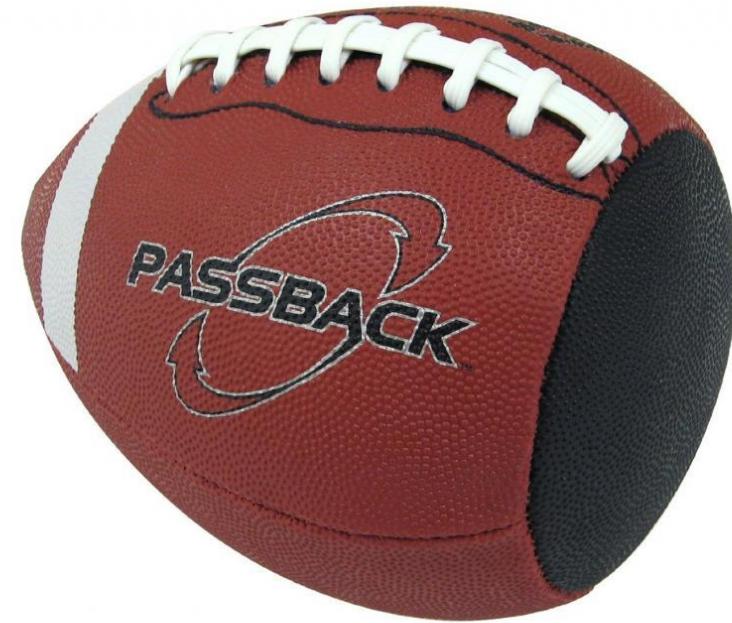


I recommend that you study for quizzes by reviewing slides & watching screencasts available at [www.dre.vanderbilt.edu/~schmidt/cs891s#lectures](http://www.dre.vanderbilt.edu/~schmidt/cs891s#lectures)

# Structure of the Lecture Material

---

- There will be weekly quizzes on material covered in the lectures
  - All quizzes (& the final) are “closed book”
  - We’ll pass back & review quizzes at the start of the next class



---

One of the benefits of a smaller class ;-)

# Structure of the Lecture Material

---

- There will be weekly quizzes on material covered in the lectures
  - All quizzes (& the final) are “closed book”
  - We’ll pass back & review quizzes at the start of 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/cs891s/work-summary.html#quizzes](http://www.dre.vanderbilt.edu/~schmidt/cs891s/work-summary.html#quizzes)

# Structure of the Lecture Material

---

- There will be weekly quizzes on material covered in the lectures
  - All quizzes (& the final) are “closed book”
  - We’ll pass back & review quizzes at the start of 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 quiz & don’t attend class you’ll be penalized 50%



---

See [www.dre.vanderbilt.edu/~schmidt/cs891s/work-summary.html#quizzes](http://www.dre.vanderbilt.edu/~schmidt/cs891s/work-summary.html#quizzes)

# Structure of the Lecture Material

---

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



---

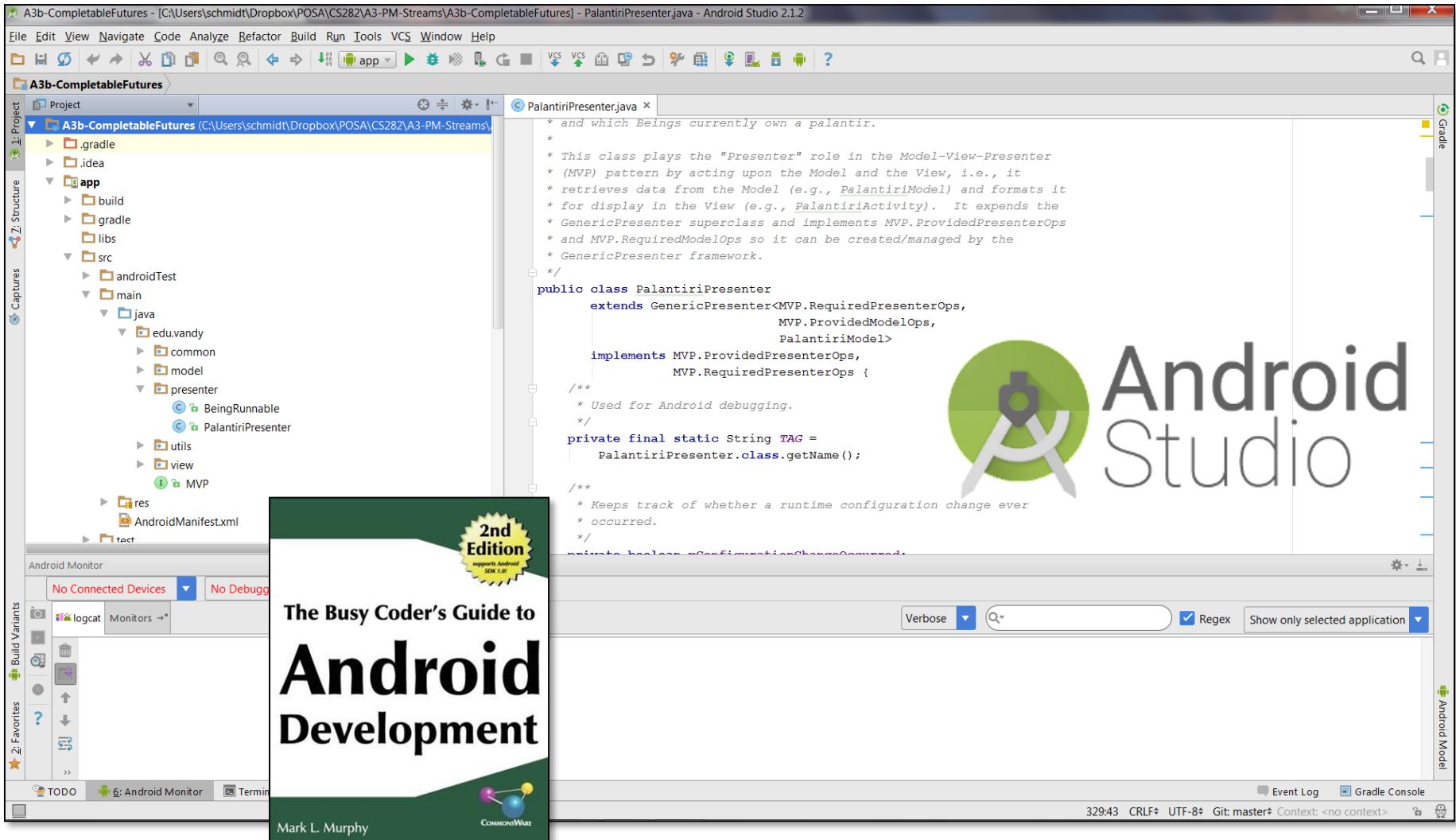
The final exam is 2pm to 5pm, Thursday, April 25<sup>th</sup> in this room

---

# Overview of the Assignments & Assessments

# Overview of Assignments & Assessments

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



You can use any IDE, but your final submission *must* build/run with Android Studio 3.2.x & Android Pie 9 (API 28)

# Overview of Assignments & Assessments

---

- Programming assignments should be written in Java 8 using Android Studio
  - You'll need to install the Java 8 runtime environment (JRE)



---

See [github.com/douglasraigschmidt/CS891/wiki/Installing-Software](https://github.com/douglasraigschmidt/CS891/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](http://developer.android.com/sdk)

# Overview of Assignments & Assessments

---

- Android programming assignments must be submitted using Android Studio
  - Please install Android 9.x Pie (API level 28)



See [en.wikipedia.org/wiki/Android\\_Pie](https://en.wikipedia.org/wiki/Android_Pie)

# Overview of Assignments & Assessments

- All source code for assignments & examples available at GitHub

The screenshot shows a GitHub repository page for 'douglasraigschmidt / CS891'. At the top, there are navigation links for 'Code', 'Issues 0', 'Pull requests 0', 'Projects 0', 'Wiki', 'Insights', and 'Settings'. To the right are buttons for 'Unwatch', 'Star', 'Fork', and 'Edit'. Below these are statistics: 40 commits, 1 branch, 0 releases, and 1 contributor. A dropdown menu shows 'Branch: master' and a 'New pull request' button. On the right, there's a green 'Clone or download' button. The main area lists repository contents: 'assignments' (updated a minute ago), 'ex' (updated 4 months ago), and 'README.md' (created just now). The 'README.md' file was created by 'douglasraigschmidt'.

Contains examples and assignments for my CS 891 course at Vanderbilt University, which can be accessed via <http://www.dre.vanderbilt.edu/~schmidt/cs891>.

Add topics

40 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

douglasraigschmidt Create README.md Latest commit a940fc3 just now

assignments updates a minute ago

ex updates 4 months ago

README.md Create README.md just now

Go to GitHub at [github.com/douglasraigschmidt/CS891](https://github.com/douglasraigschmidt/CS891)

# Overview of Assignments & Assessments

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



 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.

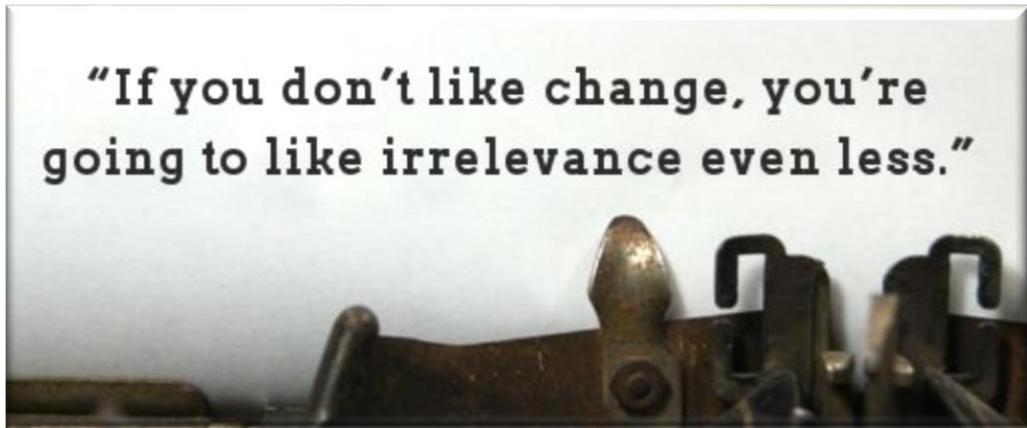
Do you want more from your GitLab installation? A subscription bundles the Enterprise Edition with support from the GitLab team. The Enterprise Edition allows you to sync LDAP groups, control pushes via git hooks, integrate better with Jenkins and Jira, and to run MySQL and forward logs when using our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

[GitLab Community Edition](#) [Get a subscription](#)

# 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 multiple times



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



# Overview of Assignments & Assessments

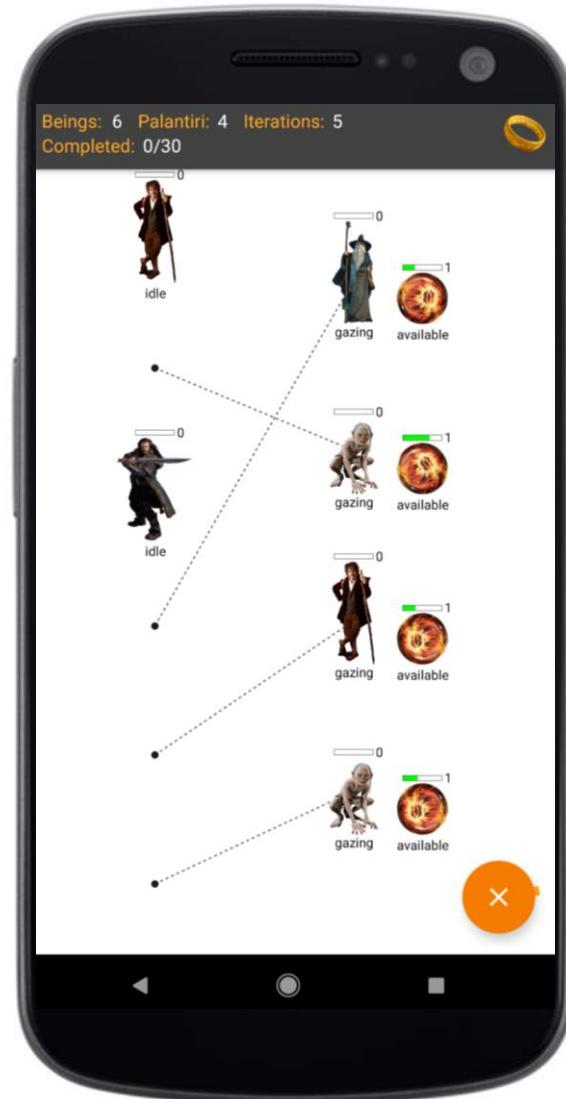
- Assignments will provide a range of experience with Java 8 & Android concurrent/communication programs



See [github.com/douglasraigschmidt/CS891/tree/master/assignments](https://github.com/douglasraigschmidt/CS891/tree/master/assignments)

# Overview of Assignments & Assessments

- In particular, you'll implement multiple variants of a Java concurrent resource manager & an associated Android app, e.g.
  - Java Threads, Runnables, & Executor frameworks
  - Android HaMeR & AsyncTask frameworks
  - Java Semaphores, built-in monitor objects, & ConcurrentHashMaps
  - Android Bound Services & AIDL



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

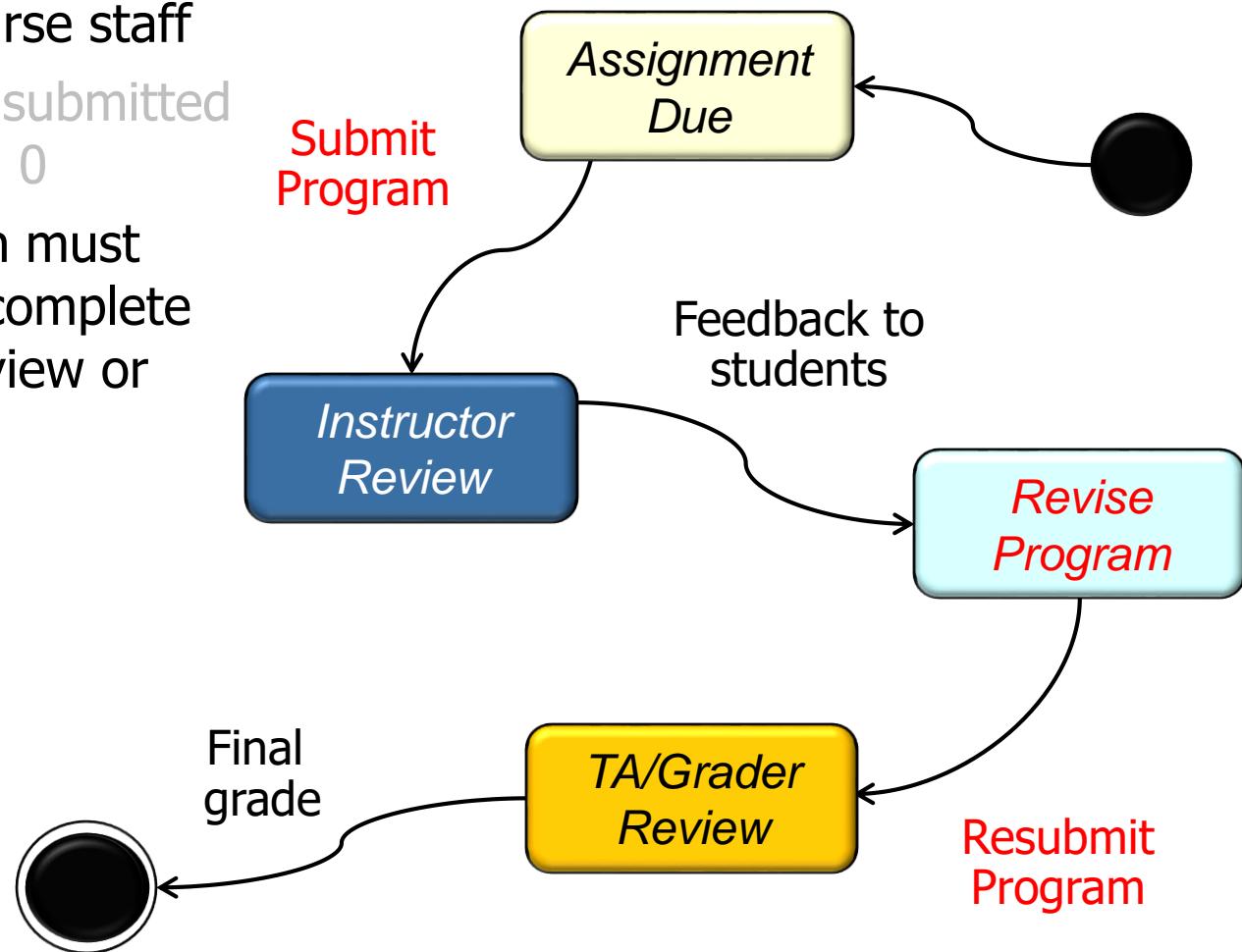


---

See [github.com/douglasraigschmidt/CS891/wiki/CS-891-FAQ](https://github.com/douglasraigschmidt/CS891/wiki/CS-891-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

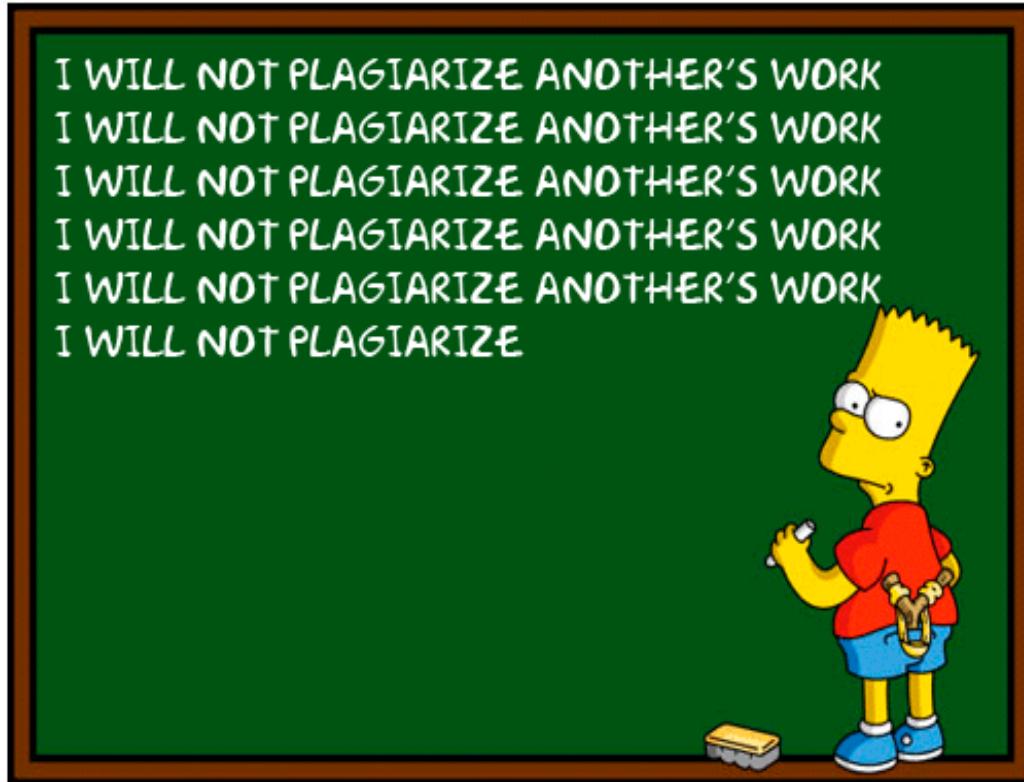


---

See [www.dre.vanderbilt.edu/~schmidt/cs891s/assignments.html](http://www.dre.vanderbilt.edu/~schmidt/cs891s/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

- Assessment criteria

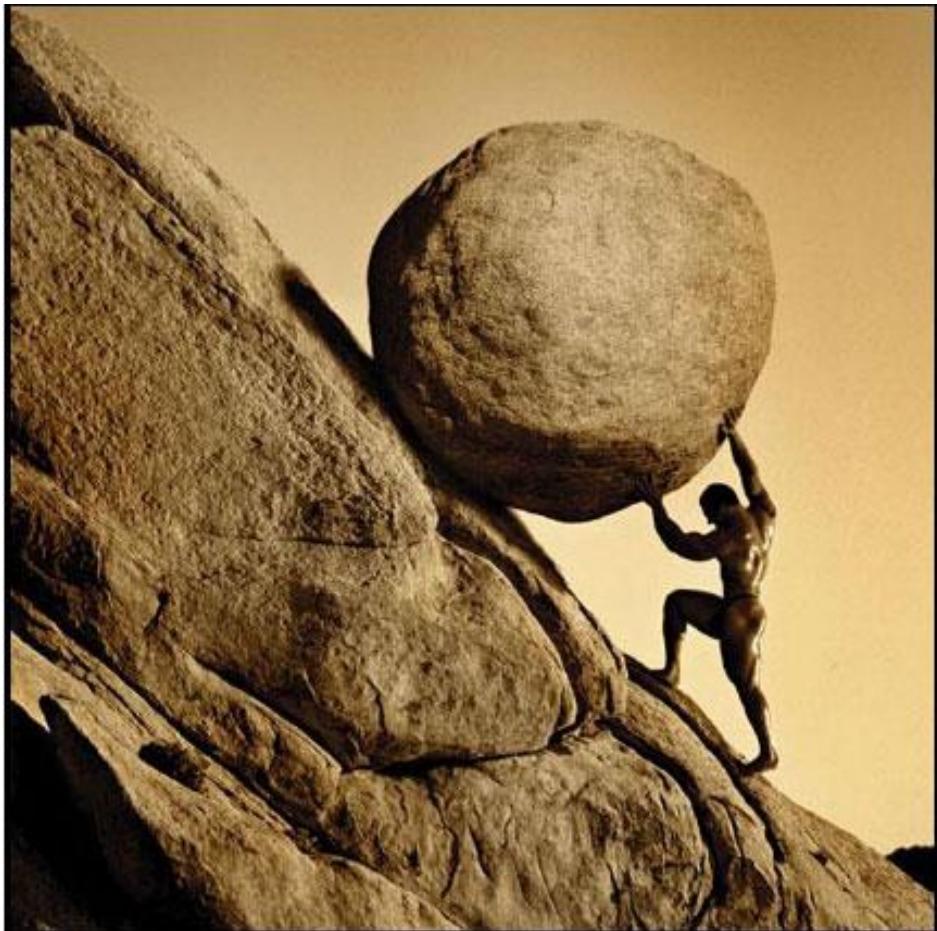
Assessment Category	%
Execution correctness	40%
Structure (e.g., modularization, information hiding, etc.)	30%
Insightful programming (e.g., developing reusable class components, etc.)	10%
Consistent style (e.g., capitalization, indenting, etc.)	10%
Appropriate commenting style	10%

See [www.dre.vanderbilt.edu/~schmidt/cs891s/assignments.html](http://www.dre.vanderbilt.edu/~schmidt/cs891s/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



# Overview of Assignments & Assessments

---

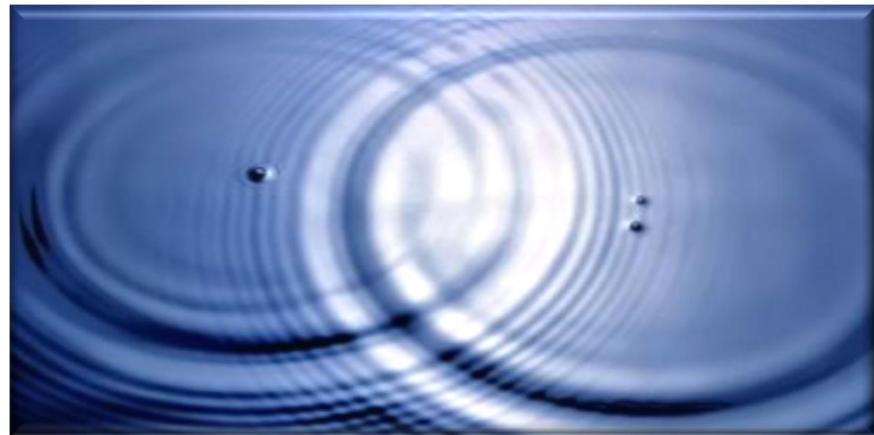
- The relative weighting of each portion of the course is:
  - 40% Quizzes
  - 40% Programming projects
  - 10% Final exam
  - 10% Participation
    - Participation is roughly 5% attendance & 5% in-class involvement in discussions



# 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 roughly 5% attendance & 5% in-class involvement in discussions

*Attendance also affects other aspects of your quiz & assignment grades*



See [www.dre.vanderbilt.edu/~schmidt/cs891s/work-summary.html#quizzes](http://www.dre.vanderbilt.edu/~schmidt/cs891s/work-summary.html#quizzes)  
& [www.dre.vanderbilt.edu/~schmidt/cs891s/assignments.html](http://www.dre.vanderbilt.edu/~schmidt/cs891s/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 roughly 5% attendance & 5% in-class involvement in discussions



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

- See [developer.android.com/sdk](http://developer.android.com/sdk)

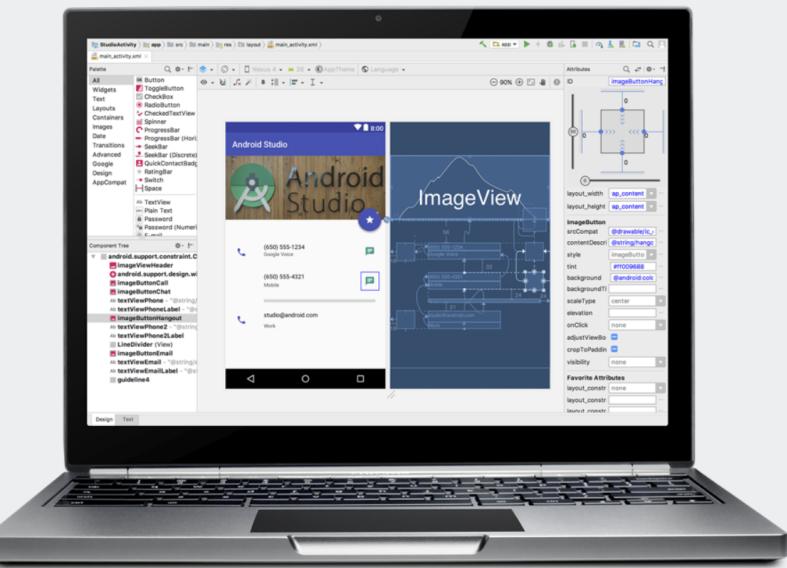
## Android Studio

The Official IDE for Android

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

World-class code editing, debugging, performance tooling, a flexible build system, and an instant build/deploy system all allow you to focus on building unique and high quality apps.

**DOWNLOAD ANDROID STUDIO**  
3.0.1 FOR WINDOWS (683 MB)



› Read the docs      › See the release notes

› Features    › Latest    › Resources    › Videos    › Download Options



# Installing Eclipse Java/Android Developer Tools

---

- Installation steps



# Installing Eclipse Java/Android Developer Tools

- Installation steps
  - Download & install the Java Standard Edition JDK & JRE 8

## Java SE Development Kit 8 Downloads

Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications, applets, and components using the Java programming language.

The JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform.

### See also:

- Java Developer Newsletter (tick the checkbox under Subscription Center > Oracle Technology News)
- Java Developer Day hands-on workshops (free) and other events
- Java Magazine

### JDK MD5 Checksum

### Looking for JDK 8 on ARM?

JDK 8 for ARM downloads have moved to the [JDK 8 for ARM download page](#).

## Java SE Development Kit 8u25

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

Accept License Agreement  Decline License Agreement

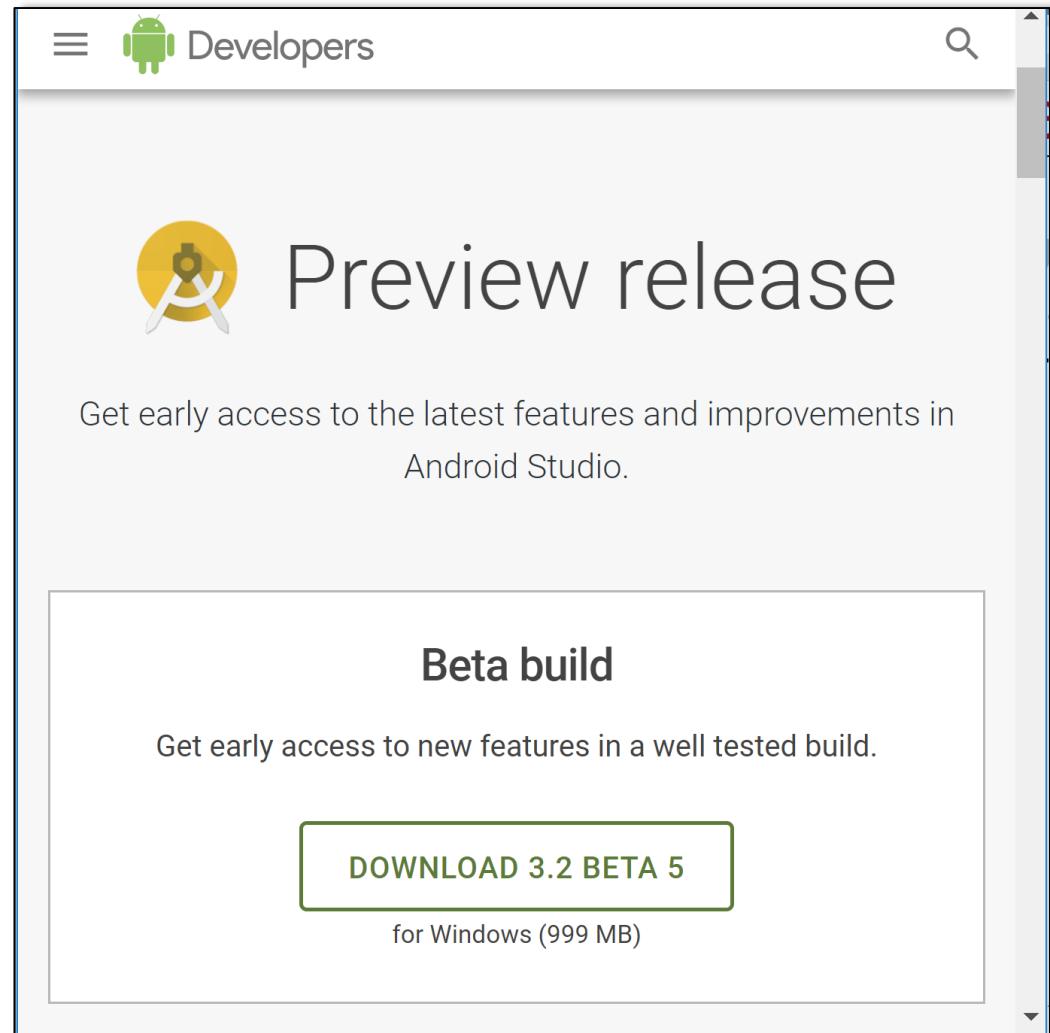
Product / File Description	File Size	Download
Linux x86	135.24 MB	 <a href="#">jdk-8u25-linux-i586.rpm</a>
Linux x86	154.88 MB	 <a href="#">jdk-8u25-linux-i586.tar.gz</a>
Linux x64	135.6 MB	 <a href="#">jdk-8u25-linux-x64.rpm</a>

[www.oracle.com/technetwork/java/javase/downloads](http://www.oracle.com/technetwork/java/javase/downloads)

# Installing Eclipse Java/Android Developer Tools

- Installation steps

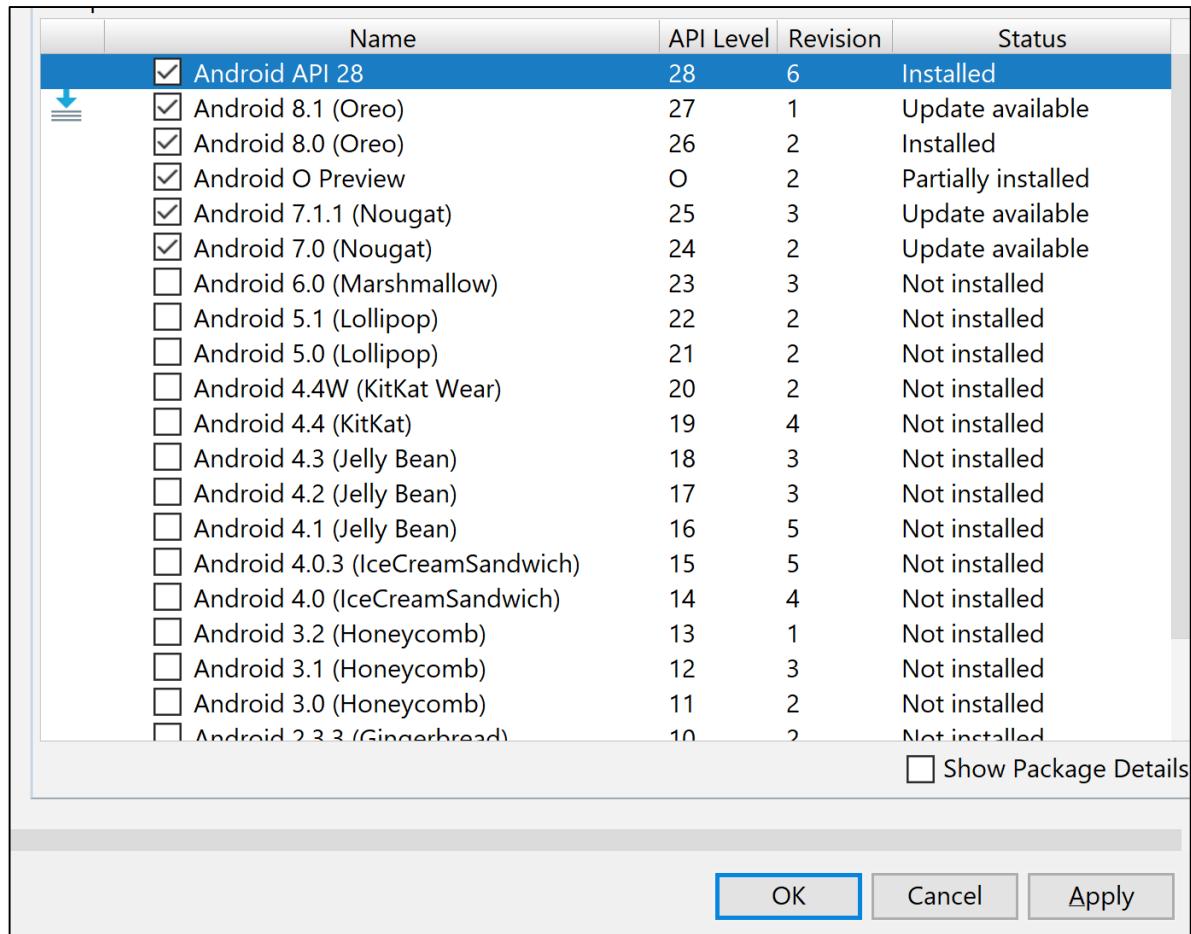
- Download & install the Java Standard Edition JDK & JRE 8
- Download & install Android Studio 3.2.x



[developer.android.com/studio/preview](http://developer.android.com/studio/preview)

# Add Components to the SDK

- Launch the Android Studio SDK Manager
  - Select “Pie” version of Android (9, API 28)



See [developer.android.com/studio/intro/update.html](https://developer.android.com/studio/intro/update.html)

# Add Components to the SDK

- Launch the Android Studio Virtual Device Manager
  - Create an Android API 28 emulator

The screenshot shows the "Android Virtual Device Manager" window. At the top, there's a title bar with the Android Studio logo and the window title. Below the title bar, the header "Your Virtual Devices" is displayed, along with the "Android Studio" logo. The main area is a table listing seven virtual devices. The columns are: Type, Name, Play Store, Resolution, API, Target, CPU/ABI, Size on Disk, and Actions. The "Actions" column contains icons for each device: a green triangle (Run), a pencil (Edit), and a dropdown arrow (More). The first device listed is a Nexus 5X API 28 emulator. The last two devices listed are "VisualStudio..." entries, which have a yellow warning icon and a "Download" link next to them. At the bottom left of the table is a blue button labeled "+ Create Virtual Device...". At the bottom right are two small buttons: one with a circular arrow and another with a question mark.

Type	Name	Play Store	Resolution	API	Target	CPU/ABI	Size on Disk	Actions
Emulator	Nexus 5X A...	►	1080 × 1920...	28	Android null...	x86	101 MB	► edit ▾
Emulator	Nexus 6P A...		1440 × 2560...	24	Android 7.0 ...	x86	1.5 GB	► edit ▾
Emulator	Nexus 6P A...		1440 × 2560...	25	Android 7.1....	x86	1.4 GB	► edit ▾
Emulator	Nexus 6P A...		1440 × 2560...	26	Android 8.0 ...	x86	3.6 GB	► edit ▾
Emulator	Nexus 6 API...		1440 × 2560...	27	Android 8.1 ...	x86	3.0 GB	► edit ▾
Emulator	VisualStudio...		Unknown R...	23	Android 6.0 ...	arm	2.5 GB	⚠ Download ▾
Emulator	VisualStudio...		Unknown R...	23	Android 6.0 ...	arm	2.5 GB	⚠ Download ▾

[developer.android.com/tools/devices/managing-avds.html](https://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/tools/devices/emulator.html#acceleration](http://developer.android.com/tools/devices/emulator.html#acceleration)

---

# Accessing Java & Android Source Code

# Accessing Java & Android Source Code

- Android source code is available
  - For browsing  
[android.googlesource.com](https://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](http://android.googlesource.com)
  - For downloading  
[source.android.com](http://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  
[grepcode.com/file/repository.grepcode.com/java/root/jdk/openjdk/8-b132/java](http://grepcode.com/file/repository.grepcode.com/java/root/jdk/openjdk/8-b132/java)



The screenshot shows the Java.net website with the following content:

**Java.net** The Source for Java Technology Collaboration

[Login](#) | [Register](#) | [Help](#)

**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  
[grepcode.com/file/repository.grepcode.com/java/root/jdk/openjdk/8-b132/java](http://grepcode.com/file/repository.grepcode.com/java/root/jdk/openjdk/8-b132/java)
  - For downloading  
[jdk8.java.net/download.html](http://jdk8.java.net/download.html)



The screenshot shows the Java.net website with the following content:

- Java.net** logo and tagline "The Source for Java Technology Collaboration".
- Login | Register | Help** links.
- JDK 8** sidebar menu with options: Downloads, Feedback Forum, OpenJDK, and Planet JDK.
- JDK 8 Project** title and subtitle "Building the next generation of the JDK platform".
- JDK 8 snapshot builds** section listing:
  - Download 8u40 early access snapshot builds
  - Source code (instructions)
  - Official Java SE 8 Reference Implementations
  - Early Access Build Test Results (instructions)
- We Want Contributions!** callout: "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** section: "Please use the [Project Feedback](#) forum if you have suggestions for or encounter issues using JDK 8." It also states: "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."

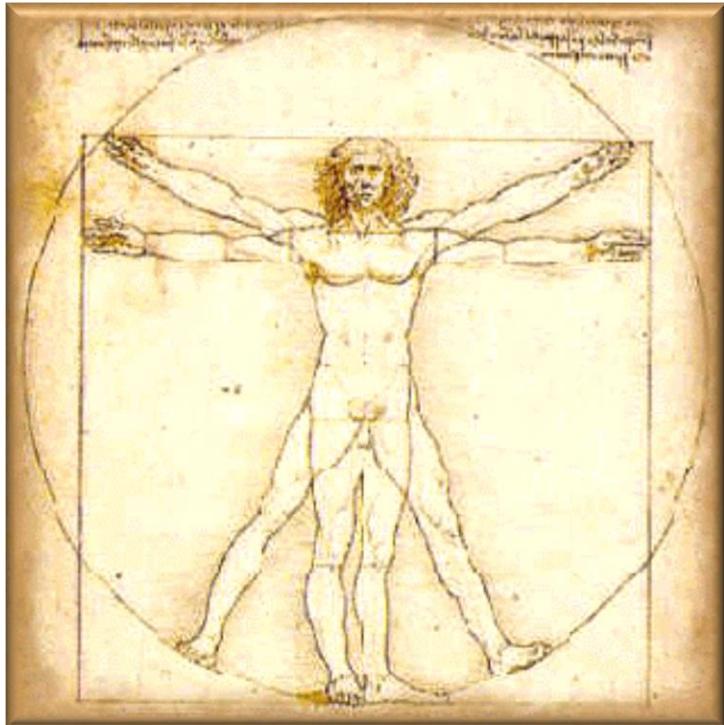
# Summary



# Summary

---

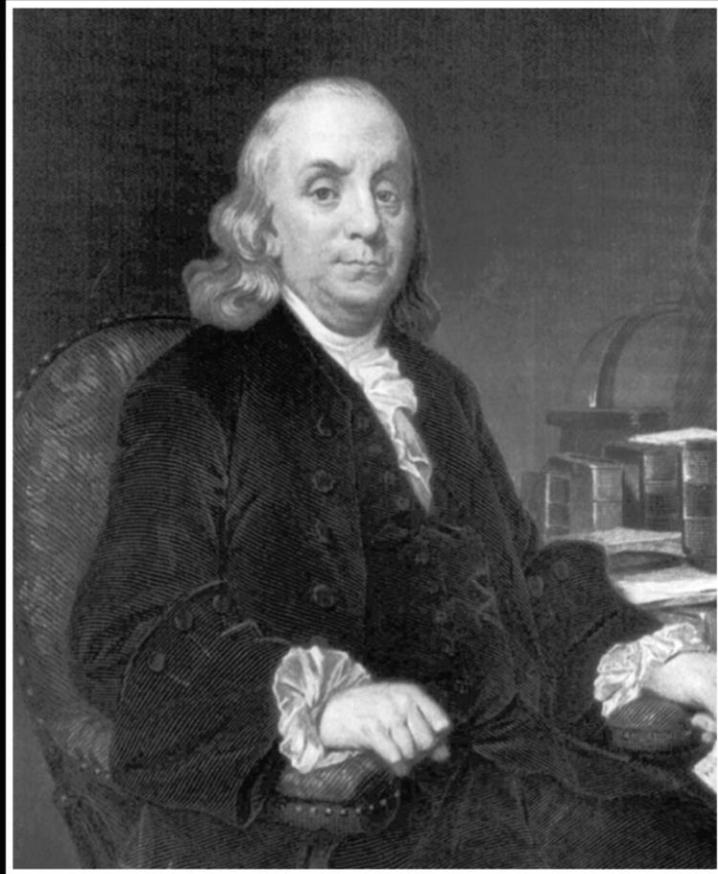
- You will get out of this course what you put into it



# 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

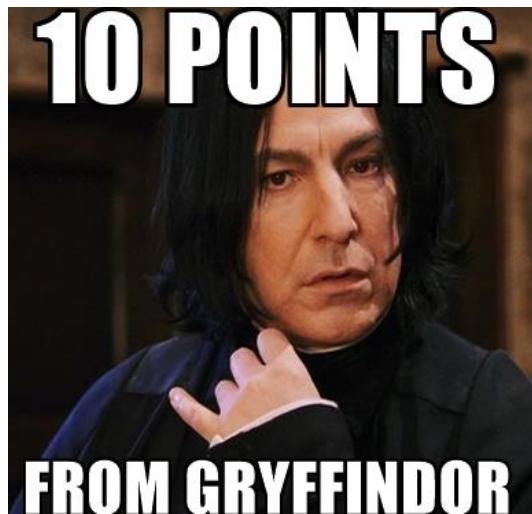


---

See [piazza.com/vanderbilt/spring2019/cs891](https://piazza.com/vanderbilt/spring2019/cs891)

# 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
  - No laptops/phones in class unless explicitly allowed

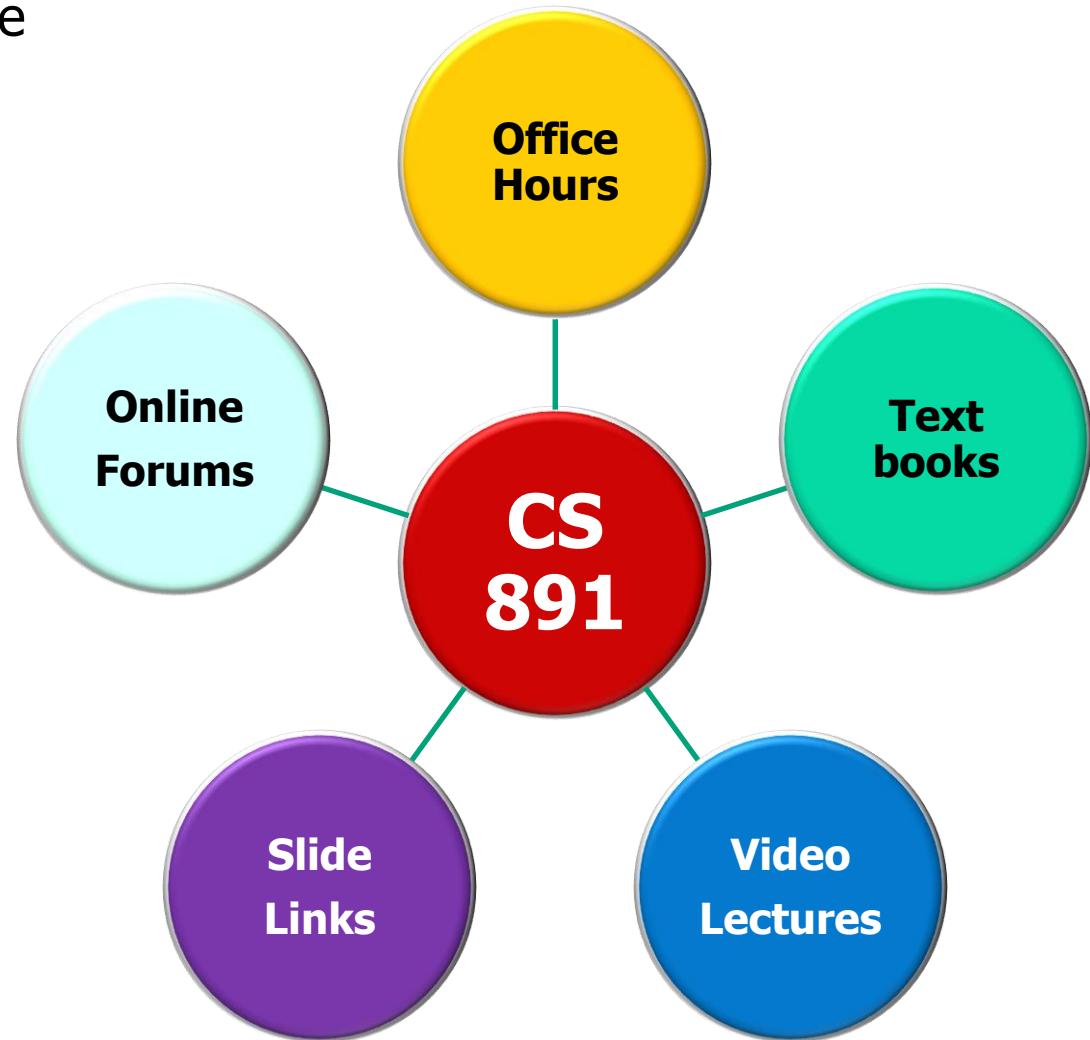


Failure to comply with this rule will cost you participation points..

# 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
  - No laptops/phones in class unless explicitly allowed
  - Avail yourself of available resources



# 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
  - No laptops/phones in class unless explicitly allowed
  - 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

---

- 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
  - No laptops/phones in class unless explicitly allowed
  - Avail yourself of available resources
  - There are abundant opportunities!



---

See [www.naceweb.org/job-market/compensation/  
the-top-paid-majors-for-the-class-of-2018](http://www.naceweb.org/job-market/compensation/the-top-paid-majors-for-the-class-of-2018)

# Summary

---

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

