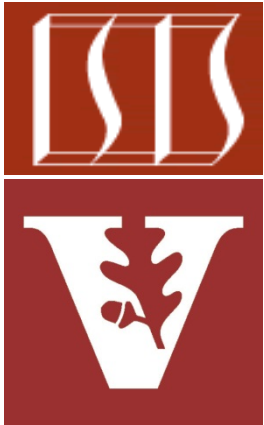


CS 891 Overview & Logistics

Douglas C. Schmidt

d.schmidt@vanderbilt.edu

www.dre.vanderbilt.edu/~schmidt



Professor of Computer Science

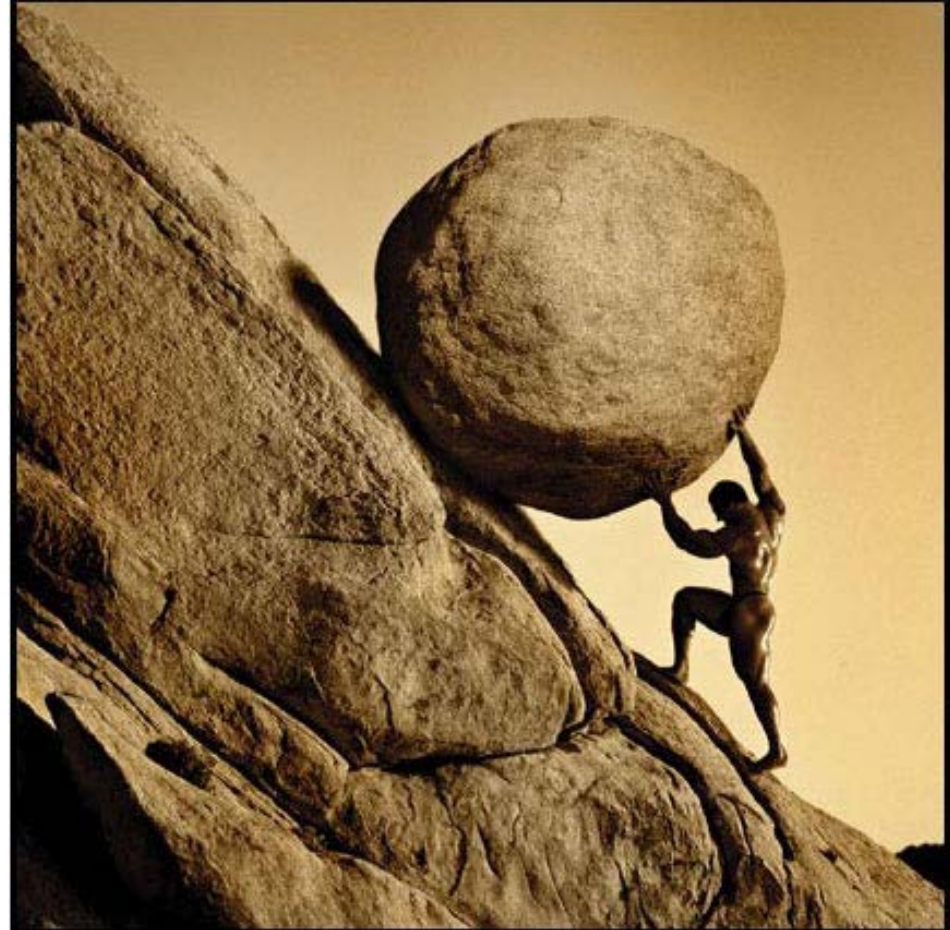
Institute for Software
Integrated Systems

Vanderbilt University
Nashville, Tennessee, USA



Learning Objectives in this Lesson

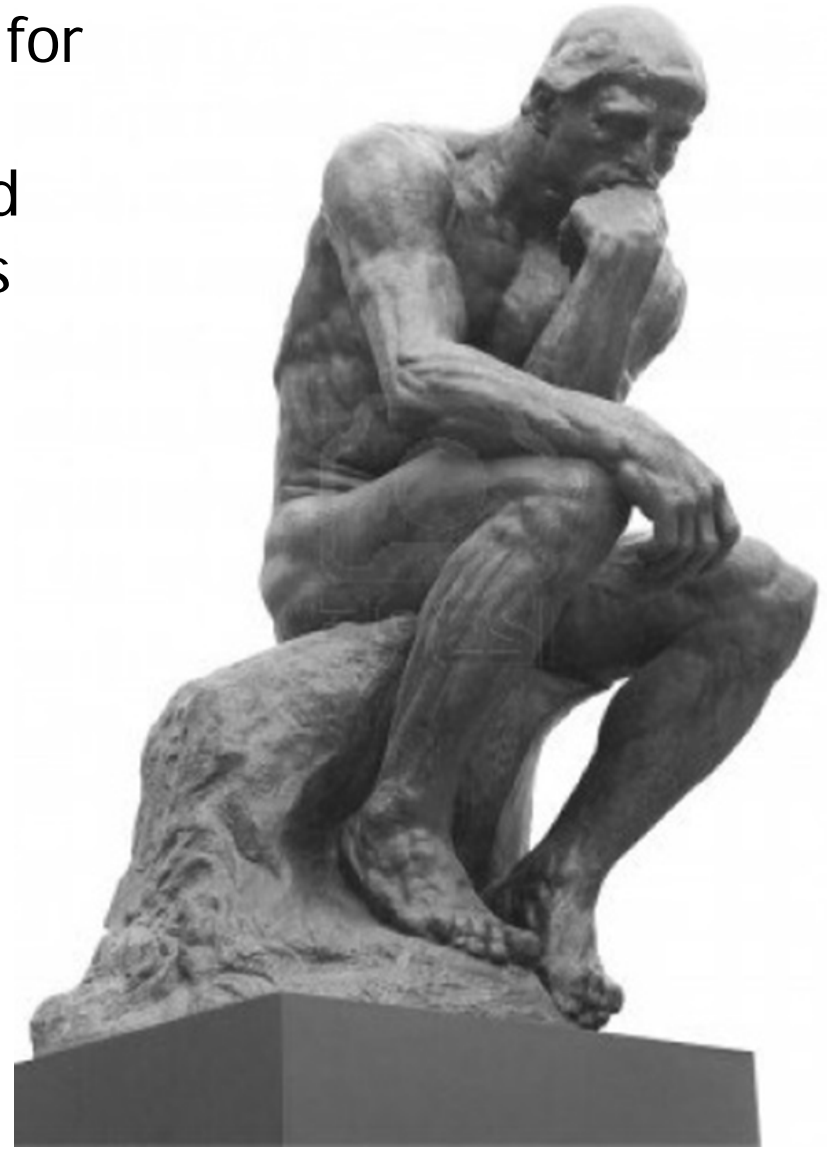
- Understand the course topics & logistics
 - Course philosophy
 - Course contents
 - Structure of the lecture material
 - Overview of the assignments & assessments
 - Setting up the Java & Android IDE on Android Studio
 - Accessing Android & Java source code



Course Philosophy

Course Philosophy

- Effective techniques & practices for developing concurrent & parallel 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 & resilientby applying time-proven software patterns & object-oriented & functional design & programming techniques

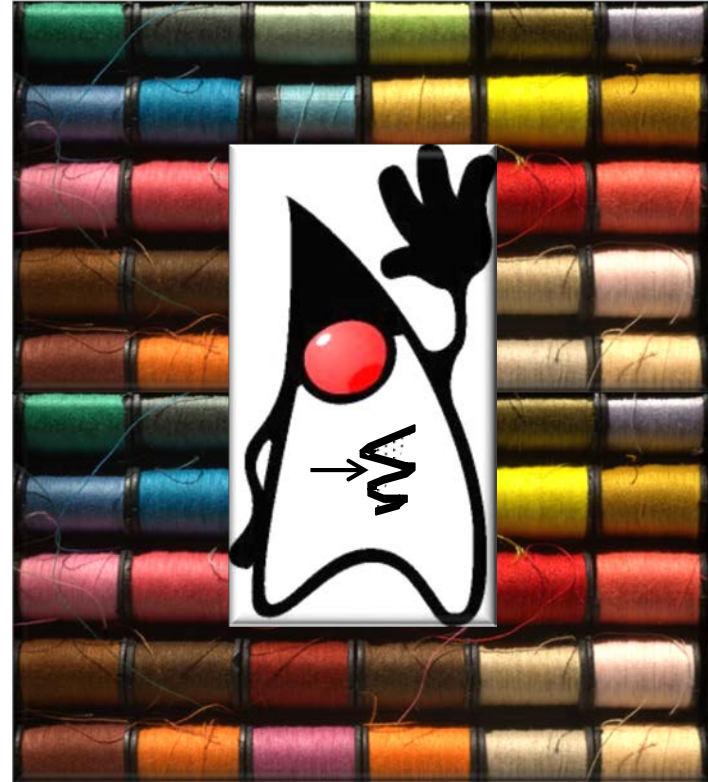


This course involves a lot of hands-on software development & testing

Summary of the Course Contents

Summary of Course Contents

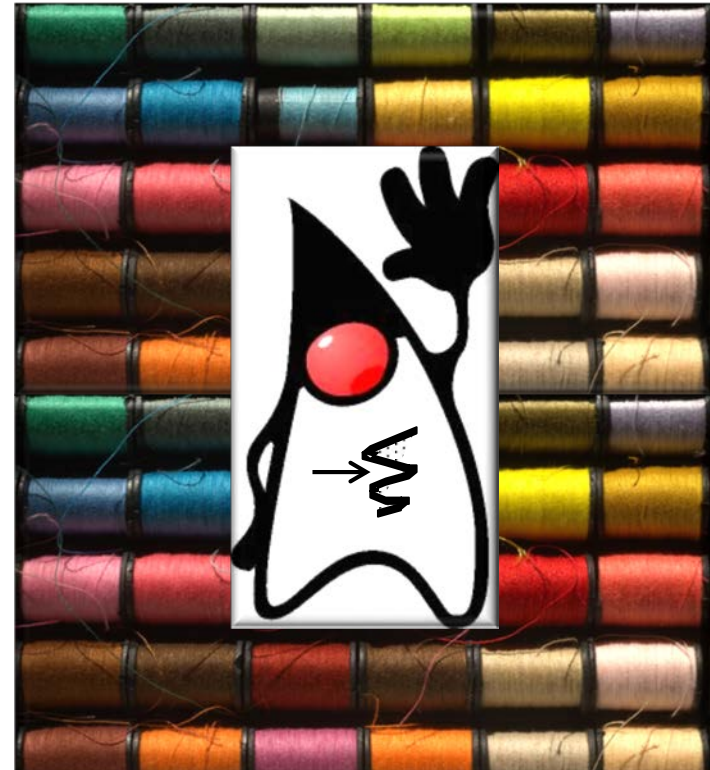
- Key Java 8 concurrency & parallelism frameworks



Including Java 8 object-oriented & functional programming language features

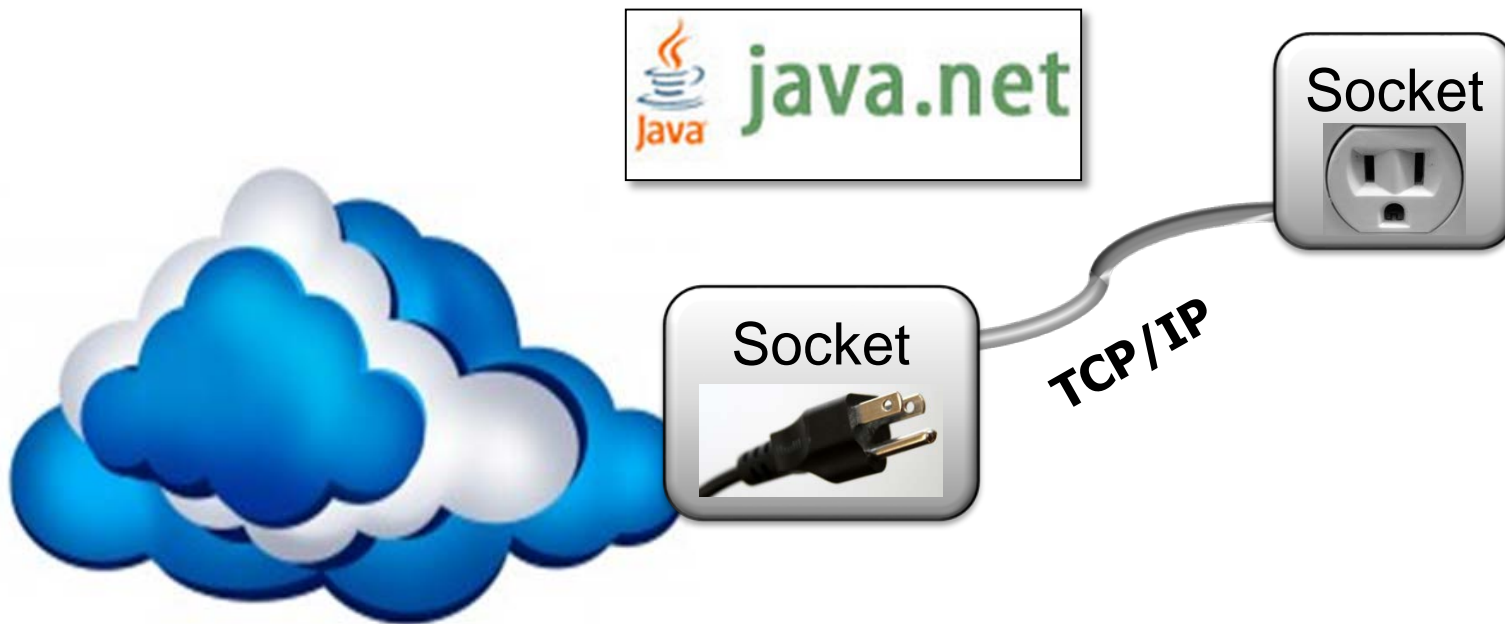
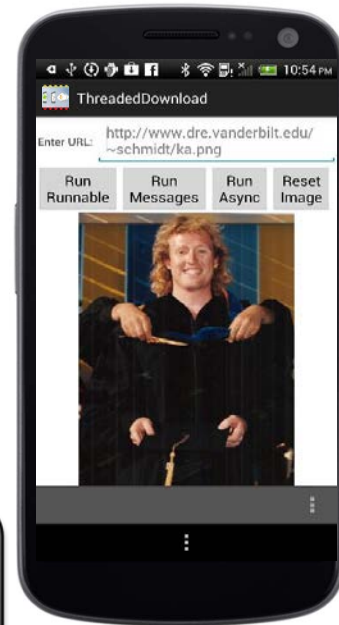
Summary of Course Contents

- Key Java 8 concurrency & parallelism frameworks
- Some Android UI & inter-process communication (IPC) mechanisms



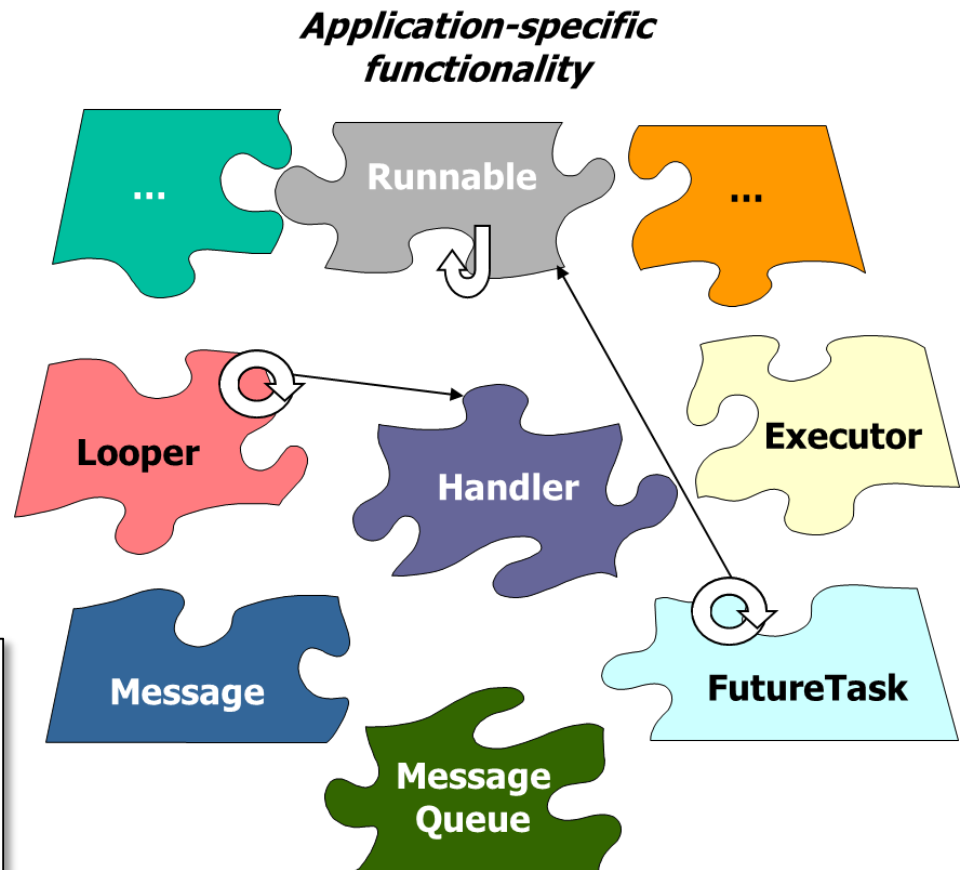
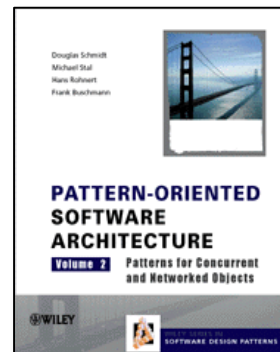
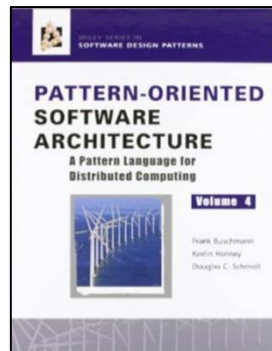
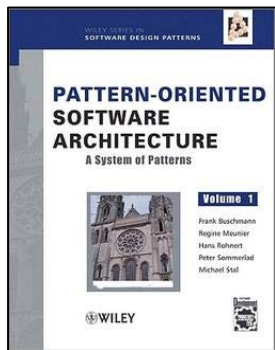
Summary of Course Contents

- Key Java 8 concurrency & parallelism frameworks
- Some Android UI & inter-process communication (IPC) mechanisms
- Some mobile & web communication mechanisms



Summary of Course Contents

- Key Java 8 concurrency & parallelism frameworks
- Some Android UI & inter-process communication (IPC) mechanisms
- Some mobile & web communication mechanisms
- Patterns/frameworks for concurrent & networked programming



Summary of Course Contents

- Key Java 8 concurrency & parallelism frameworks
- Some Android UI & inter-process communication (IPC) mechanisms
- Some mobile & web communication mechanisms
- Patterns/frameworks for concurrent & networked programming
- We assume you know (or can quickly learn) Android, Java, & 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 object oriented & functional programming features	<ul style="list-style-type: none">• Coverage of basic & advanced Java 8 programming features, e.g.<ul style="list-style-type: none">• Abstraction, inheritance, & polymorphism• Lambda expressions, method references, & functional interfaces

Structure of the Lecture Material

- This course has four main modules

Section	Topics
Java object oriented & functional programming features	<ul style="list-style-type: none">• Coverage of basic & advanced Java 8 programming features, e.g.<ul style="list-style-type: none">• Abstraction, inheritance, & polymorphism• Lambda expressions, method references, & functional interfaces
Java Concurrency & Parallelism	<ul style="list-style-type: none">• Coverage of basic & advanced Java 8 concurrency & parallelism frameworks, e.g.<ul style="list-style-type: none">• Java parallel (& sequential) streams• Java completable futures

Structure of the Lecture Material

- This course has four main modules

Section	Topics
Java object oriented & functional programming features	<ul style="list-style-type: none">• Coverage of basic & advanced Java 8 programming features, e.g.<ul style="list-style-type: none">• Abstraction, inheritance, & polymorphism• Lambda expressions, method references, & functional interfaces
Java Concurrency & Parallelism	<ul style="list-style-type: none">• Coverage of basic & advanced Java 8 concurrency & parallelism frameworks, e.g.<ul style="list-style-type: none">• Java parallel (& sequential) streams• Java completable futures
Mobile ↔ Web Communication	<ul style="list-style-type: none">• HTTP communication & parsing libraries


Structure of the Lecture Material

- This course has four main modules

Section	Topics
Java object oriented & functional programming features	<ul style="list-style-type: none">• Coverage of basic & advanced Java 8 programming features, e.g.<ul style="list-style-type: none">• Abstraction, inheritance, & polymorphism• Lambda expressions, method references, & functional interfaces
Java Concurrency & Parallelism	<ul style="list-style-type: none">• Coverage of basic & advanced Java 8 concurrency & parallelism frameworks, e.g.<ul style="list-style-type: none">• Java parallel (& sequential) streams• Java completable futures
Mobile ↔ Web Communication	<ul style="list-style-type: none">• HTTP communication & parsing libraries
Software Patterns	<ul style="list-style-type: none">• Concurrency & communication patterns

Structure of the Lecture Material

- This course has four main modules

Section	Topics
Java object oriented & functional programming features	 <p>Coverage of basic & advanced Java 8 programming features, e.g.</p> <ul style="list-style-type: none">• Abstraction, inheritance, & polymorphism• Lambda expressions, method references, & functional interfaces
Java Concurrency & Parallelism	<ul style="list-style-type: none">• Coverage of basic & advanced Java 8 concurrency & parallelism frameworks, e.g.<ul style="list-style-type: none">• Java parallel (& sequential) streams• Java completable futures
Mobile \leftrightarrow Web Communication	<ul style="list-style-type: none">• HTTP communication & parsing libraries
Software Patterns	<ul style="list-style-type: none">• Concurrency & communication patterns

We will bounce around a bit when covering these topics

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/cs891#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



Structure of the Lecture Material

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



All quizzes (& the final)
are "closed book"

Structure of the Lecture Material

- There will be a bi-weekly quizzes on material covered in the lectures
- 1st quiz will be next Wednesday



Structure of the Lecture Material

- There will be a bi-weekly quizzes on material covered in the lectures
 - 1st quiz will be next Wednesday
- We'll hand 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 a bi-weekly quizzes on material covered in the lectures
 - 1st quiz will be next Wednesday
- We'll hand back & review quizzes at the start of the next class



I recommend that you study for quizzes by reviewing slides & watching screencasts available at www.dre.vanderbilt.edu/~schmidt/cs891#lectures

Structure of the Lecture Material

- There will be a cumulative final exam that covers all the lectures

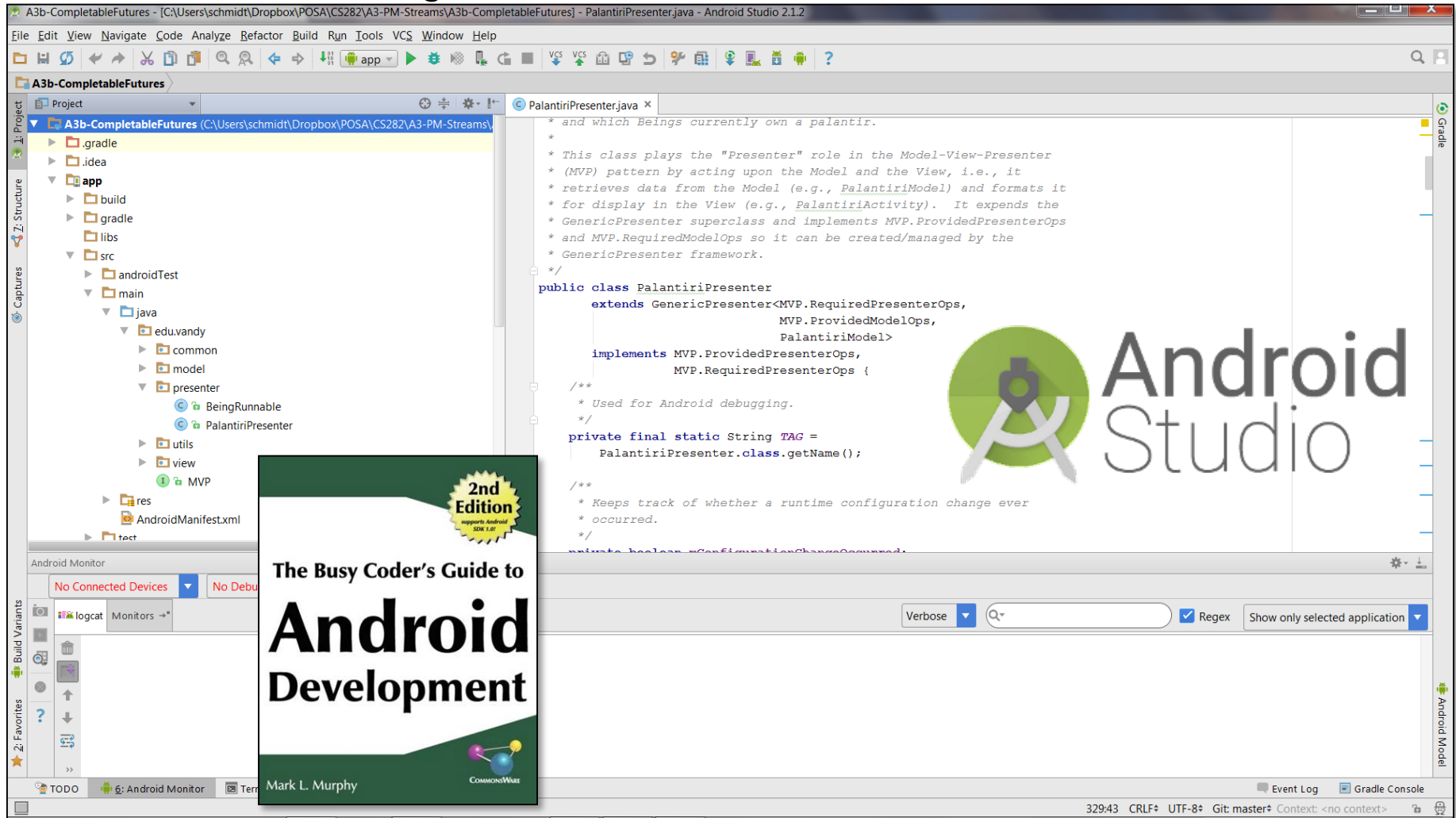


The final exam will be held 9am to noon,
Saturday, December 9th in this room

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.x & Android Nougat 7.1 (API 25)

Overview of Assignments & Assessments

- Programming assignments should be written in Java 8 using Android Studio
- Please install the Java 8 runtime environment (JRE)



See 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

Overview of Assignments & Assessments

- Android programming assignments must be submitted using Android Studio
- Please install Android 7.1 Nougat (API level 25)



See en.wikipedia.org/wiki/Android_Nougat

Overview of Assignments & Assessments

- All source code for assignments & examples available at GitHub

The screenshot shows the GitHub repository page for `douglascraigschmidt / CS891`. At the top, there are buttons for `Unwatch` (1), `Star` (1), and `Fork` (0). Below this is a navigation bar with tabs for `Code`, `Issues` (0), `Pull requests` (0), `Projects` (0), `Wiki`, `Settings`, and `Insights`. The main content area contains a description: "Contains examples and assignments for my CS 891 course at Vanderbilt University, which can be accessed via <http://www.dre.vanderbilt.edu/~schmidt/cs891>." Below the description are statistics: `5 commits`, `1 branch`, `0 releases`, and `1 contributor`. There is a progress bar below these statistics. At the bottom of the repository page, there are buttons for `Branch: master`, `New pull request`, `Create new file`, `Upload files`, `Find file`, and `Clone or download`. A section titled `douglascraigschmidt updates` shows the latest commit `7e107e4` 17 hours ago. Below this, a section titled `ex updates` shows updates 17 hours ago. At the bottom, there is a prompt to "Help people interested in this repository understand your project by adding a README." with a button `Add a README`.

Go to GitHub at github.com/douglascraigschmidt/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.

A screenshot of the GitLab 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 this paragraph is another paragraph of white text: "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 of the page are two buttons: a light blue button with the text "GitLab Community Edition" and a green button with the text "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](#)

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

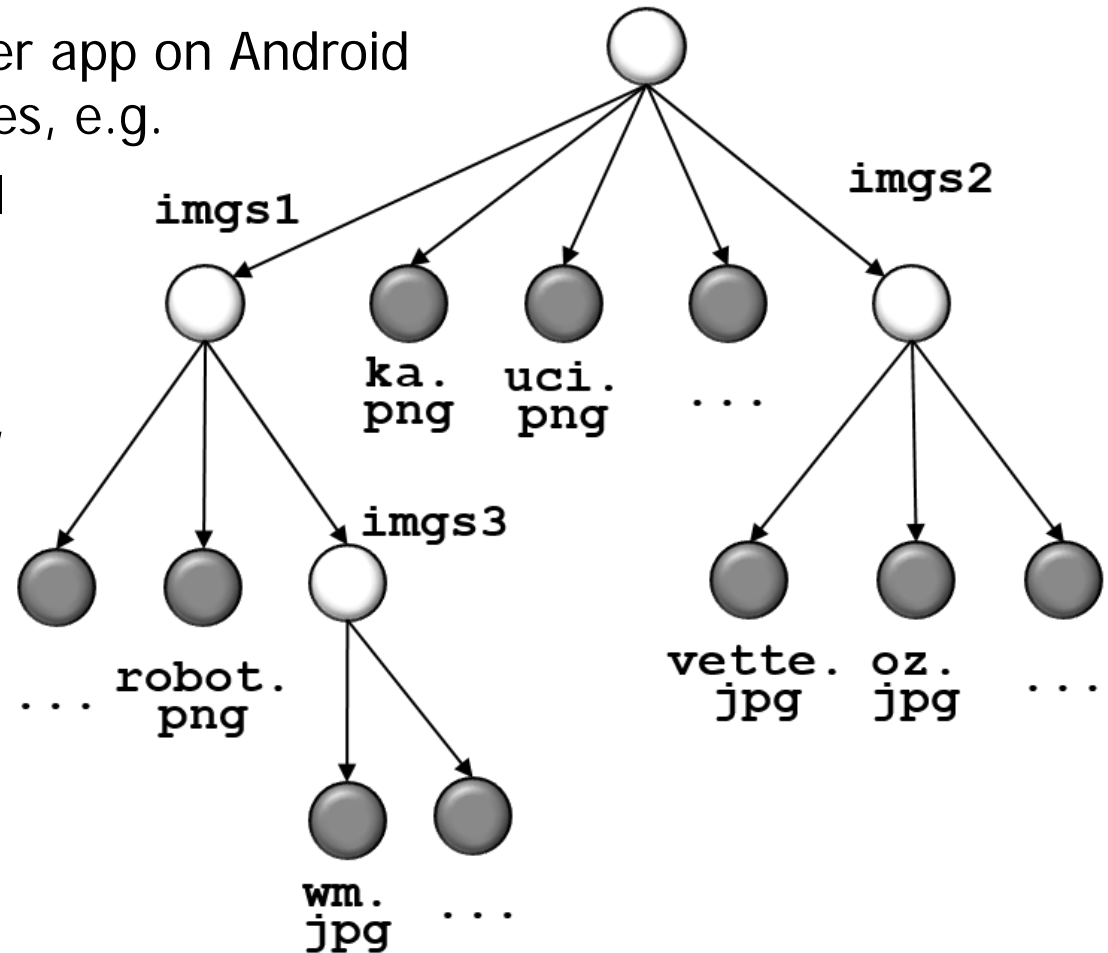
- Assignments will provide a range of experience with Java 8 & Android concurrent/parallel programs



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

Overview of Assignments & Assessments

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



The topics covered by the assignments may change a bit during the semester

Overview of Assignments & Assessments

- Assignment assessments will be done via reviews by course staff



Overview of Assignments & Assessments

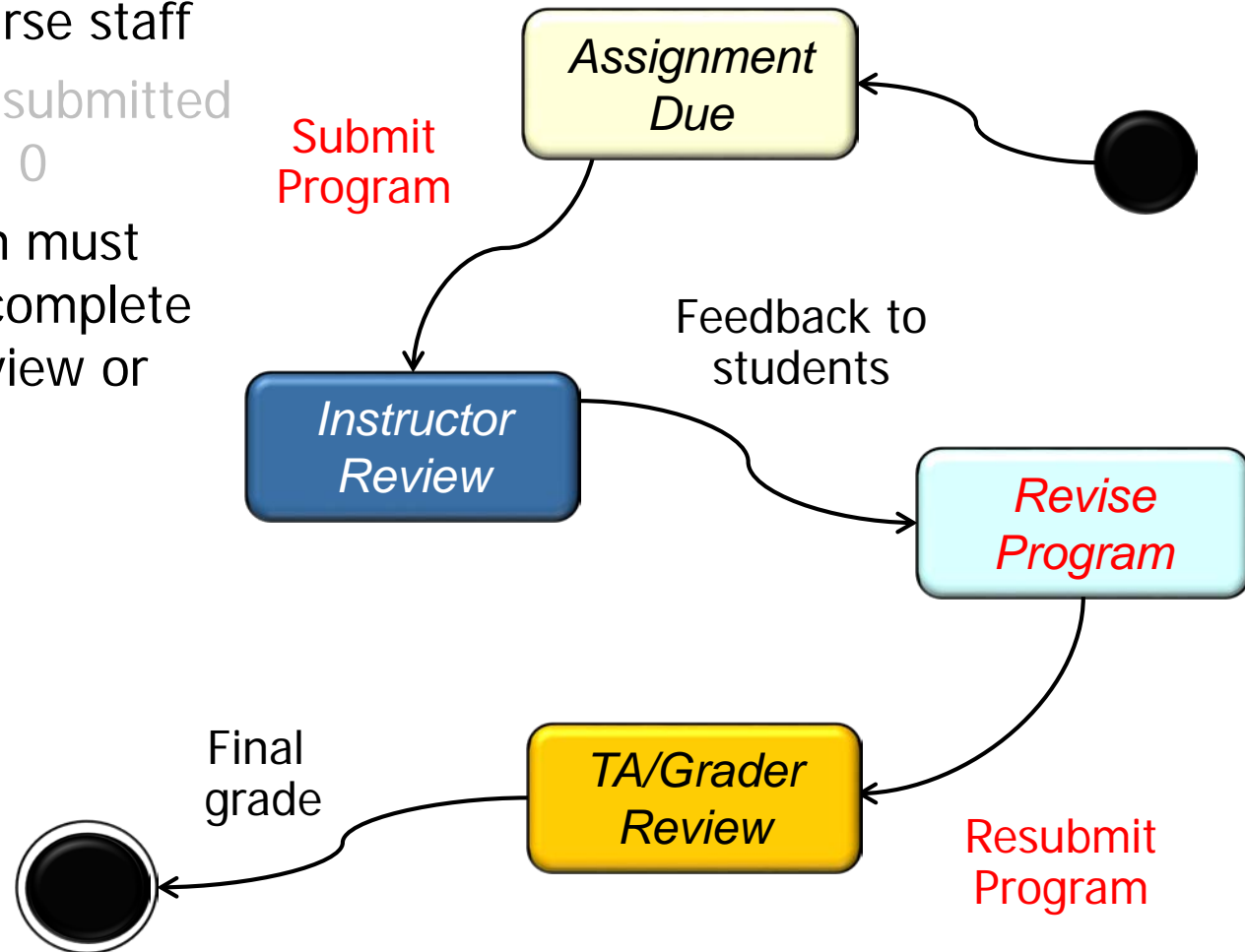
- Assignment assessments will be done via reviews by course staff
- Assignments *must* be submitted on time or you'll get a 0



See github.com/douglasraigschmidt/CS891/wiki/CS-891-FAQ

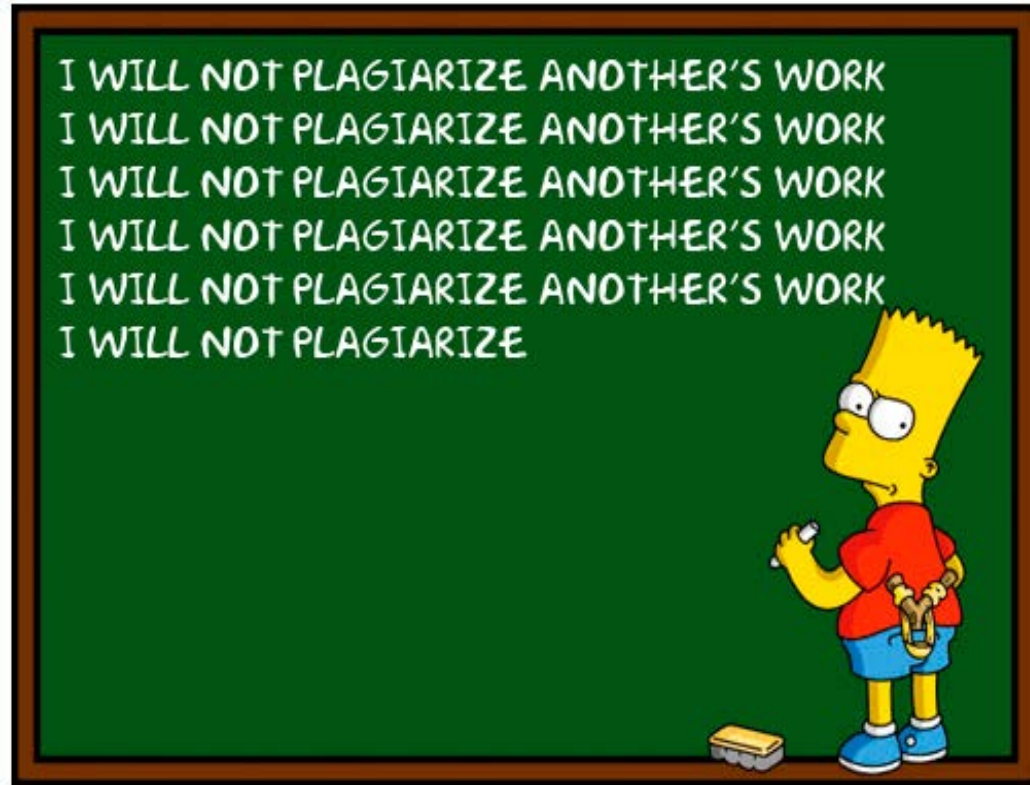
Overview of Assignments & Assessments

- 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
- Work *must* be your own
 - This goes for quizzes & programming assignments



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

Overview of Assignments & Assessments

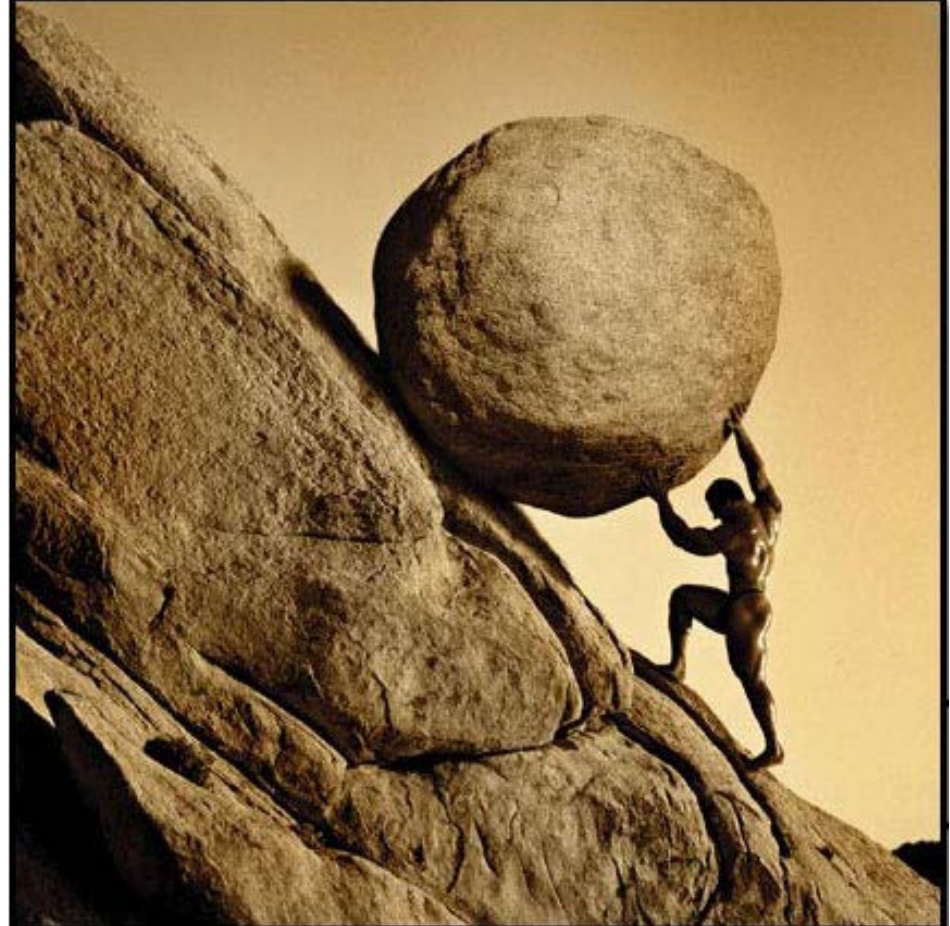
- Assessment criteria

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/cs891/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

IMPORTANT



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 participate!!!!

Setting Up the Android & Java IDE on Android Studio

Installing Eclipse Java/Android Developer Tools

- See 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
2.3.3 FOR WINDOWS (1,926 MB)

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	 jdk-8u25-linux-i586.rpm
Linux x86	154.88 MB	 jdk-8u25-linux-i586.tar.gz
Linux x64	135.6 MB	 jdk-8u25-linux-x64.rpm

[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.x

Be an Early Adopter

Get early access to the latest features and improvements in Android Studio by downloading the Android Studio Preview.

The Android Studio Preview can run side-by-side with your stable version, so you can work on the same projects in both versions.

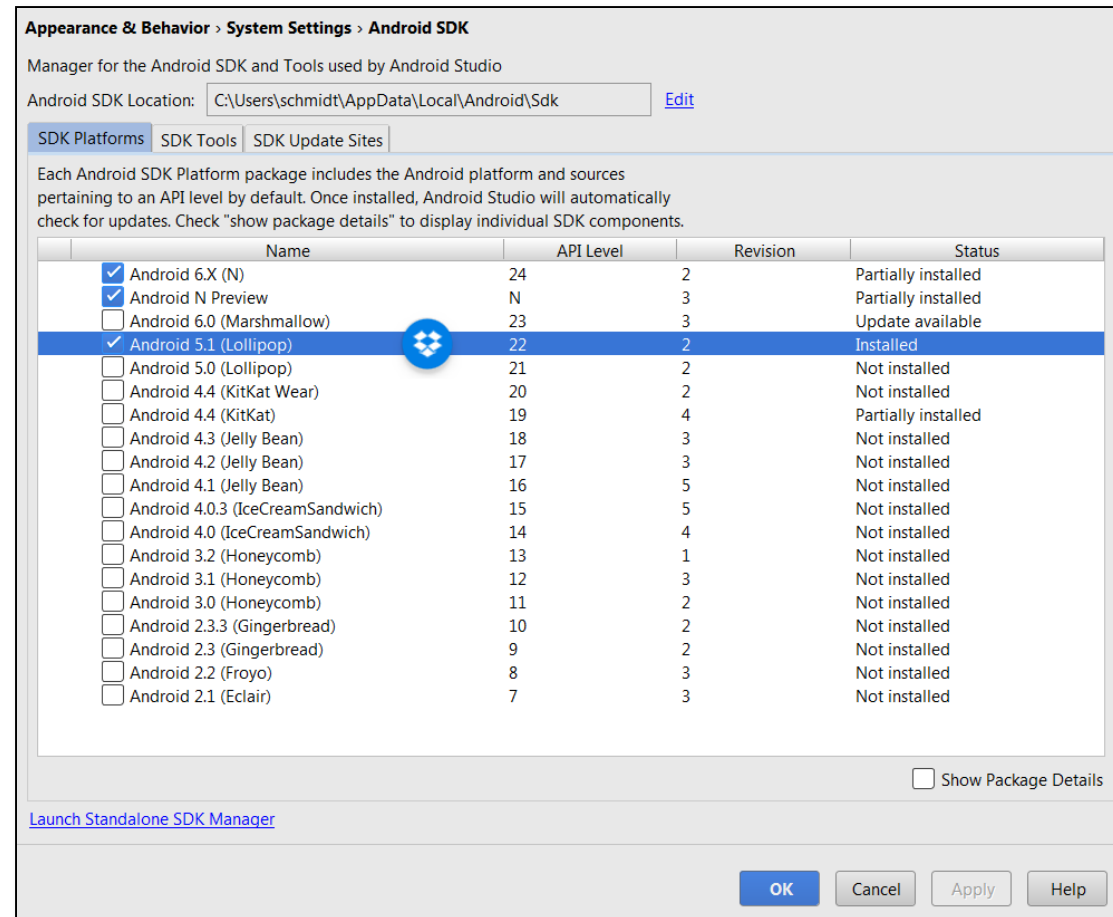
Important: If you've upgraded a project from using an alpha version of Android plugin 3.0.0 to using a beta version of the plugin, you'll need to first clean your project by selecting **Build > Clean Project** from the menu bar.

DOWNLOAD 3.0 BETA 2
FOR WINDOWS (704 MB)

developer.android.com/studio/preview

Add Components to the SDK

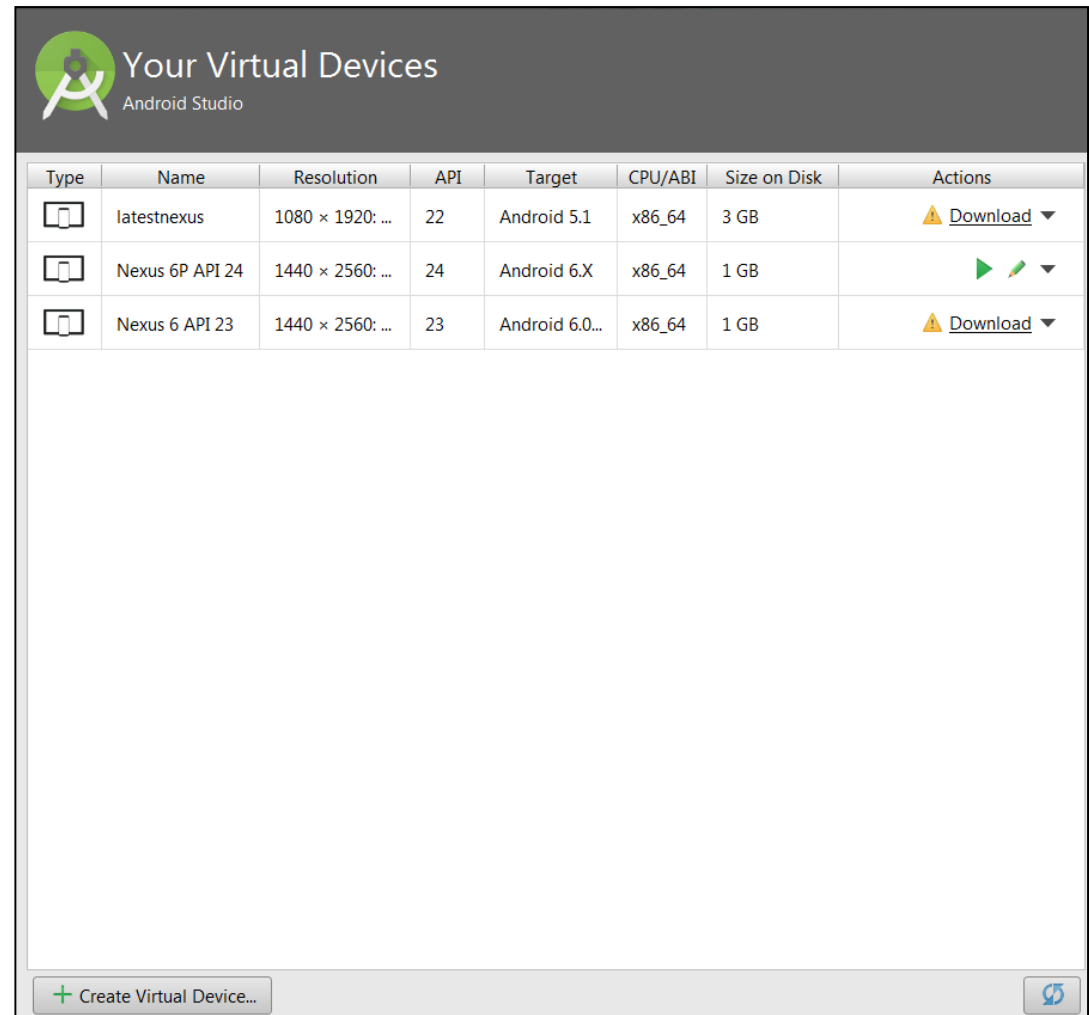
- Launch the Android Studio SDK Manager
- Select “Nougat” version of Android (7.1, API 25)



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

Add Components to the SDK

- Launch the Android Studio Virtual Device Manager
- Create an Android API 25 emulator



[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

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
 - grepcode.com/file/repository.grepcode.com/java/root/jdk/openjdk/8-b132/java



The screenshot shows the Java.net website. At the top left is the Java logo and "Java.net" text, followed by the tagline "The Source for Java Technology Collaboration". On the top right are links for "Login", "Register", and "Help". A sidebar on the left under the heading "JDK 8" contains links for "Downloads", "Feedback Forum", "OpenJDK", and "Planet JDK". The main content area features the "JDK 8 Project" heading with the subtitle "Building the next generation of the JDK platform". Below this, there are three columns of text. The first column, titled "JDK 8 snapshot builds", lists links for "Download 8u40 early access snapshot builds", "Source code (instructions)", "Official Java SE 8 Reference Implementations", and "Early Access Build Test Results (instructions)". The second column, titled "We Want Contributions!", contains a paragraph about contributing to the platform. The third column, titled "Feedback", contains two paragraphs about using the Project Feedback forum and submitting bug reports.

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.

Accessing Java & Android Source Code

- Java 8 source code is available
 - For Browsing
greppcode.com/file/repository.greppcode.com/java/root/jdk/openjdk/8-b132/java
 - For downloading
jdk8.java.net/download.html



The screenshot shows the Java.net website. At the top left is the Java logo and "Java.net" text, followed by the tagline "The Source for Java Technology Collaboration". On the top right are links for "Login", "Register", and "Help". A sidebar on the left lists "JDK 8" with sub-links: "Downloads", "Feedback Forum", "OpenJDK", and "Planet JDK". The main content area features the "JDK 8 Project" heading with the subtitle "Building the next generation of the JDK platform". Below this, there are three columns: "JDK 8 snapshot builds" with a list of links (Download 8u40 early access snapshot builds, Source code (instructions), Official Java SE 8 Reference Implementations, Early Access Build Test Results (instructions)), "We Want Contributions!" with a paragraph about contributing, and "Feedback" with a paragraph about using the Project Feedback forum and submitting bugs.

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.

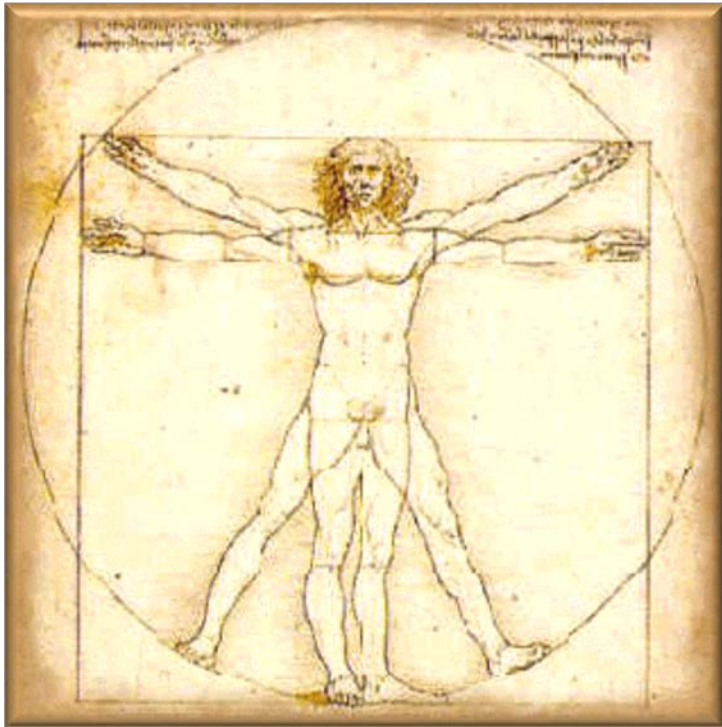
If you find bugs in a release, please submit them using the usual [Java SE bug reporting channels](#), not with the Issue tracker accompanying this project. Be sure to include complete version information from the output of the `java -version` command.

Summary



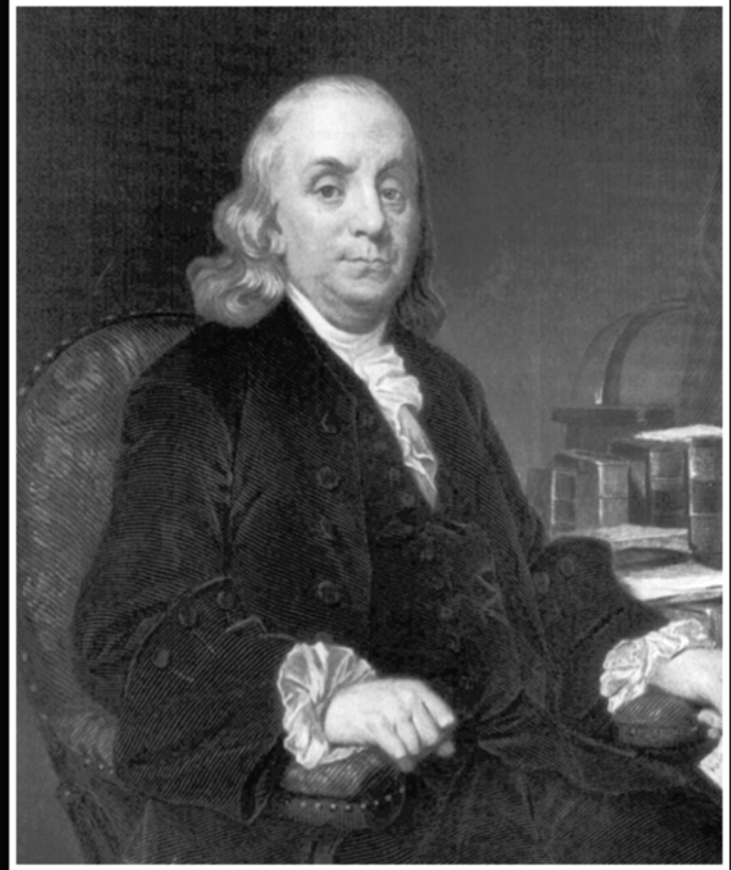
Summary

- 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

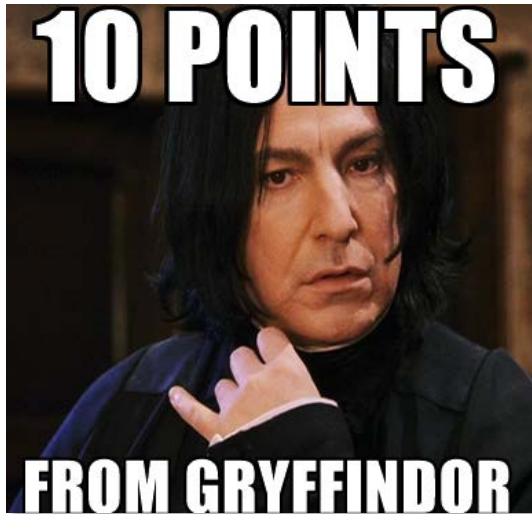
- You will get out of this course what you put into it
 - Be prepared to work hard
 - Do *not* miss deadlines...
- Participate in discussions in class & on piazza



See piazza.com/vanderbilt/fall2017/cs891/home

Summary

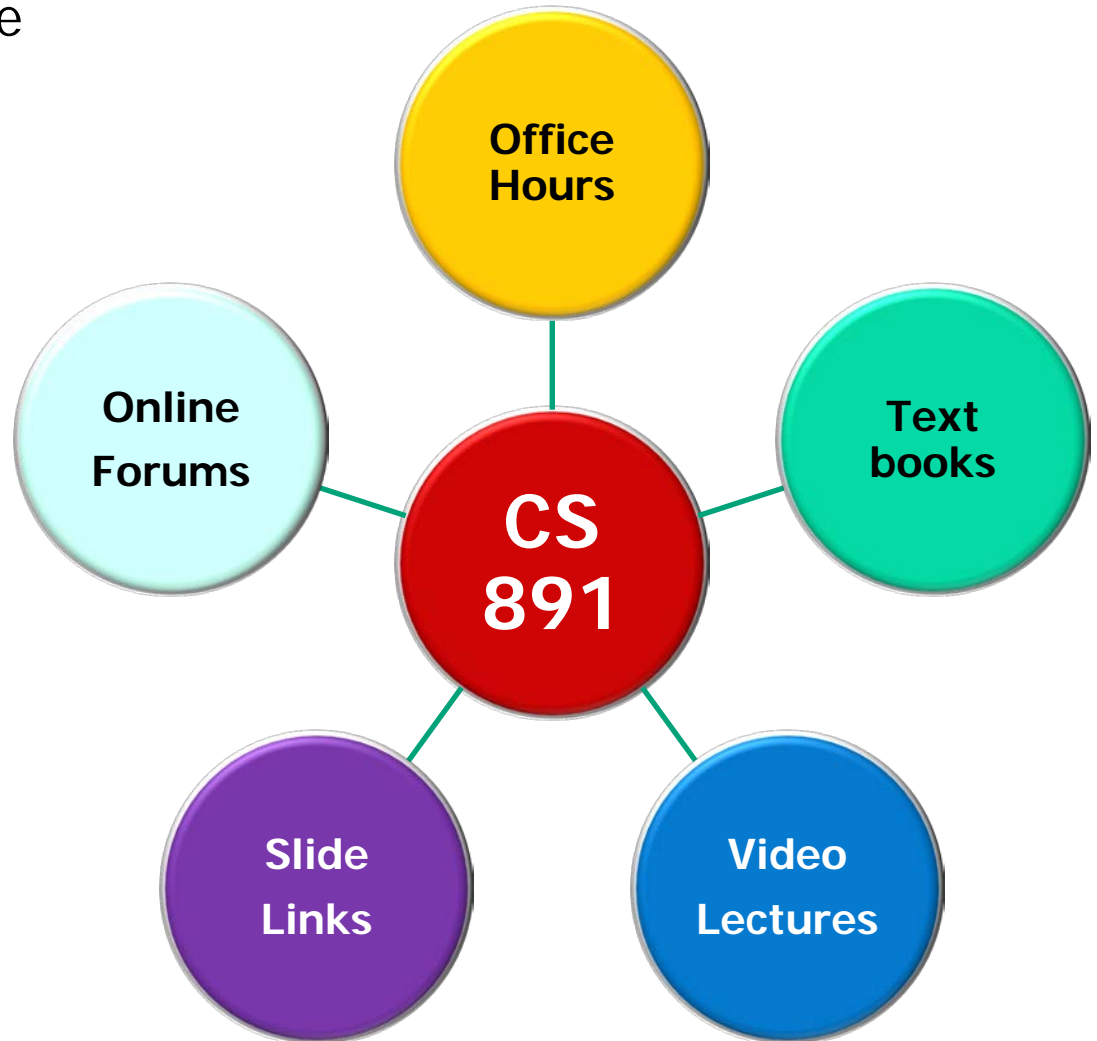
- You will get out of this course what you put into it
 - Be prepared to work hard
 - Do *not* miss deadlines...
 - Participate in discussions in class & on piazza
- No laptops/phones in class unless explicitly allowed



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



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

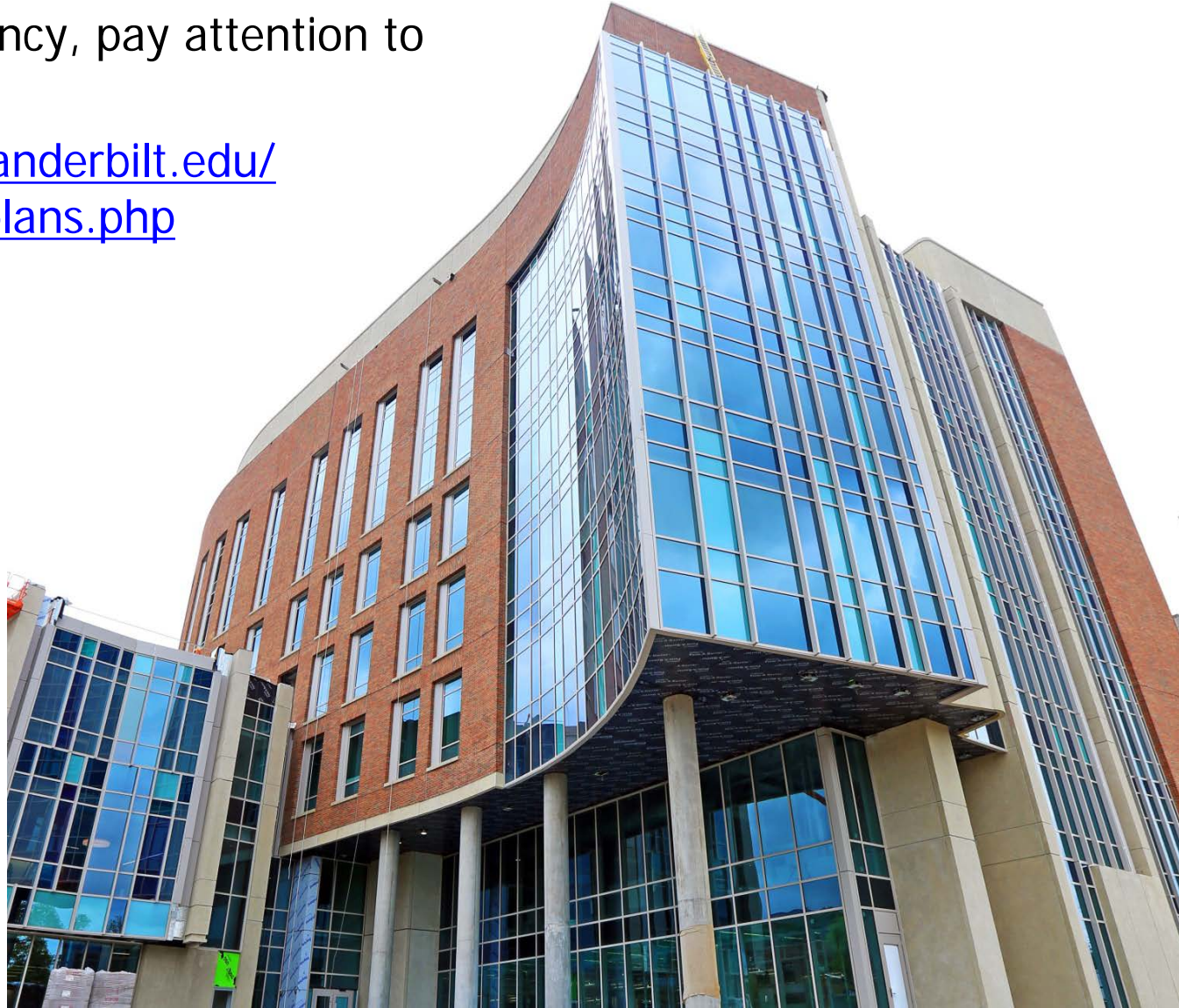
- You will get out of this course what you put into it
 - Be prepared to work hard
 - Do *not* miss deadlines...
 - Participate in discussions in class & on piazza
 - No laptops/phones in class unless explicitly allowed
 - Avail yourself of available resources
- There are abundant opportunities!



See www.forbes.com/sites/susanadams/2013/04/15/college-degrees-with-the-highest-starting-salaries-3

Summary

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



See video at www.youtube.com/watch?v=zXpFznXb_vI starting ~20 seconds