# CS 253: Parallel Functional Programming with Java & Android: Overview (Part 2)

Douglas C. Schmidt

d.schmidt@vanderbilt.edu

www.dre.vanderbilt.edu/~schmidt

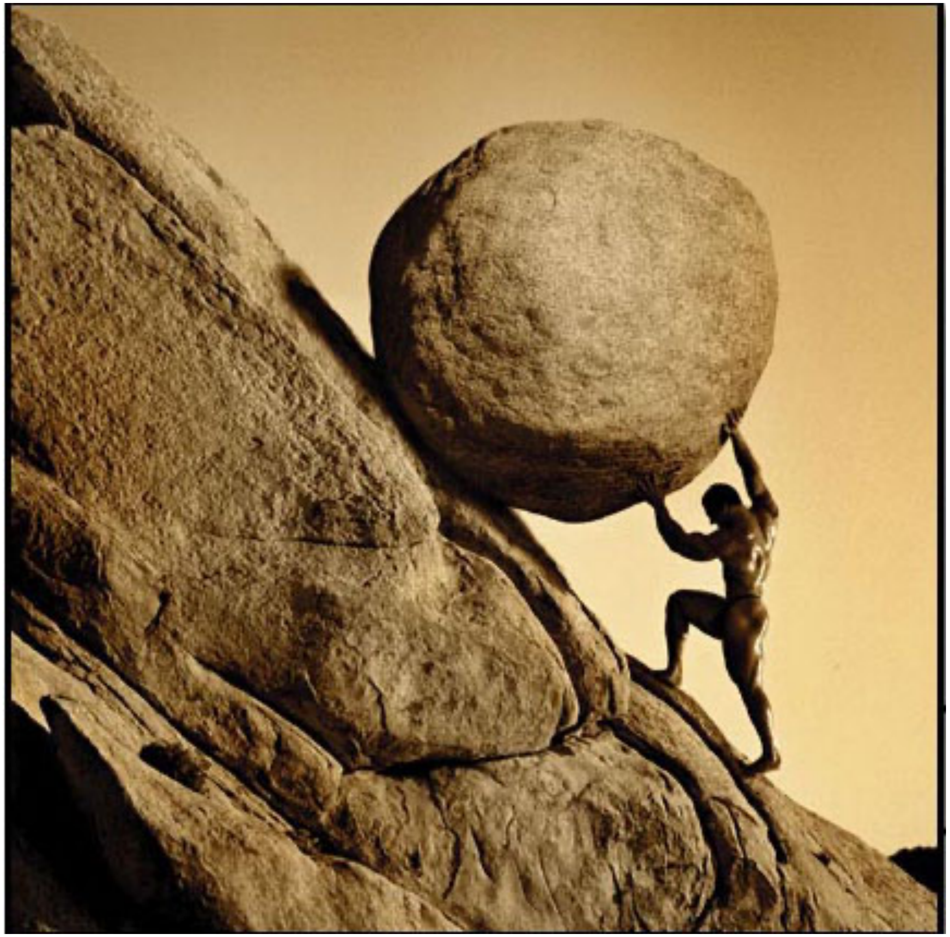
Professor of Computer Science

Institute for Software
Integrated Systems

Vanderbilt University
Nashville, Tennessee, USA

# Learning Objectives in this Lesson

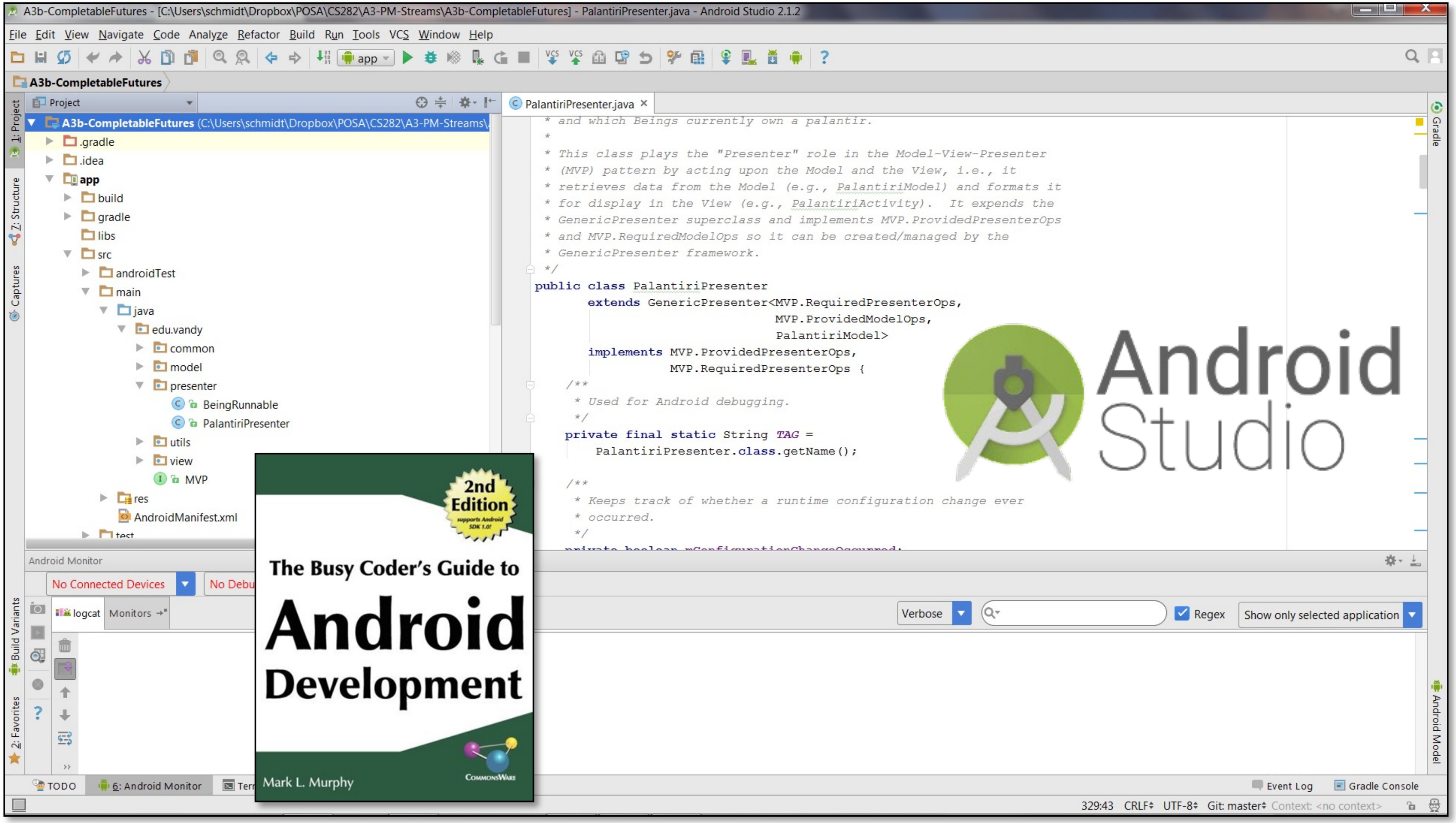
- Understand the course topics & logistics
  - Course philosophy
  - Course contents
  - Structure of the lecture material
  - Overview of the assignments & assessments



# Overview of Assignments & Assessments

# Overview of Assignments & Assessments

- Programming assignments are written in modern Java using Android Studio



```
 * 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 13 (API 33)

# Overview of Assignments & Assessments

- Programming assignments are written in modern Java using Android Studio
  - The Java runtime environment (JRE) is pre-installed with Android

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

# Overview of Assignments & Assessments

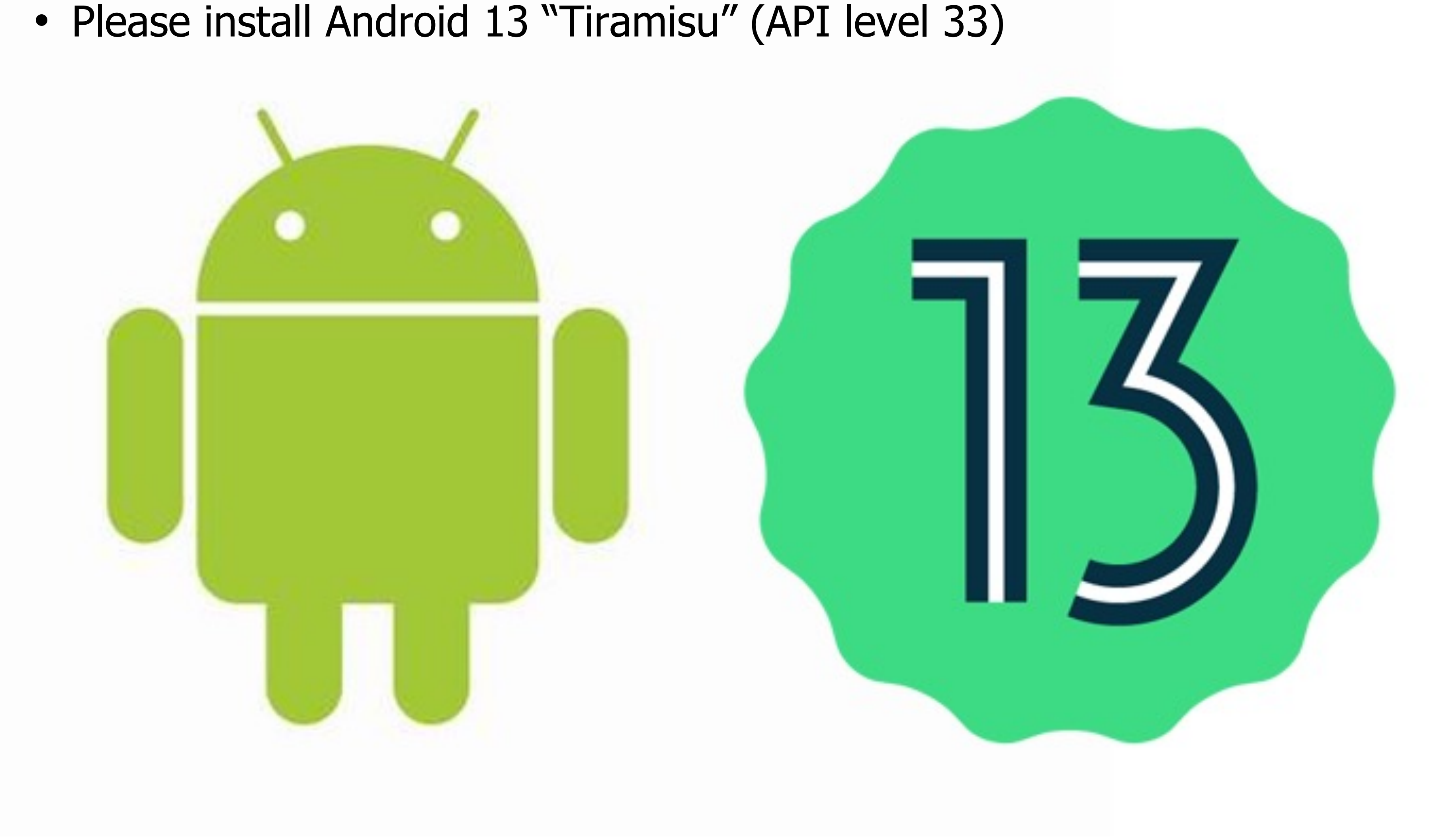
- Android programming assignments must be submitted using Android Studio

- 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/studio](developer.android.com/studio)

# Overview of Assignments & Assessments

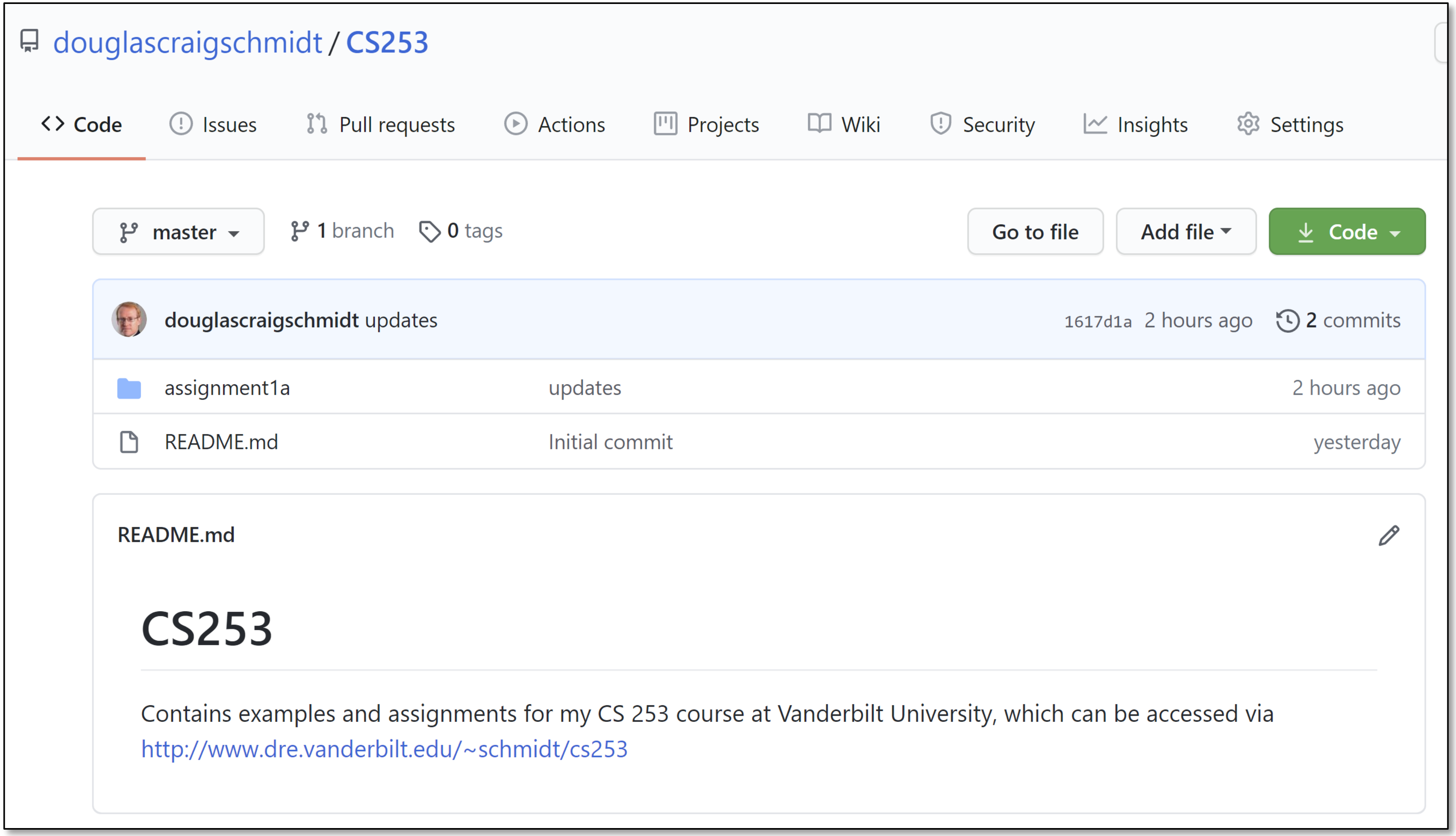
- Android programming assignments must be submitted using Android Studio
  - Please install Android 13 "Tiramisu" (API level 33)



See [en.wikipedia.org/wiki/Android_13](en.wikipedia.org/wiki/Android_13)

# Overview of Assignments & Assessments

- All source code for assignments & examples available at GitHub



douglascraigschmidt updates 1617d1a 2 hours ago 2 commits

| | | |
|---|---|---|
| assignment1a | updates | 2 hours ago |
| README.md | Initial commit | yesterday |

README.md

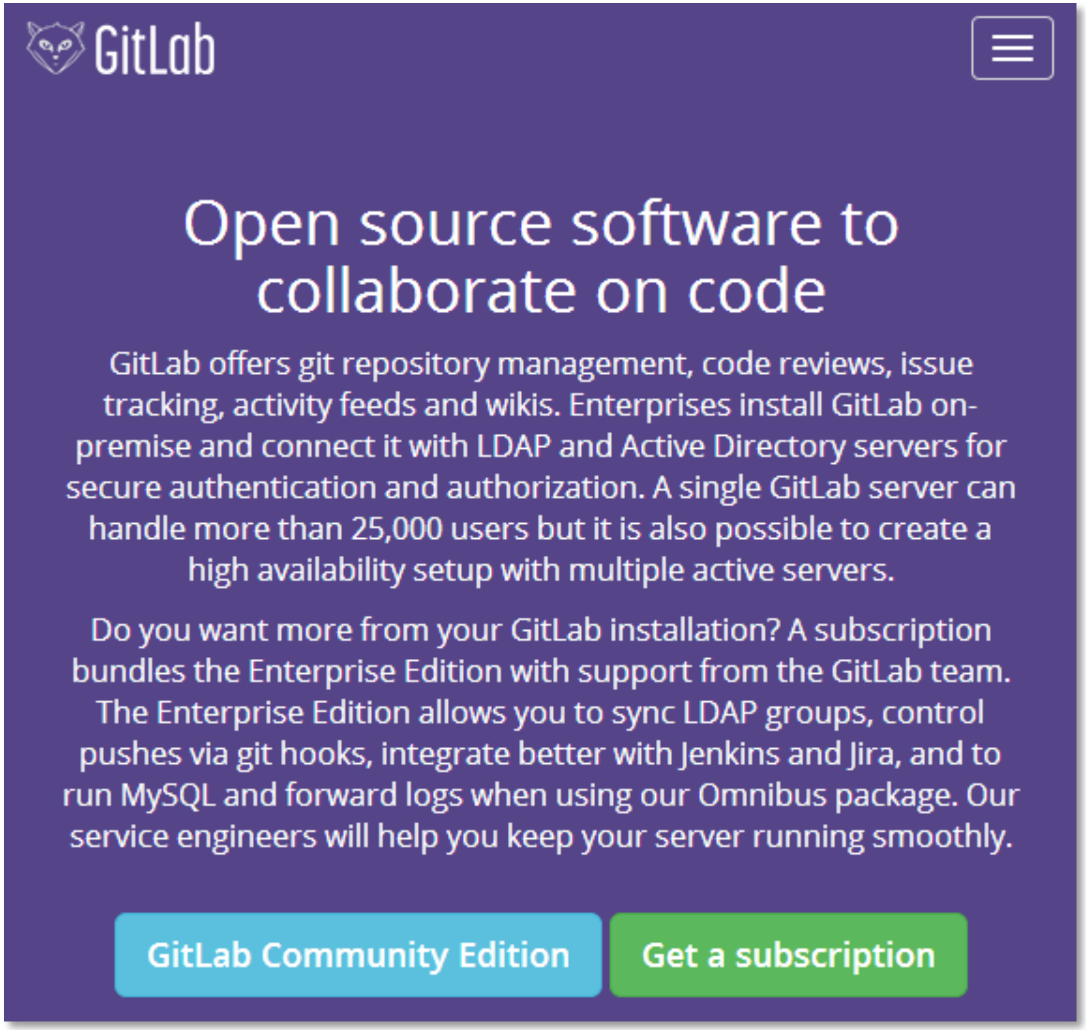
# CS253

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.

We'll discuss how to setup GitLab shortly

# 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.”

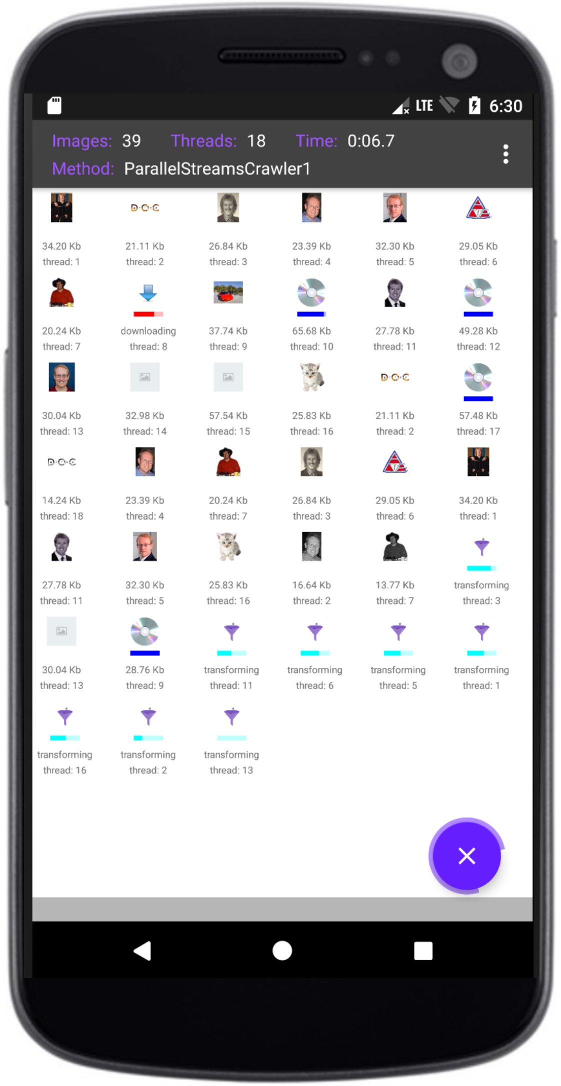
# Overview of Assignments & Assessments

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

Go to GitHub at github.com/douglascraigschmidt/CS253

# Overview of Assignments & Assessments

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



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

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

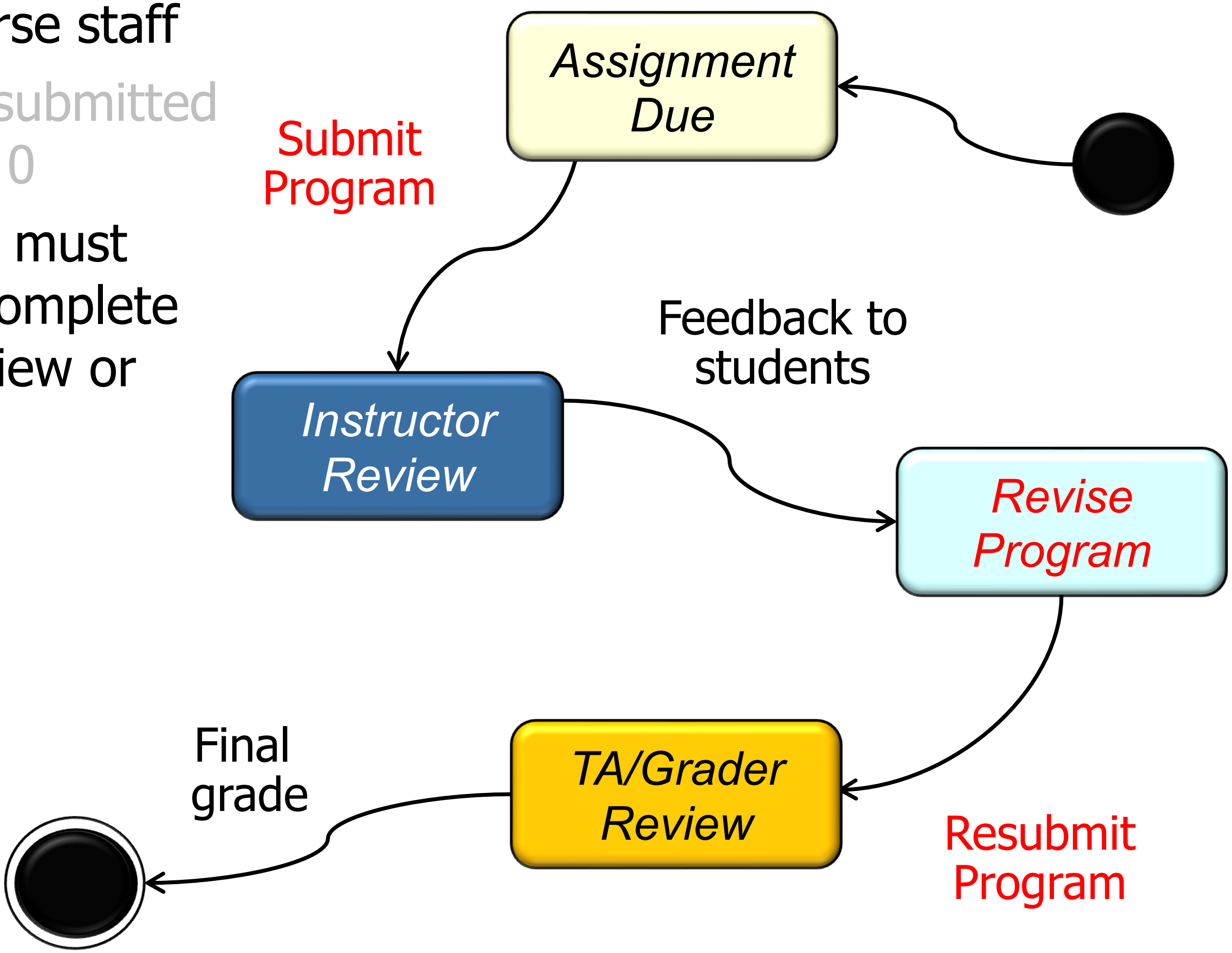
# Overview of Assignments & Assessments

- 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



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

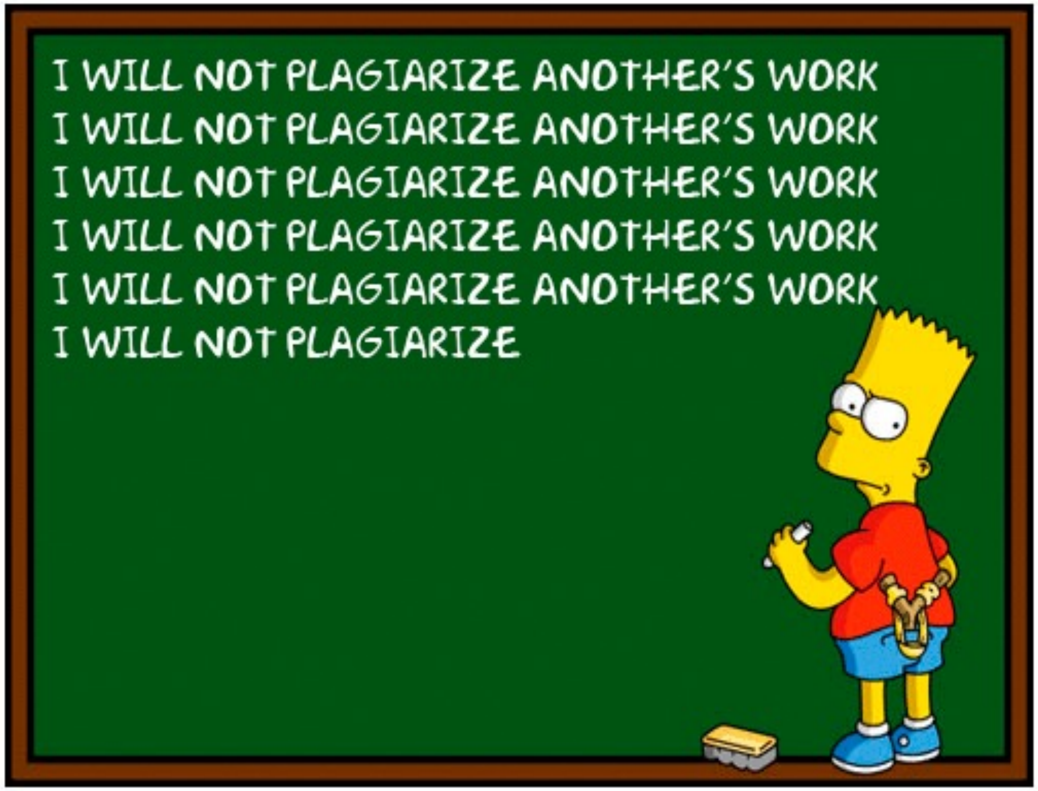
# Overview of Assignments & Assessments

- 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

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

# Overview of Assignments & Assessments

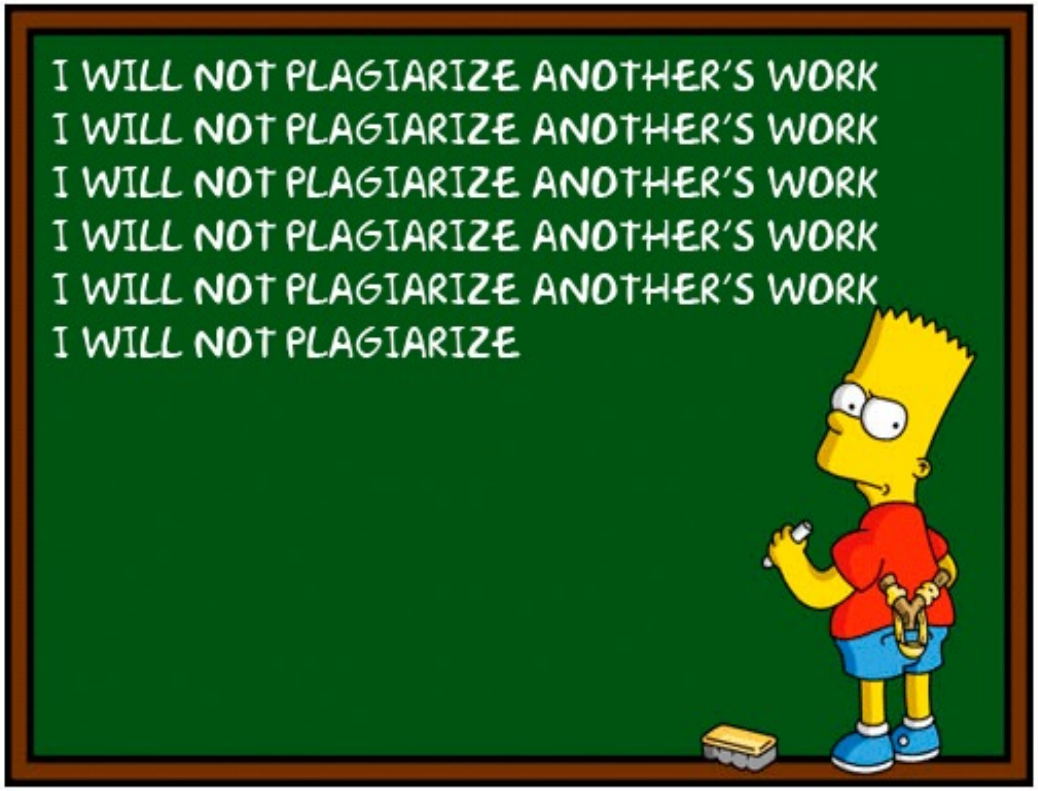
- Assignment assessments will be done via reviews by course staff
  - Assignments *must* be submitted on time or you'll get a 0
  - Your initial submission must compile & be largely complete or you won't get a review or a final grade
  - You will not receive a grade for assignments if you do not attend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

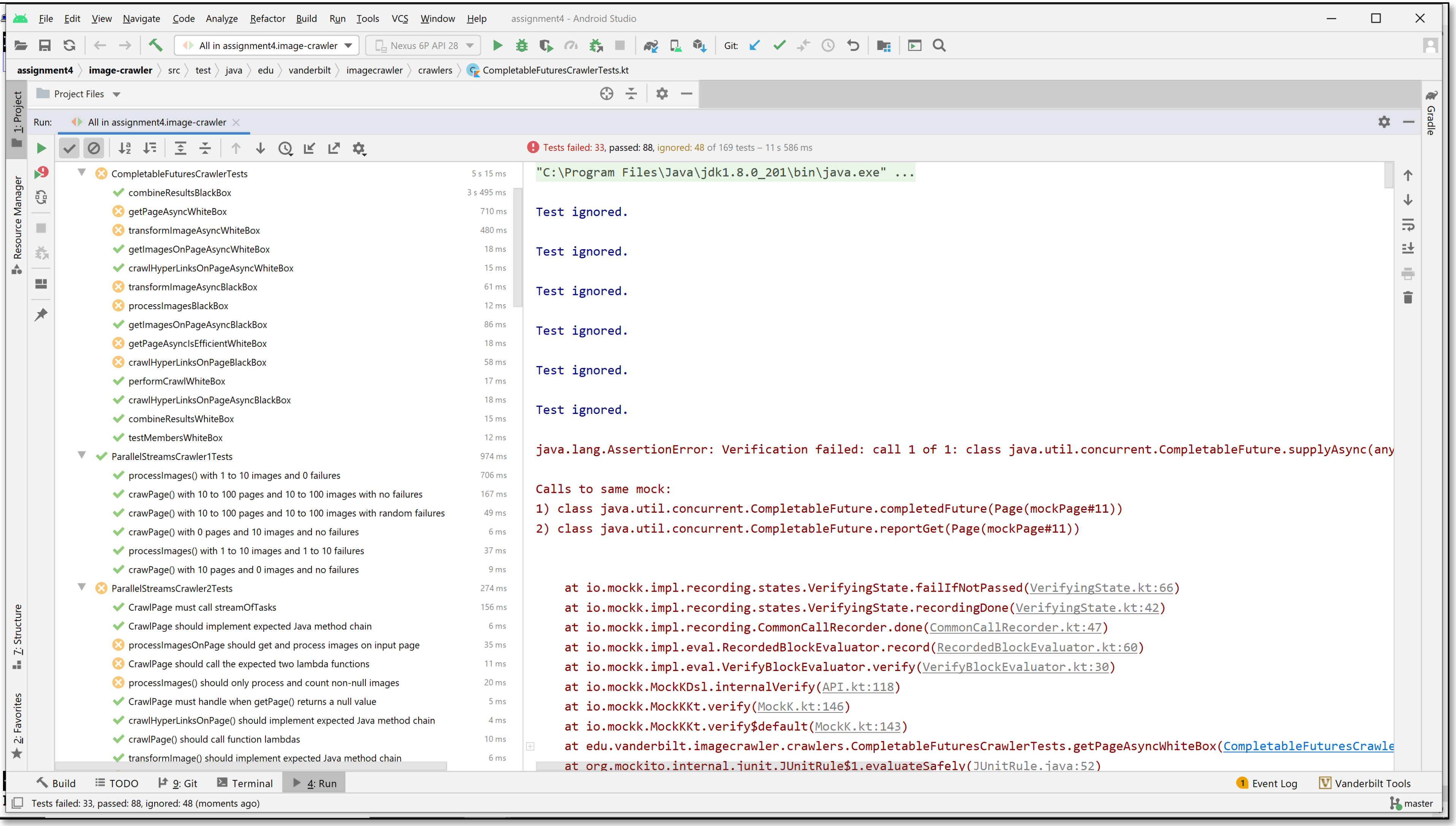
- Assignment assessments will be done via reviews by course staff
  - Assignments *must* be submitted on time or you'll get a 0
  - Your initial submission must compile & be largely complete or you won't get a review or a final grade
  - You will not receive a grade for assignments if you do not attend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


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

See cdn.vanderbilt.edu/vu-URL/wp-content/uploads/sites/241/2023/08/16143452/Vanderbilt-University-Academic-Affairs-Guidance-for-Artificial-Intelligence.pdf

# Overview of Assignments & Assessments

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


File
Edit
View
Navigate
Code
Analyze
Refactor
Build
Run
Tools
VCS
Window
Help
assignment4 - Android Studio
All in assignment4.image-crawler
Nexus 6P API 28
Git:
assignment4 〉 image-crawler 〉 src 〉 test 〉 java 〉 edu 〉 vanderbilt 〉 imagecrawler 〉 crawlers 〉 CompletableFuturesCrawlerTests.kt
1: Project
Resource Manager
Project Files
Run:
All in assignment4.image-crawler
Gradle
7: Structure
2: Favorites
Build
TODO
9: Git
Terminal
4: Run
Event Log
Vanderbilt Tools
master

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

```
CompletableFuturesCrawlerTests                                              5 s 15 ms
    combineResultsBlackBox                                                  3 s 495 ms
    getPageAsyncWhiteBox                                                    710 ms
    transformImageAsyncWhiteBox                                             480 ms
    getImagesOnPageAsyncWhiteBox                                            18 ms
    crawlHyperLinksOnPageAsyncWhiteBox                                      15 ms
    transformImageAsyncBlackBox                                             61 ms
    processImagesBlackBox                                                   12 ms
    getImagesOnPageAsyncBlackBox                                            86 ms
    getPageAsyncIsEfficientWhiteBox                                         18 ms
    crawlHyperLinksOnPageBlackBox                                           58 ms
    performCrawlWhiteBox                                                    17 ms
    crawlHyperLinksOnPageAsyncBlackBox                                      18 ms
    combineResultsWhiteBox                                                  15 ms
    testMembersWhiteBox                                                     12 ms
ParallelStreamsCrawler1Tests                                                974 ms
    processImages() with 1 to 10 images and 0 failures                      706 ms
    crawPage() with 10 to 100 pages and 10 to 100 images with no failures   167 ms
    crawPage() with 10 to 100 pages and 10 to 100 images with random failures 49 ms
    crawPage() with 0 pages and 10 images and no failures                   6 ms
    processImages() with 1 to 10 images and 1 to 10 failures                37 ms
    crawPage() with 10 pages and 0 images and no failures                   9 ms
ParallelStreamsCrawler2Tests                                                274 ms
    CrawlPage must call streamOfTasks                                       156 ms
    CrawlPage should implement expected Java method chain                   6 ms
    processImagesOnPage should get and process images on input page         35 ms
    CrawlPage should call the expected two lambda functions                 11 ms
    processImages() should only process and count non-null images           20 ms
    CrawlPage must handle when getPage() returns a null value               5 ms
    crawlHyperLinksOnPage() should implement expected Java method chain     4 ms
    crawlPage() should call function lambdas                                10 ms
    transformImage() should implement expected Java method chain            6 ms
```

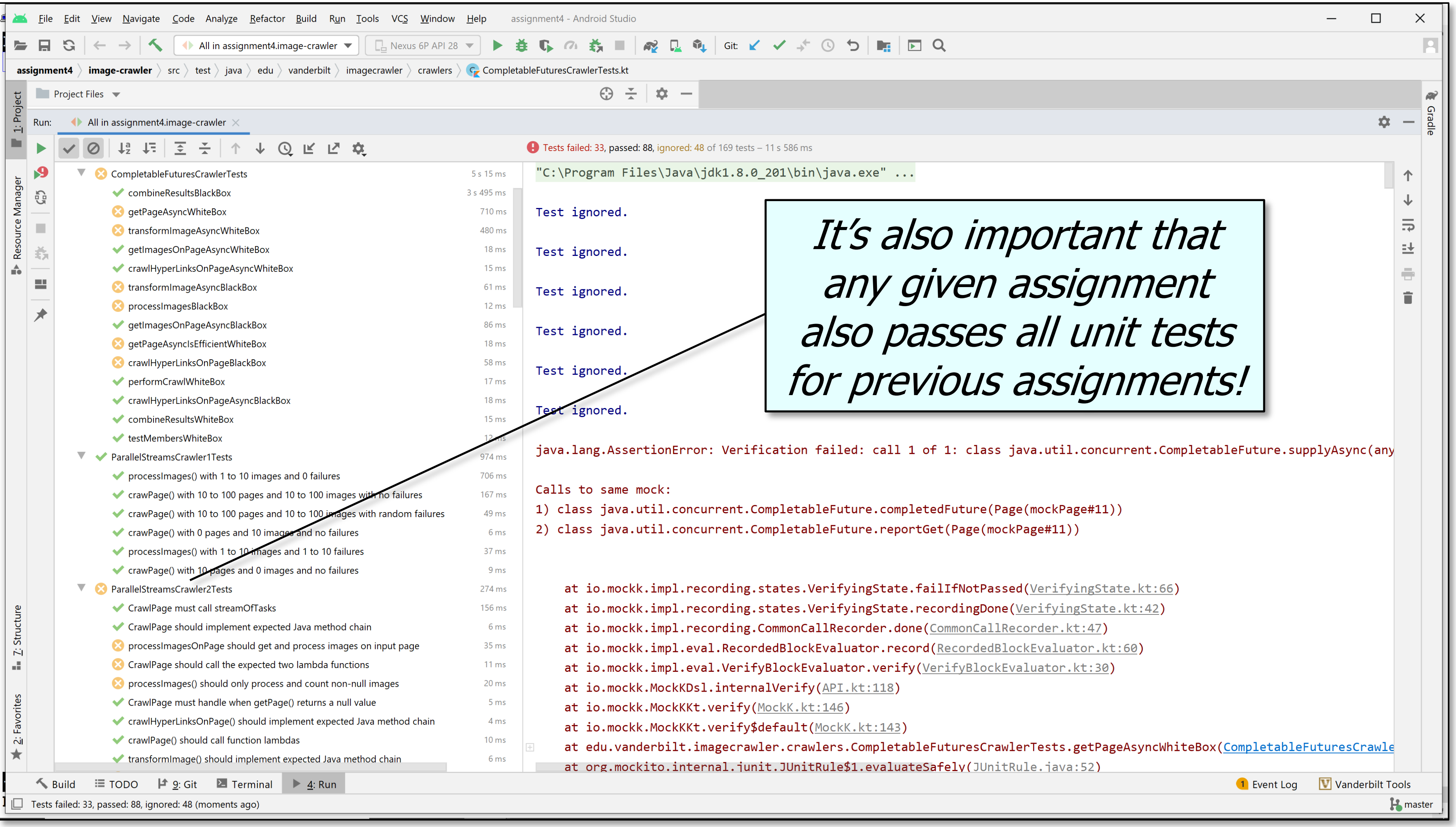
```
"C:\Program Files\Java\jdk1.8.0_201\bin\java.exe" ...

Test ignored.

Test ignored.

Test ignored.

Test ignored.

Test ignored.

Test ignored.

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

Calls to same mock:
1) class java.util.concurrent.CompletableFuture.completedFuture(Page(mockPage#11))
2) class java.util.concurrent.CompletableFuture.reportGet(Page(mockPage#11))


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

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

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

# Overview of Assignments & Assessments

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


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
assignment4 - Android Studio
All in assignment4.image-crawler
Nexus 6P API 28
Git:
assignment4 〉 image-crawler 〉 src 〉 test 〉 java 〉 edu 〉 vanderbilt 〉 imagecrawler 〉 crawlers 〉 CompletableFuturesCrawlerTests.kt
Project Files
Run: All in assignment4.image-crawler
1: Project
Resource Manager
Z: Structure
2: Favorites
Gradle
Build
TODO
9: Git
Terminal
4: Run
Event Log
Vanderbilt Tools
master

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

```
CompletableFuturesCrawlerTests                                           5 s 15 ms
    combineResultsBlackBox                                               3 s 495 ms
    getPageAsyncWhiteBox                                                 710 ms
    transformImageAsyncWhiteBox                                          480 ms
    getImagesOnPageAsyncWhiteBox                                         18 ms
    crawlHyperLinksOnPageAsyncWhiteBox                                   15 ms
    transformImageAsyncBlackBox                                          61 ms
    processImagesBlackBox                                                12 ms
    getImagesOnPageAsyncBlackBox                                         86 ms
    getPageAsyncIsEfficientWhiteBox                                      18 ms
    crawlHyperLinksOnPageBlackBox                                        58 ms
    performCrawlWhiteBox                                                 17 ms
    crawlHyperLinksOnPageAsyncBlackBox                                   18 ms
    combineResultsWhiteBox                                               15 ms
    testMembersWhiteBox                                                  12 ms
ParallelStreamsCrawler1Tests                                             974 ms
    processImages() with 1 to 10 images and 0 failures                   706 ms
    crawPage() with 10 to 100 pages and 10 to 100 images with no failures    167 ms
    crawPage() with 10 to 100 pages and 10 to 100 images with random failures  49 ms
    crawPage() with 0 pages and 10 images and no failures                6 ms
    processImages() with 1 to 10 images and 1 to 10 failures             37 ms
    crawPage() with 10 pages and 0 images and no failures                9 ms
ParallelStreamsCrawler2Tests                                             274 ms
    CrawlPage must call streamOfTasks                                    156 ms
    CrawlPage should implement expected Java method chain                6 ms
    processImagesOnPage should get and process images on input page      35 ms
    CrawlPage should call the expected two lambda functions              11 ms
    processImages() should only process and count non-null images        20 ms
    CrawlPage must handle when getPage() returns a null value            5 ms
    crawlHyperLinksOnPage() should implement expected Java method chain  4 ms
    crawlPage() should call function lambdas                             10 ms
    transformImage() should implement expected Java method chain         6 ms
```

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

Test ignored.

Test ignored.

Test ignored.

Test ignored.

Test ignored.

Test ignored.

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

Calls to same mock:
1) class java.util.concurrent.CompletableFuture.completedFuture(Page(mockPage#11))
2) class java.util.concurrent.CompletableFuture.reportGet(Page(mockPage#11))


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

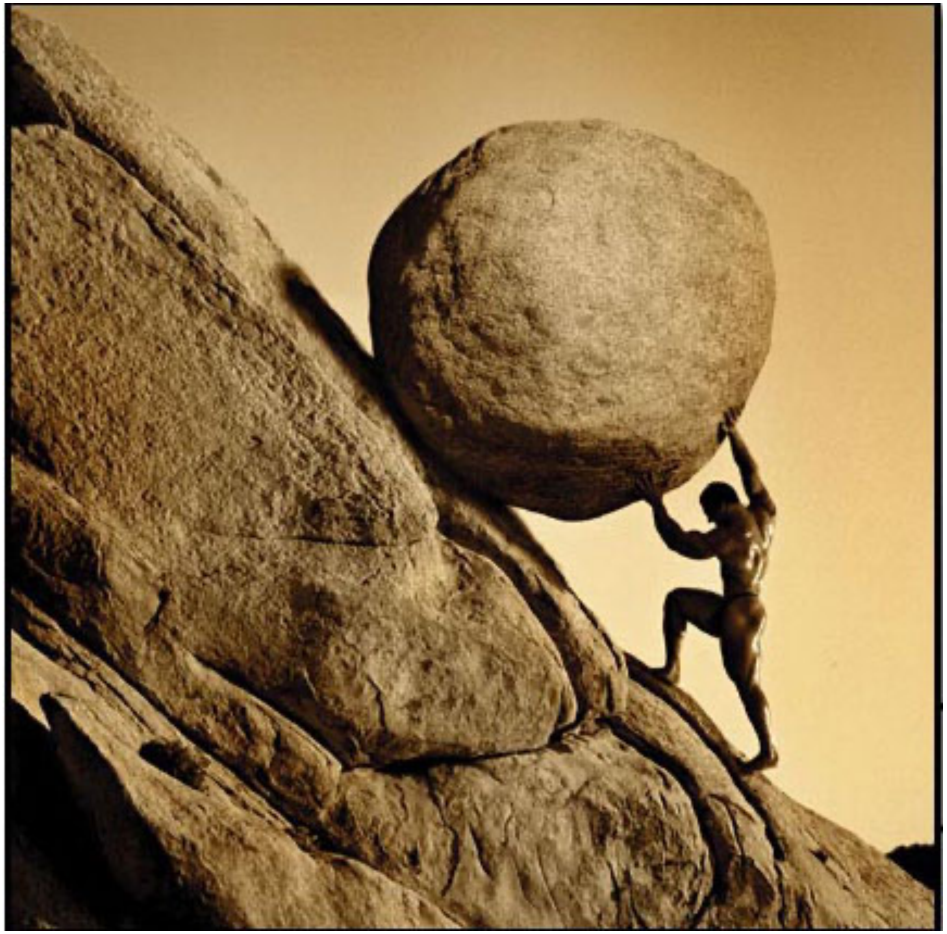
Tests failed: 33, passed: 88, ignored: 48 (moments ago)

*It's also important that any given assignment also passes all unit tests for previous assignments!*

See item #16 at github.com/douglascraigschmidt/CS253/wiki/CS-253-FAQ

# Overview of Assignments & Assessments

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



These weightings may change, depending on various factors

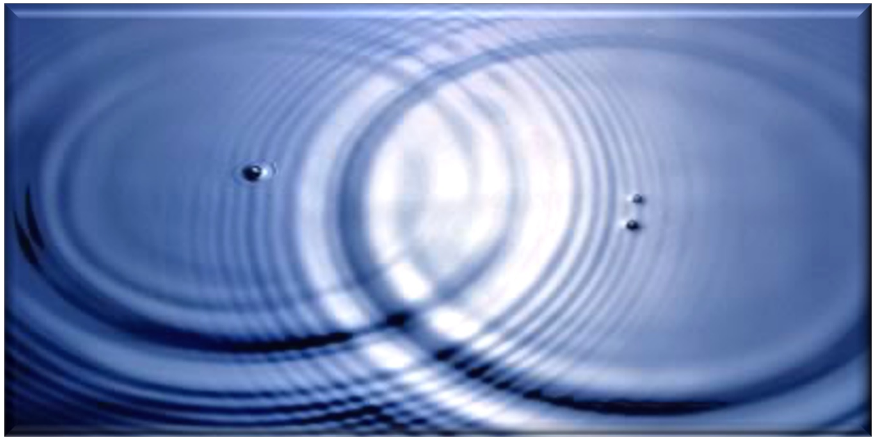
# Overview of Assignments & Assessments

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

# Overview of Assignments & Assessments

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

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



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

# Overview of Assignments & Assessments

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

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

# CS 253: Parallel Functional Programming with Java & Android: Overview (Part 2)