

CS 5254: Concurrent Object-Oriented & Functional Programming

Course Overview (Part 2)

Douglas C. Schmidt

d.schmidt@vanderbilt.edu

www.dre.vanderbilt.edu/~schmidt



Professor of Computer Science

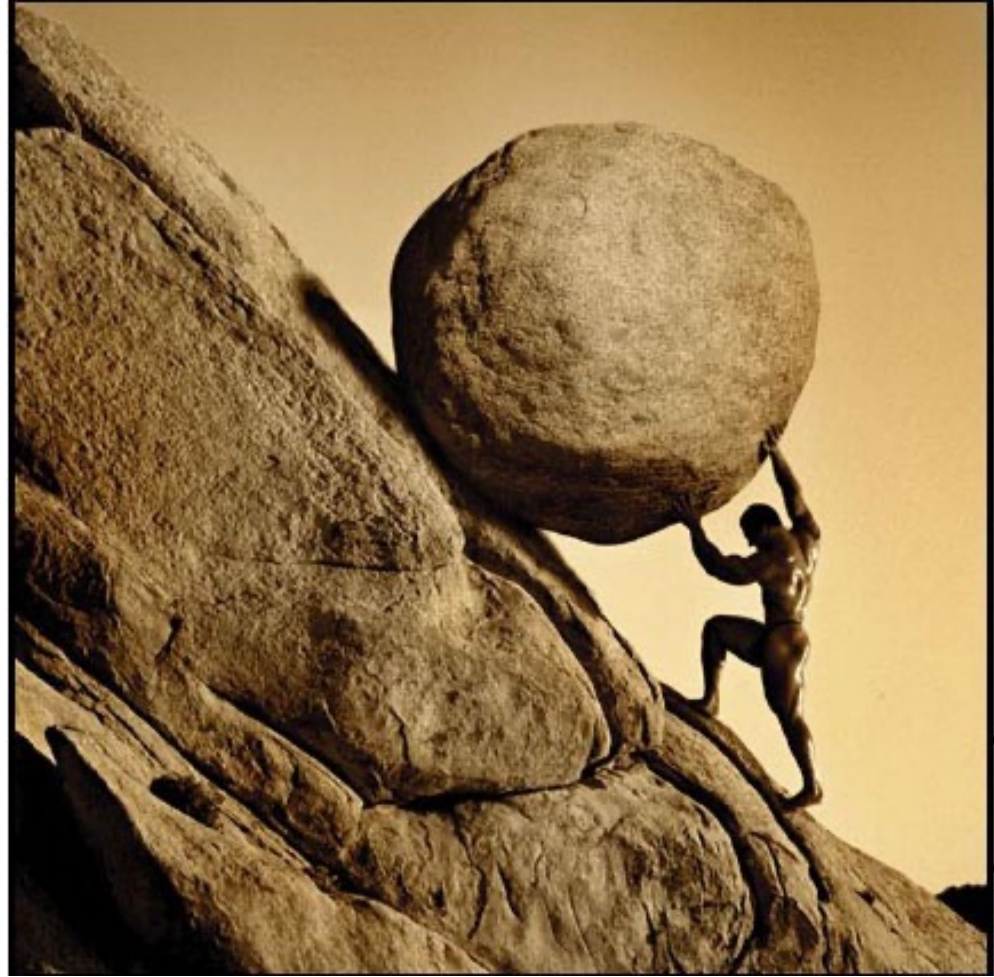
**Institute for Software
Integrated Systems**

**Vanderbilt University
Nashville, Tennessee, USA**



Learning Objectives in this Lesson

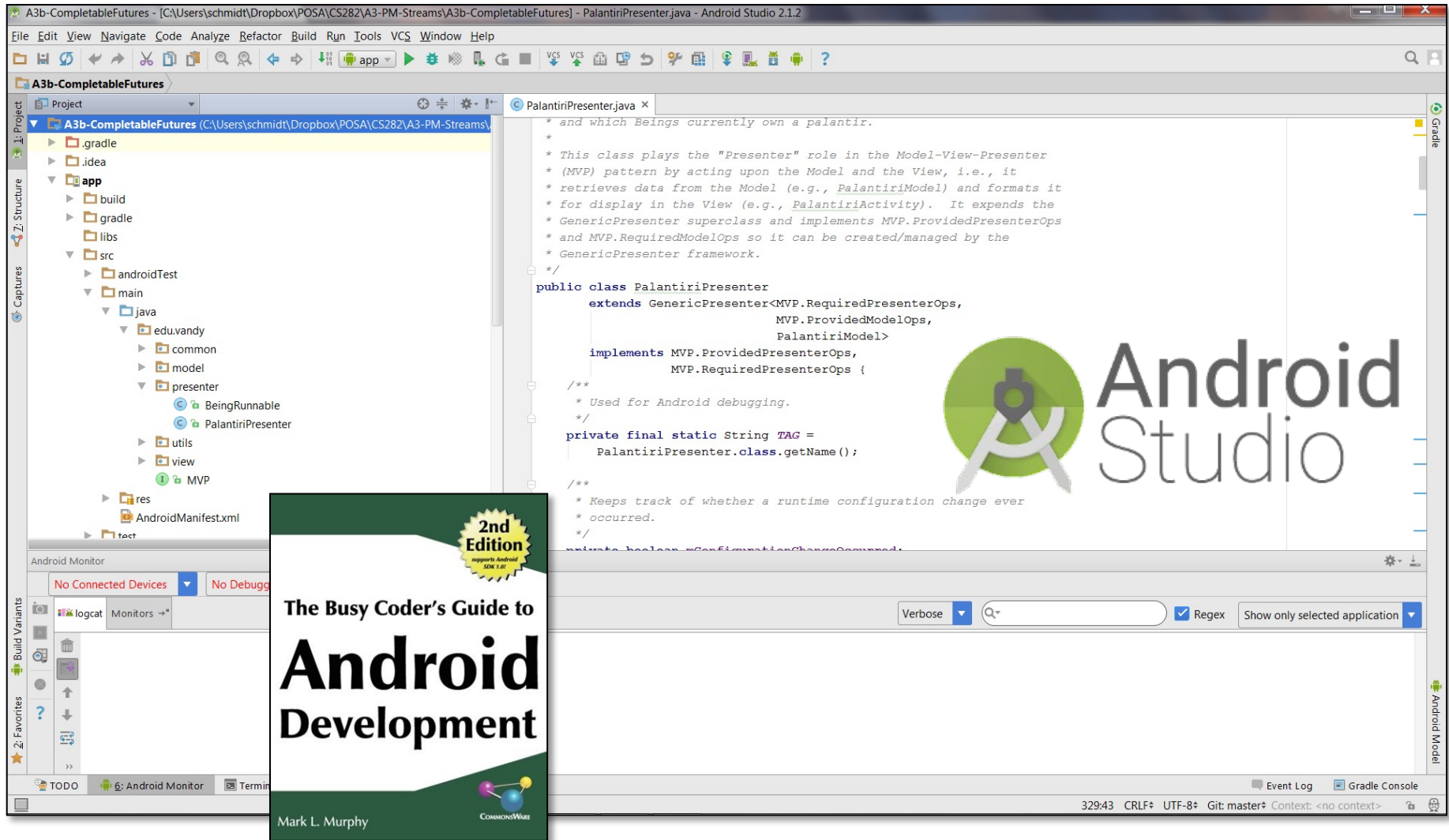
- Understand the course topics & logistics
 - Course philosophy
 - Course contents
 - Structure of the lecture material
- Overview of the assignments & assessments



Overview of the Assignments & Assessments

Overview of Assignments & Assessments

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



```

* and which Beings currently own a palantiri.
*
* 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 extends 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;

```

2nd Edition
supports Android SDK 1.0!

The Busy Coder's Guide to Android Development

Mark L. Murphy

COMMONWAVE

You can use any IDE, but your final submission *must* build/run with the latest Android Studio & Android 13 "T" (API level 33)

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 13 "T" (API level 33)



See en.wikipedia.org/wiki/Android_13

Overview of Assignments & Assessments

- All source code for assignments & examples available at GitHub

The screenshot shows a web browser window displaying the GitHub repository page for `douglascraigschmidt/CS5254`. The page is public and contains source code, issues, pull requests, actions, projects, wiki, security, and insights. The repository is currently on the `master` branch, has 1 branch, and 0 tags. The latest commit is `eaf98cc` by `douglascraigschmidt`, made 5 hours ago, with 1 commit. A folder named `assignment1a` is visible, updated 5 hours ago. The repository has 0 stars, 1 watching, and 0 forks. There are no releases published.

GitHub - douglascraigschmidt / CS5254 (Public)

Code Issues Pull requests Actions Projects Wiki Security Insights

master 1 branch 0 tags

Go to file Code

douglascraigschmidt updates eaf98cc 5 hours ago 1 commit

assignment1a updates 5 hours ago

About

Contains examples and assignments for my CS 5254 course at Vanderbilt University

0 stars

1 watching

0 forks

Releases

No releases published

Go to GitHub at github.com/douglascraigschmidt/CS5254

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 dark purple. 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 white. Below this is a paragraph of text in white: "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 that is another paragraph: "Do you want more from your GitLab installation? A subscription bundles the Enterprise Edition with support from the GitLab team. The Enterprise Edition allows you to sync LDAP groups, control pushes via git hooks, integrate better with Jenkins and Jira, and to run MySQL and forward logs when using our Omnibus package. Our service engineers will help you keep your server running smoothly." At the bottom are two buttons: a light blue button with the text "GitLab Community Edition" and a green button with the text "Get a subscription".

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 multiple times
 - i.e., you need to understand Git!



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



Overview of Assignments & Assessments

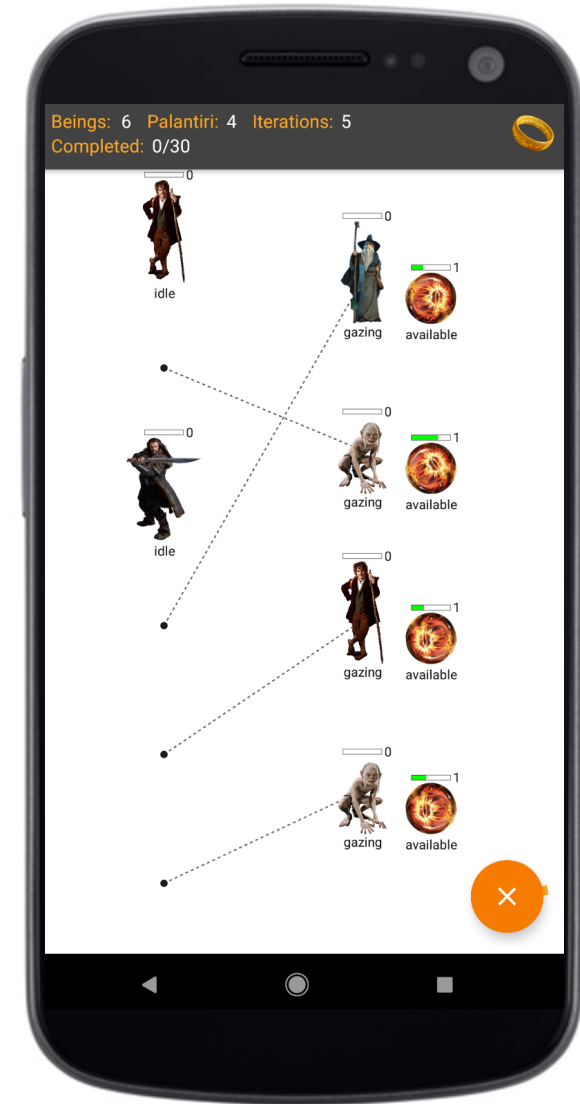
- Assignments will provide a range of experience with Java 11 & Android concurrent programs



See github.com/douglasraigschmidt/CS5254

Overview of Assignments & Assessments

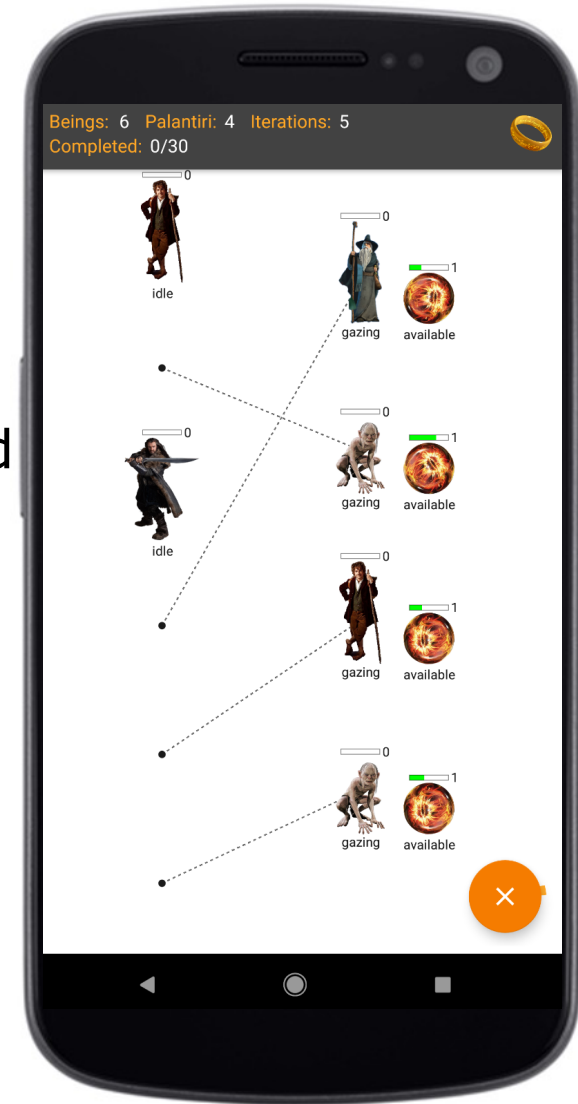
- In particular, you'll implement multiple variants of a Java concurrent resource manager & an associated Android app



See en.wikipedia.org/wiki/Palantir

Overview of Assignments & Assessments

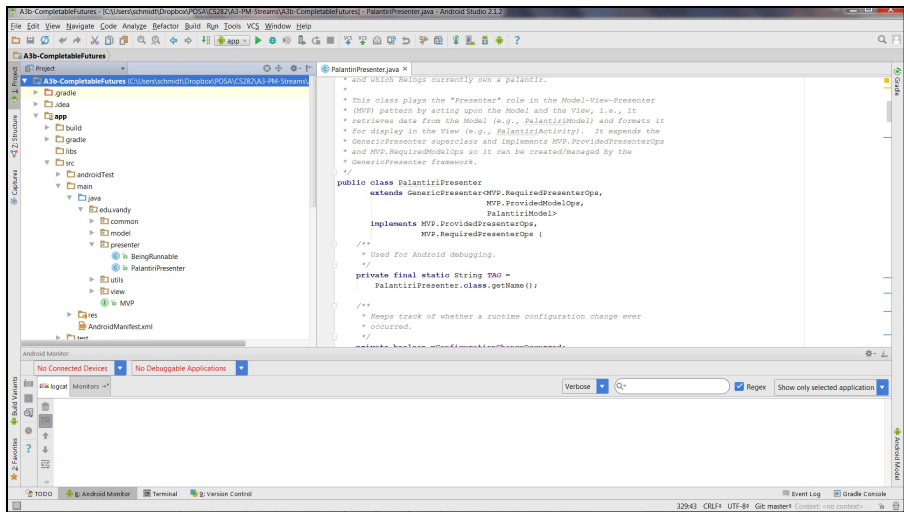
- In particular, you'll implement multiple variants of a Java concurrent resource manager & an associated Android app, e.g.
 - Java Thread & Runnable
 - Java Executor framework (e.g., ExecutorService, ExecutorCompletionService, & ForkJoinPool)
 - Java Semaphore, built-in monitor objects, Stamped Lock,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



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

Overview of Assignments & Assessments

- Assignment assessments will be done via reviews by course staff



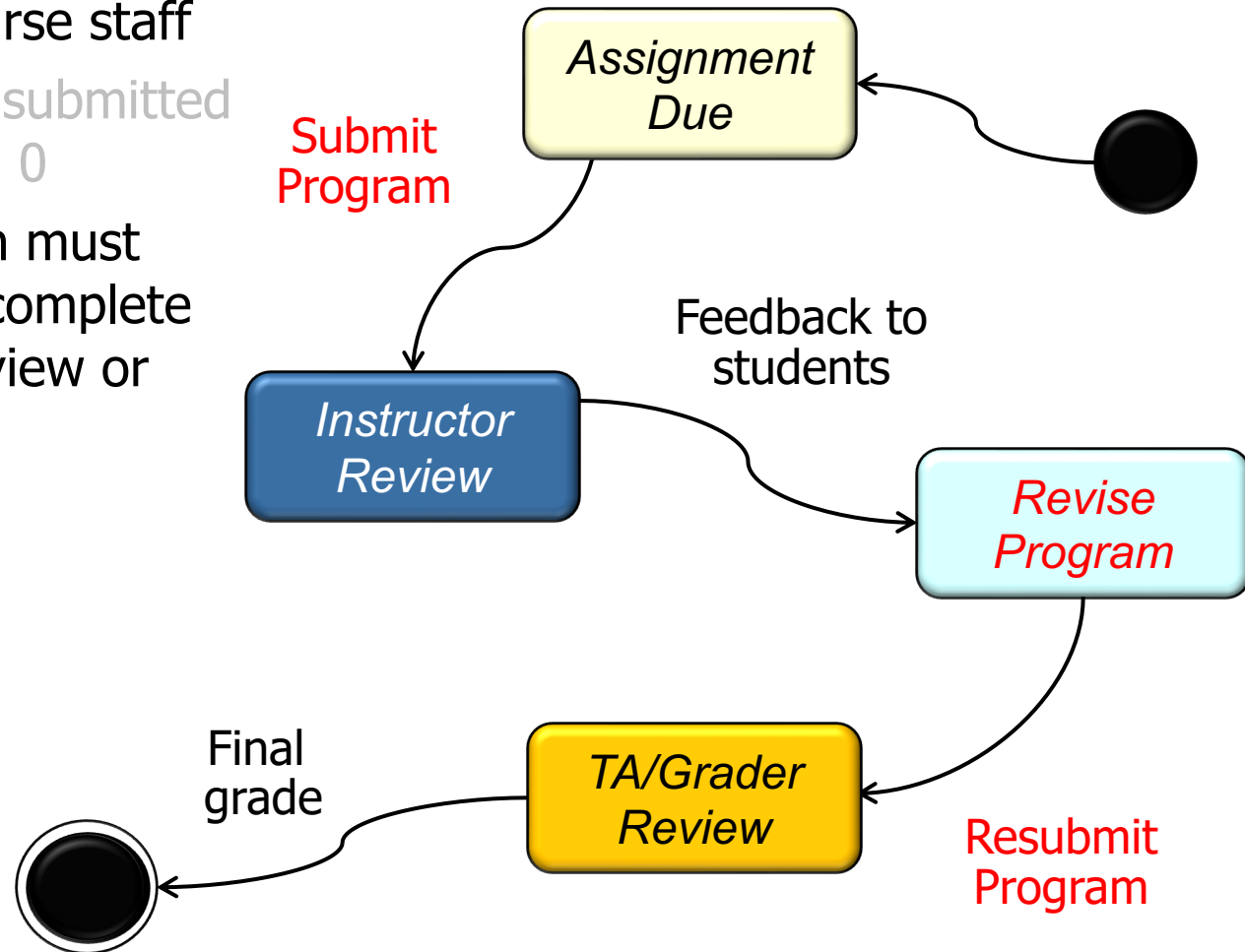
Overview of Assignments & Assessments

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



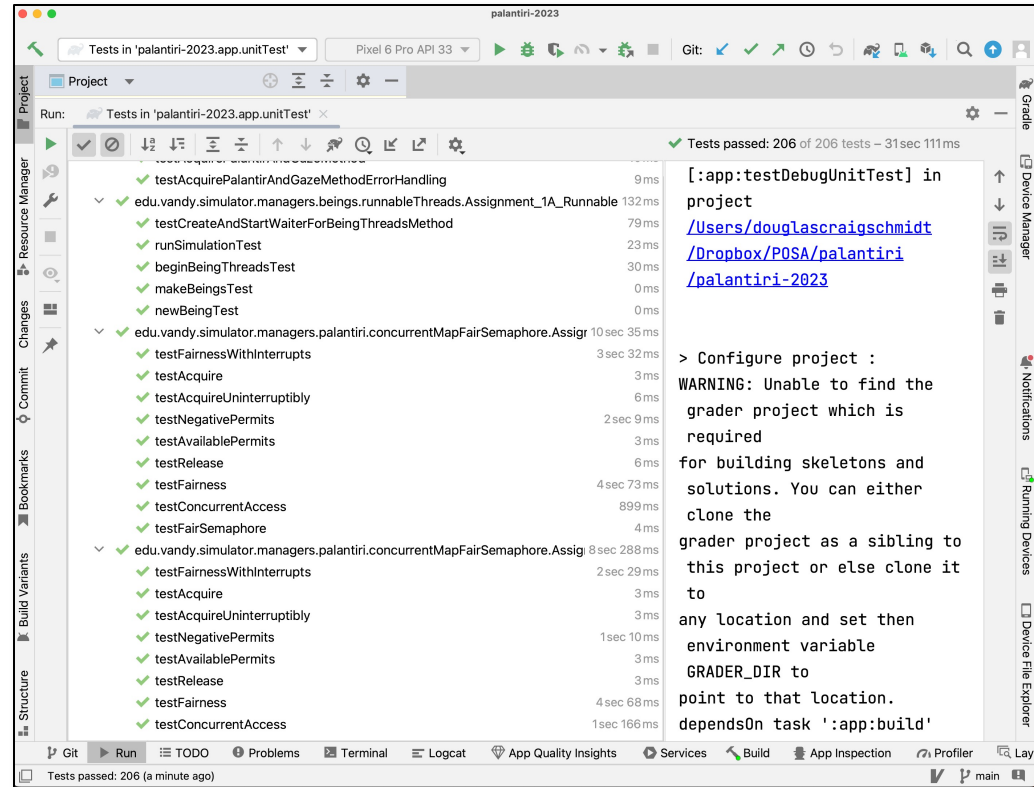
Overview of Assignments & Assessments

- 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

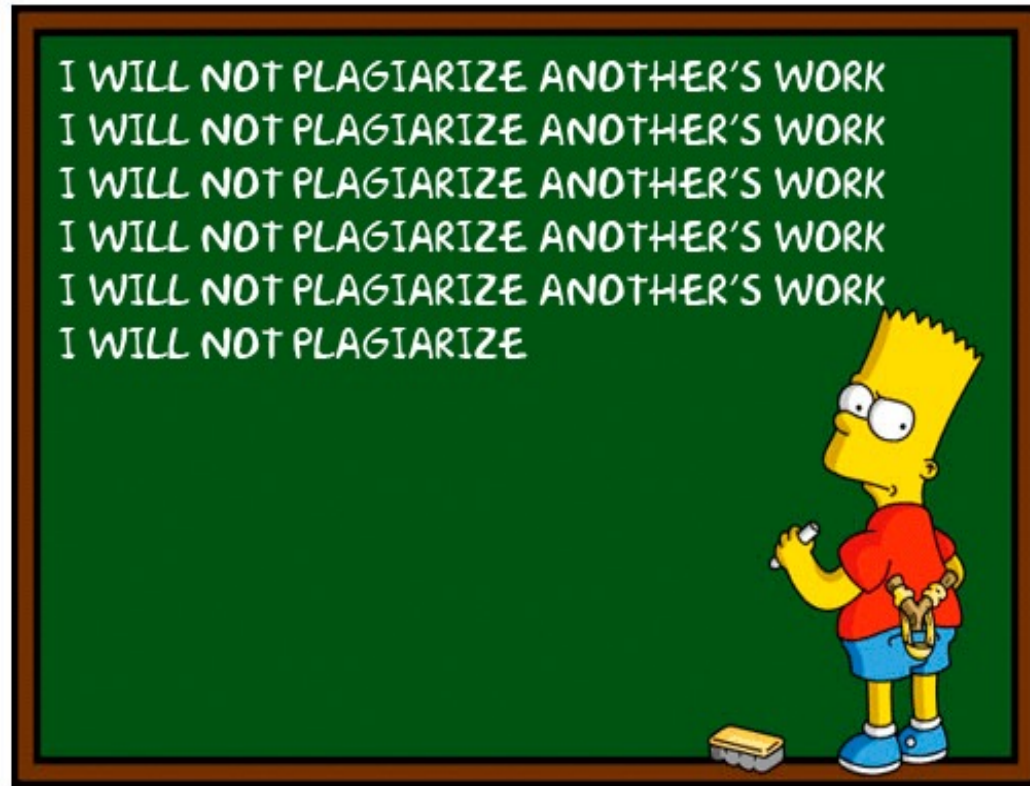
- Assignment assessments will be done via reviews by course staff
 - Assignments *must* be submitted on time or you'll get a 0
- Your initial submission must compile & be largely complete or you won't get a review or a final grade
 - You *must* also run the regression tests & push a screenshot of the results to GitLab



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

Overview of Assignments & Assessments

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



Overview of Assignments & Assessments

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

The screenshot displays the Android Studio IDE with the following components:

- Menu Bar:** File, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, VCS, Window, Help.
- Toolbar:** Navigation and development tools.
- Project Files:** assignment4 > image-crawler > src > test > java > edu > vanderbilt > imagecrawler > crawlers > ComposableFuturesCrawlerTests.kt
- Run:** All in assignment4.image-crawler
- Test Results Tree:**
 - ComposableFuturesCrawlerTests (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)
 - crawlPage() with 10 to 100 pages and 10 to 100 images with no failures (167 ms)
 - crawlPage() with 10 to 100 pages and 10 to 100 images with random failures (49 ms)
 - crawlPage() with 0 pages and 10 images and no failures (6 ms)
 - processImages() with 1 to 10 images and 1 to 10 failures (37 ms)
 - crawlPage() 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)

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

Log Output:

```
"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.MockKDs1.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.ComposableFuturesCrawlerTests.getPageAsyncWhiteBox(ComposableFuturesCrawle  
at org.mockito.internal.junit.JUnitRule$1.evaluateSafely(JUnitRule.java:52)
```

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

Overview of Assignments & Assessments

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

The screenshot shows the IDE interface with the following details:

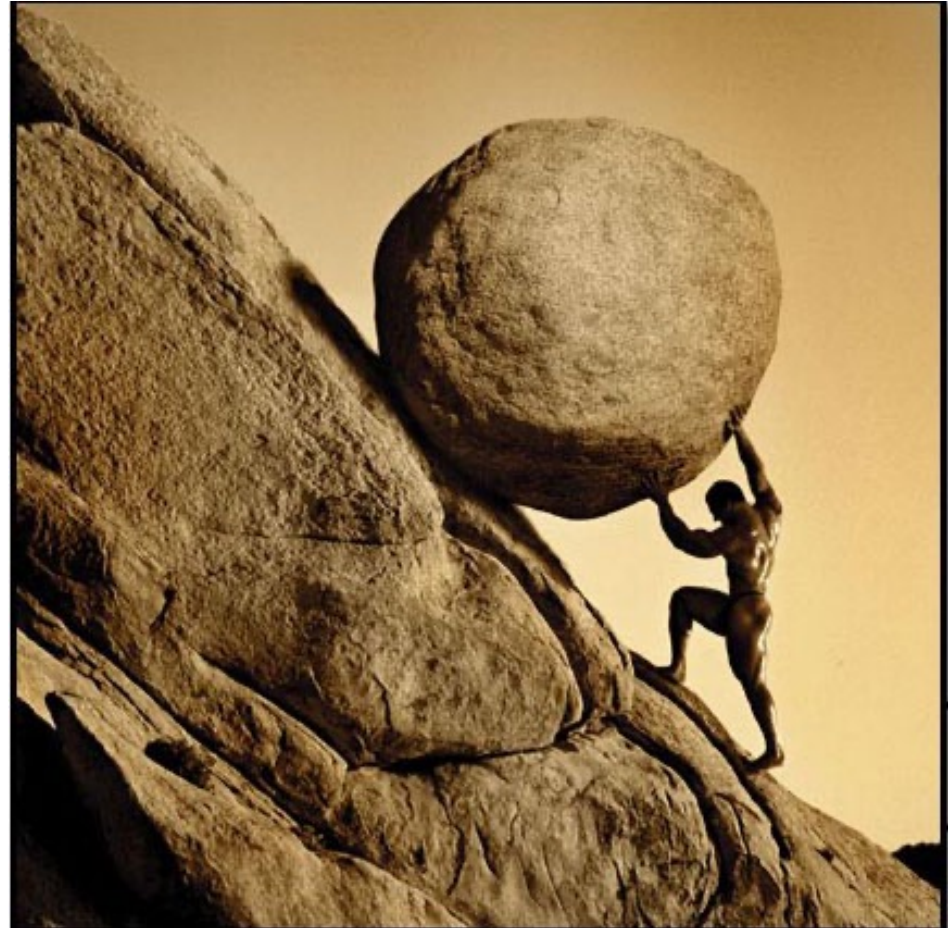
- Run Panel:** Shows test results for 'CompletableFutureCrawlerTests'. The summary indicates 33 tests failed, 88 passed, and 48 were ignored out of 169 tests, taking 11 seconds and 586 milliseconds.
- Test Results Table:**

Test Name	Duration	Status
CompletableFutureCrawlerTests	5 s 15 ms	Failed
combineResultsBlackBox	3 s 495 ms	Passed
getPageAsyncWhiteBox	710 ms	Failed
transformImageAsyncWhiteBox	480 ms	Failed
getImagesOnPageAsyncWhiteBox	18 ms	Passed
crawlHyperLinksOnPageAsyncWhiteBox	15 ms	Passed
transformImageAsyncBlackBox	61 ms	Failed
processImagesBlackBox	12 ms	Passed
getImagesOnPageAsyncBlackBox	86 ms	Passed
getPageAsyncIsEfficientWhiteBox	18 ms	Failed
crawlHyperLinksOnPageBlackBox	58 ms	Failed
performCrawlWhiteBox	17 ms	Passed
crawlHyperLinksOnPageAsyncBlackBox	18 ms	Passed
combineResultsWhiteBox	15 ms	Passed
testMembersWhiteBox	12 ms	Passed
ParallelStreamsCrawler1Tests	974 ms	Passed
processImages() with 1 to 10 images and 0 failures	706 ms	Passed
crawlPage() with 10 to 100 pages and 10 to 100 images with no failures	167 ms	Passed
crawlPage() with 10 to 100 pages and 10 to 100 images with random failures	49 ms	Passed
crawlPage() with 0 pages and 10 images and no failures	6 ms	Passed
processImages() with 1 to 10 images and 1 to 10 failures	37 ms	Passed
crawlPage() with 10 pages and 0 images and no failures	9 ms	Passed
ParallelStreamsCrawler2Tests	274 ms	Failed
CrawlPage must call streamOfTasks	156 ms	Passed
CrawlPage should implement expected Java method chain	6 ms	Passed
processImagesOnPage should get and process images on input page	35 ms	Failed
CrawlPage should call the expected two lambda functions	11 ms	Failed
processImages() should only process and count non-null images	20 ms	Failed
CrawlPage must handle when getPage() returns a null value	5 ms	Passed
crawlHyperLinksOnPage() should implement expected Java method chain	4 ms	Passed
crawlPage() should call function lambdas	10 ms	Passed
transformImage() should implement expected Java method chain	6 ms	Passed
- Callout Box:** A light blue box with a black border contains the text: *It's important that your current assignment also passes all the tests for previous assignments!*
- Terminal Output:** Shows a `java.lang.AssertionError: Verification failed: call 1 of 1: class java.util.concurrent.CompletableFuture.supplyAsync(any)` and a list of mock calls:
 - 1) class java.util.concurrent.CompletableFuture.completedFuture(Page(mockPage#11))
 - 2) class java.util.concurrent.CompletableFuture.reportGet(Page(mockPage#11))

See item #19 at github.com/douglasraigschmidt/CS5254/wiki/CS-5254-FAQ

Overview of Assignments & Assessments

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

IMPORTANT



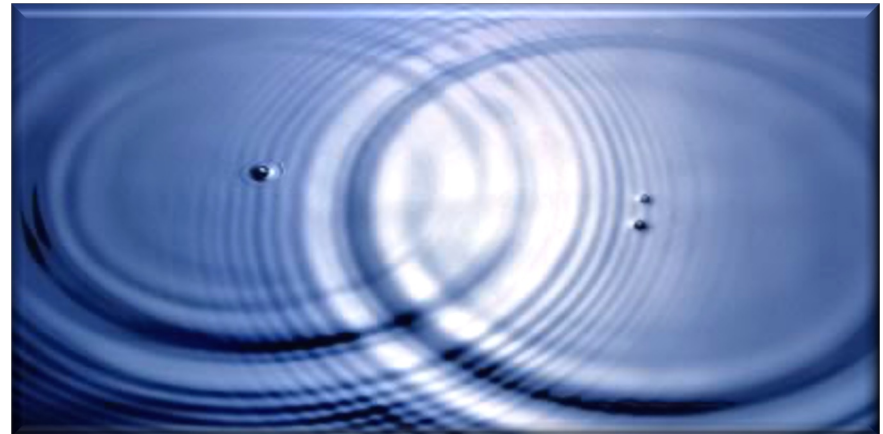
Overview of Assignments & Assessments

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

IMPORTANT



Attendance also affects other aspects of your quiz & assignment grades



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

Overview of Assignments & Assessments

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

IMPORTANT



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

CS 5254: Concurrent Object- Oriented & Functional Programming: Overview (Part 2)