IntelliJ, Maven, JUnit Submitting Assignments

CSC416/565: Design and Implementation of Compilers

WCU
WEST CHESTER UNIVERSITY

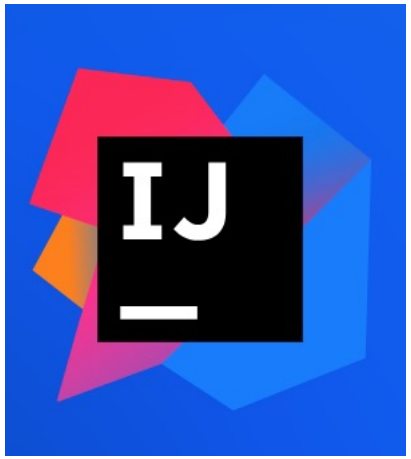
Utilize our CSC416/565 discord for your questions!

Installing IntelliJ

https://www.jetbrains.com/idea/download/

Platforms and Versions

- IntelliJ can be installed on:
  - Windows
  - Mac
  - Linux



- Versions:
  - Ultimate – free for students
  - Community Edition (CE) – open source, and free for all, **this is what I have installed**
    - Scroll down on the download page to find CE

Experimenting w/ IntelliJ and Maven

IntelliJ

- IntelliJ is just like any other full-featured IDE (Eclipse, NetBeans, Visual Studio, Visual Studio Code w/ plugins), but there is a little bit of a learning curve as to where things are
  - *The more you use a new IDE, the more familiar you will be!*
  - There's no short cut here except to spend time playing around.
- We'll also be utilizing Maven project management and JUnit in our projects

https://maven.apache.org/guides/getting-started/maven-in-five-minutes.html

Maven

- Maven helps control the lifecycles of Java projects, and standardizes where source files, compiled files, libraries, etc., are all kept

Maven – `pom.xml` configuration file

- A single configuration file that contains project properties and dependencies
- I'll be writing this file for you. You won't have to modify it. But feel free to look at it to learn more about Maven!

Name of project

Which java version is required?

This project uses Junit 4.12 which is also needed.

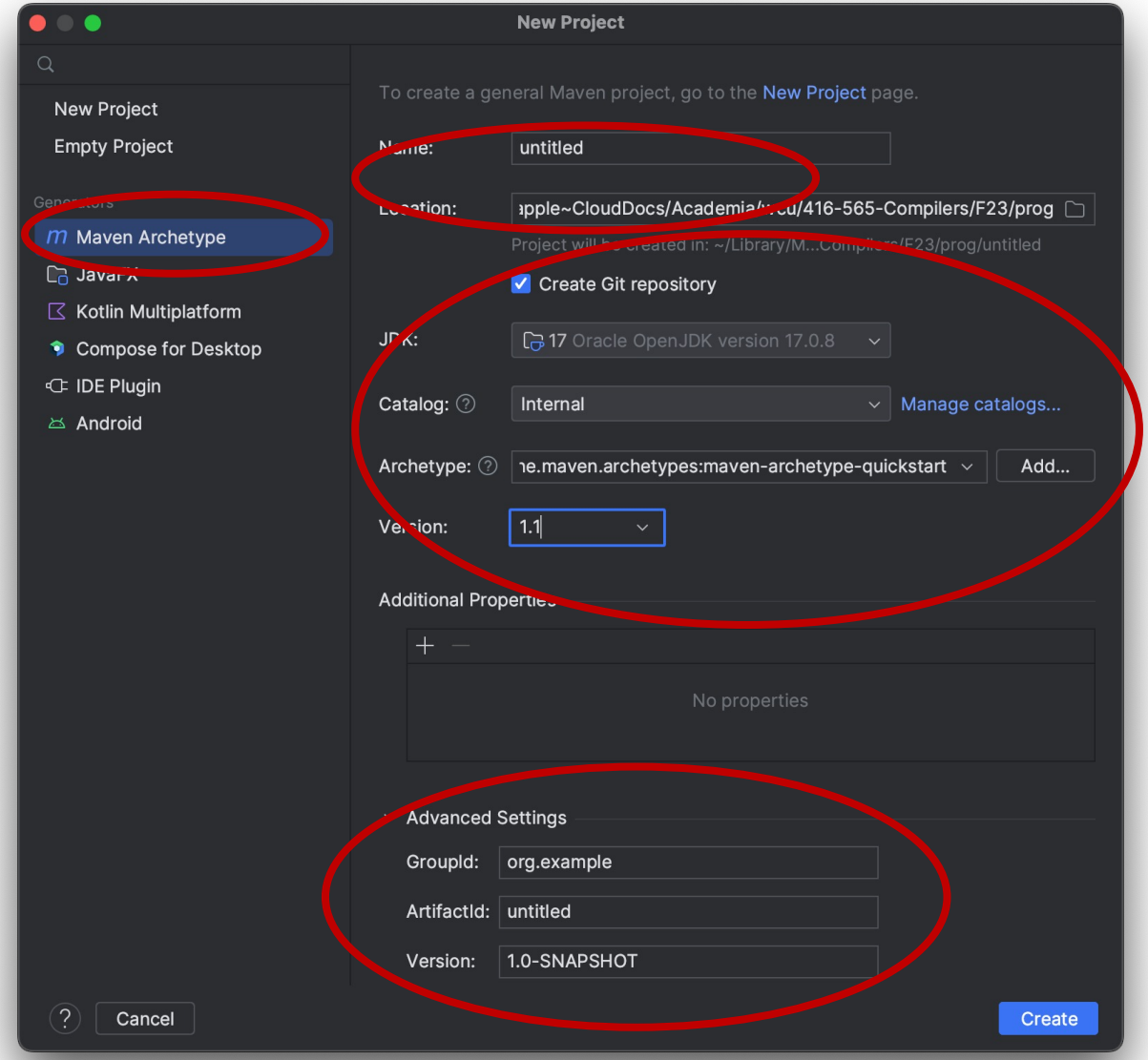
```xml
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>com.mycompany.app</groupId>
  <artifactId>my-app</artifactId>
  <version>1.0-SNAPSHOT</version>

  <properties>
    <maven.compiler.source>1.7</maven.compiler.source>
    <maven.compiler.target>1.7</maven.compiler.target>
  </properties>

  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.12</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
</project>
```

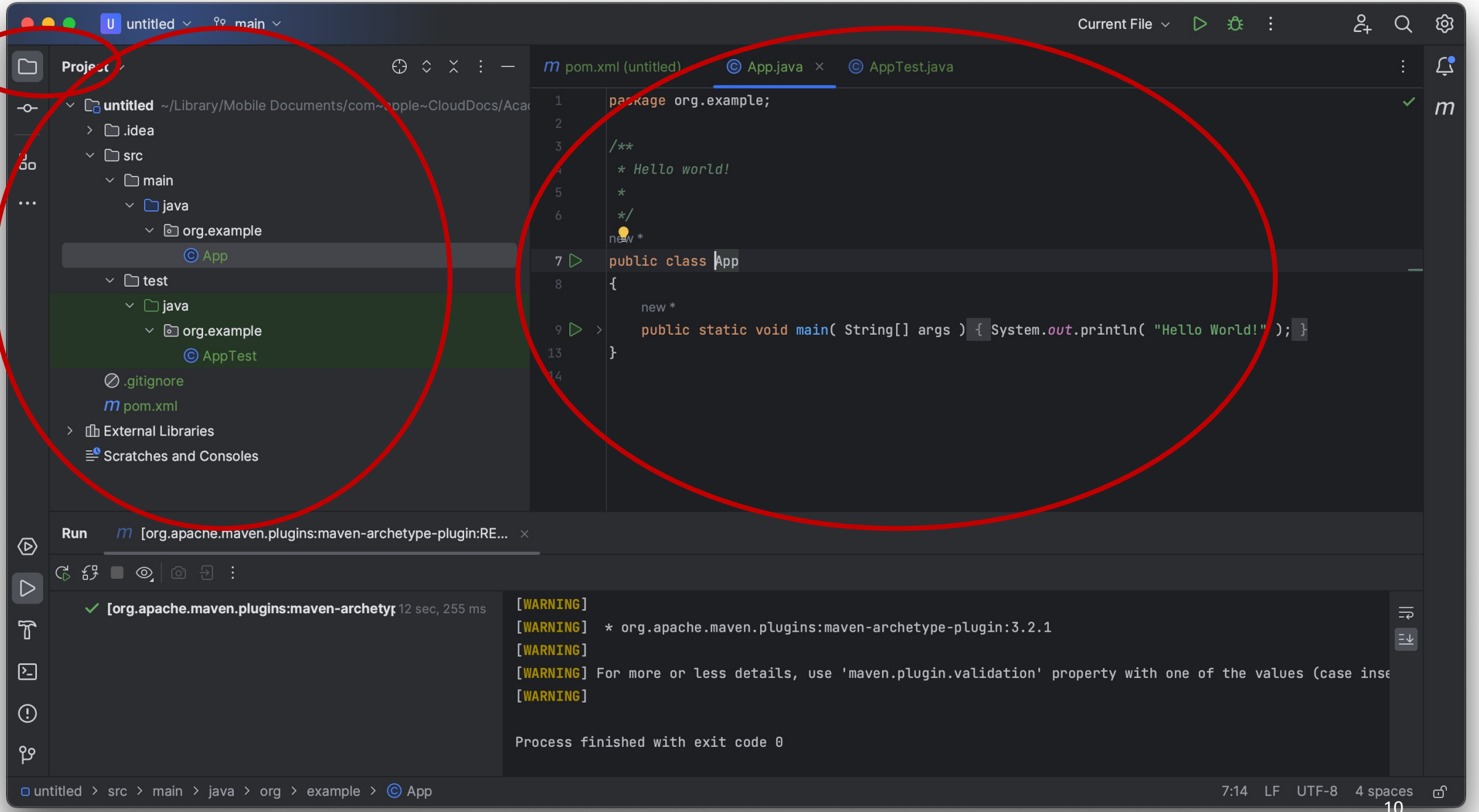
Maven – Standard Project Structure

Under this directory you will notice the following standard project structure.

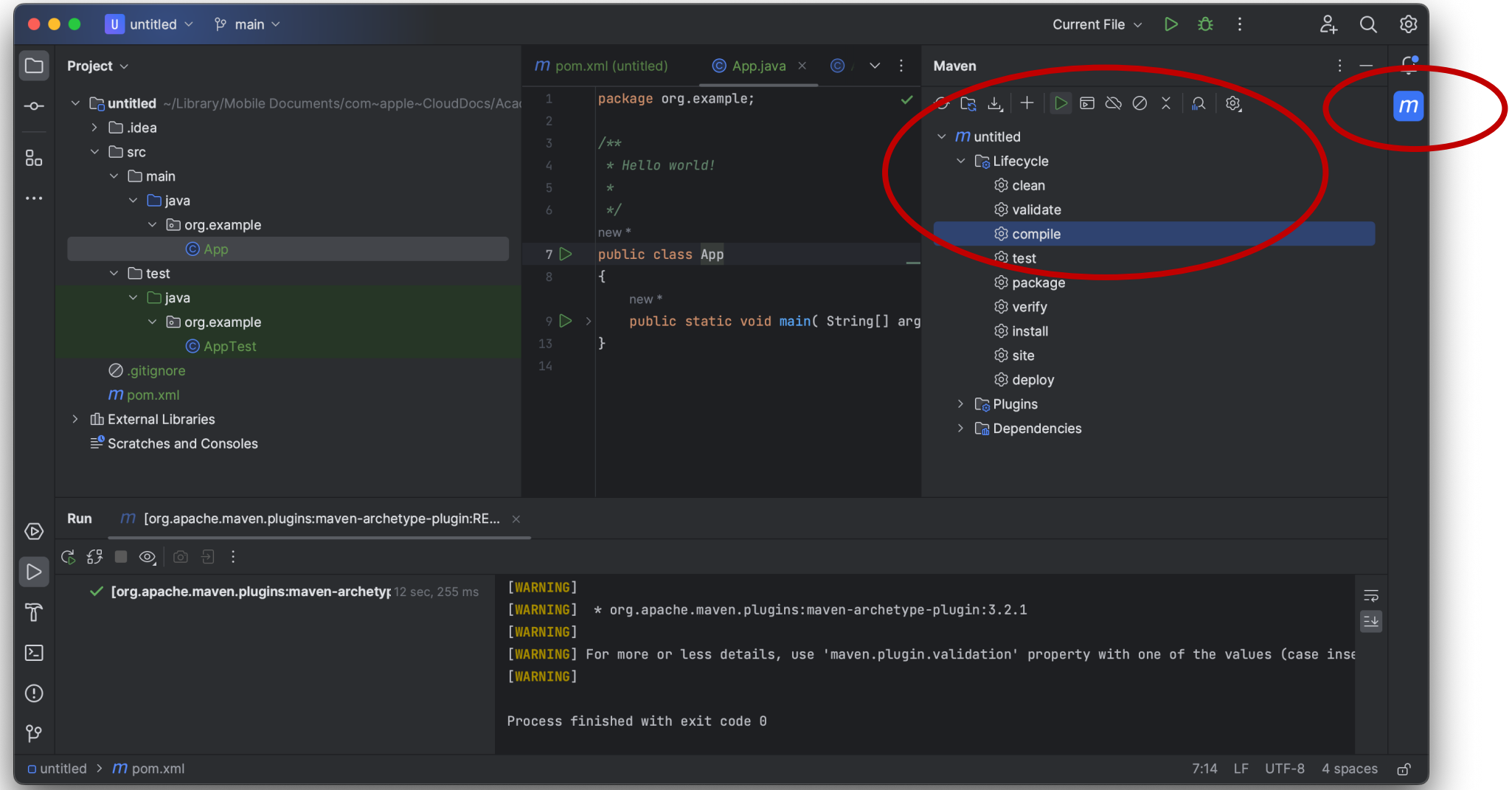
```
my-app
|-- pom.xml
`-- src
    |-- main
    |   `-- java
    |       `-- com
    |           `-- mycompany
    |               `-- app
    |                   `-- App.java
    `-- test
        `-- java
            `-- com
                `-- mycompany
                    `-- app
                        `-- AppTest.java
```

Creating a New Maven Project in IntelliJ

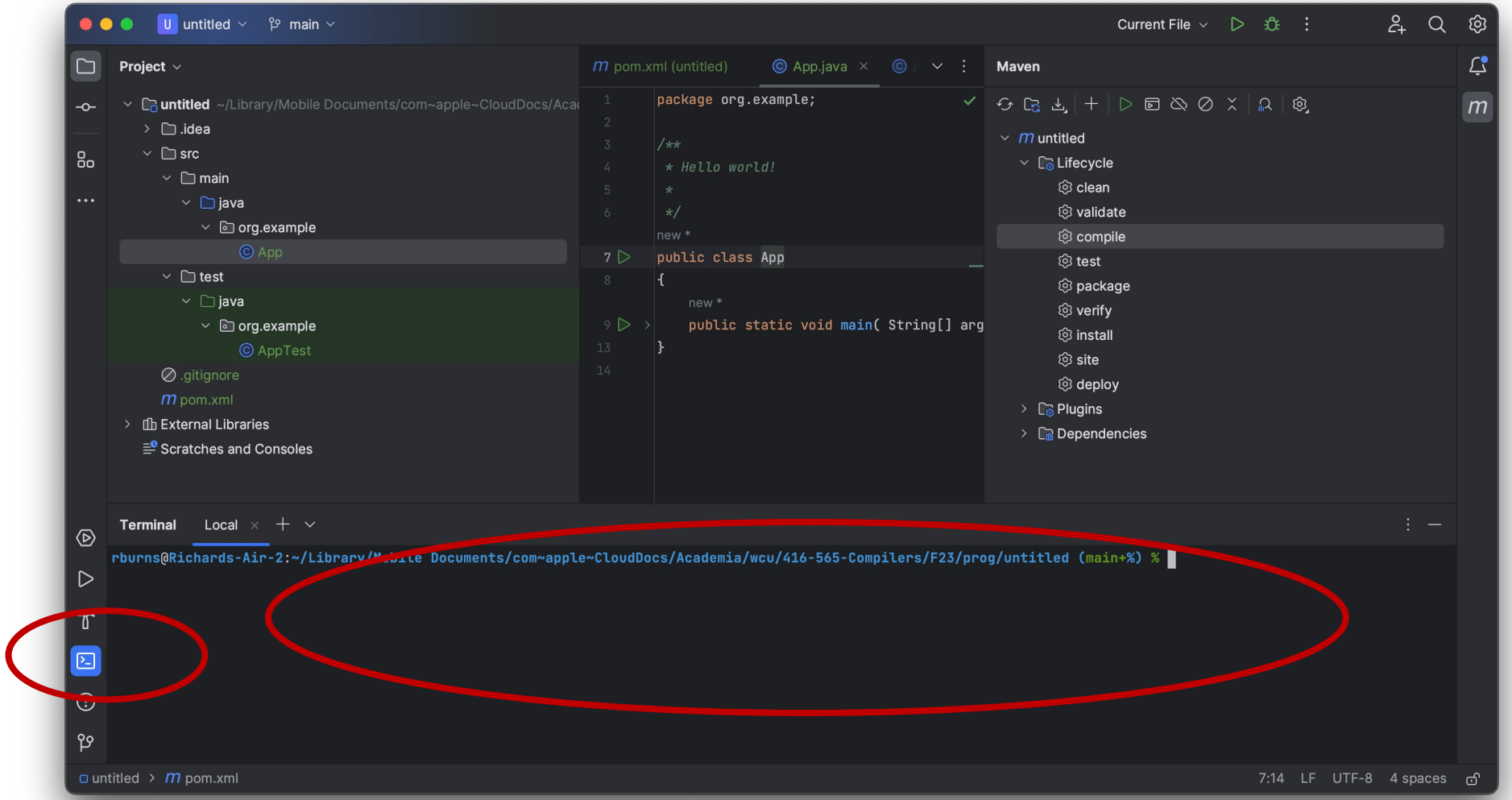



untitled
main
Current File
Project
untitled ~/Library/Mobile Documents/com~apple~CloudDocs/Aca
.idea
src
main
java
org.example
App
test
java
org.example
AppTest
.gitignore
pom.xml
External Libraries
Scratches and Consoles
pom.xml (untitled)
App.java
AppTest.java
package org.example;
/**
* Hello world!
*
*/
new *
public class App
{
new *
public static void main(String[] args) { System.out.println("Hello World!"); }
}
Run
[org.apache.maven.plugins:maven-archetype-plugin:RE...
[org.apache.maven.plugins:maven-archetyp 12 sec, 255 ms
[WARNING]
[WARNING] * org.apache.maven.plugins:maven-archetype-plugin:3.2.1
[WARNING]
[WARNING] For more or less details, use 'maven.plugin.validation' property with one of the values (case inse
[WARNING]
Process finished with exit code 0
untitled > src > main > java > org > example > App
7:14
LF
UTF-8
4 spaces

Running Maven Lifecycle Phases



Opening a Terminal



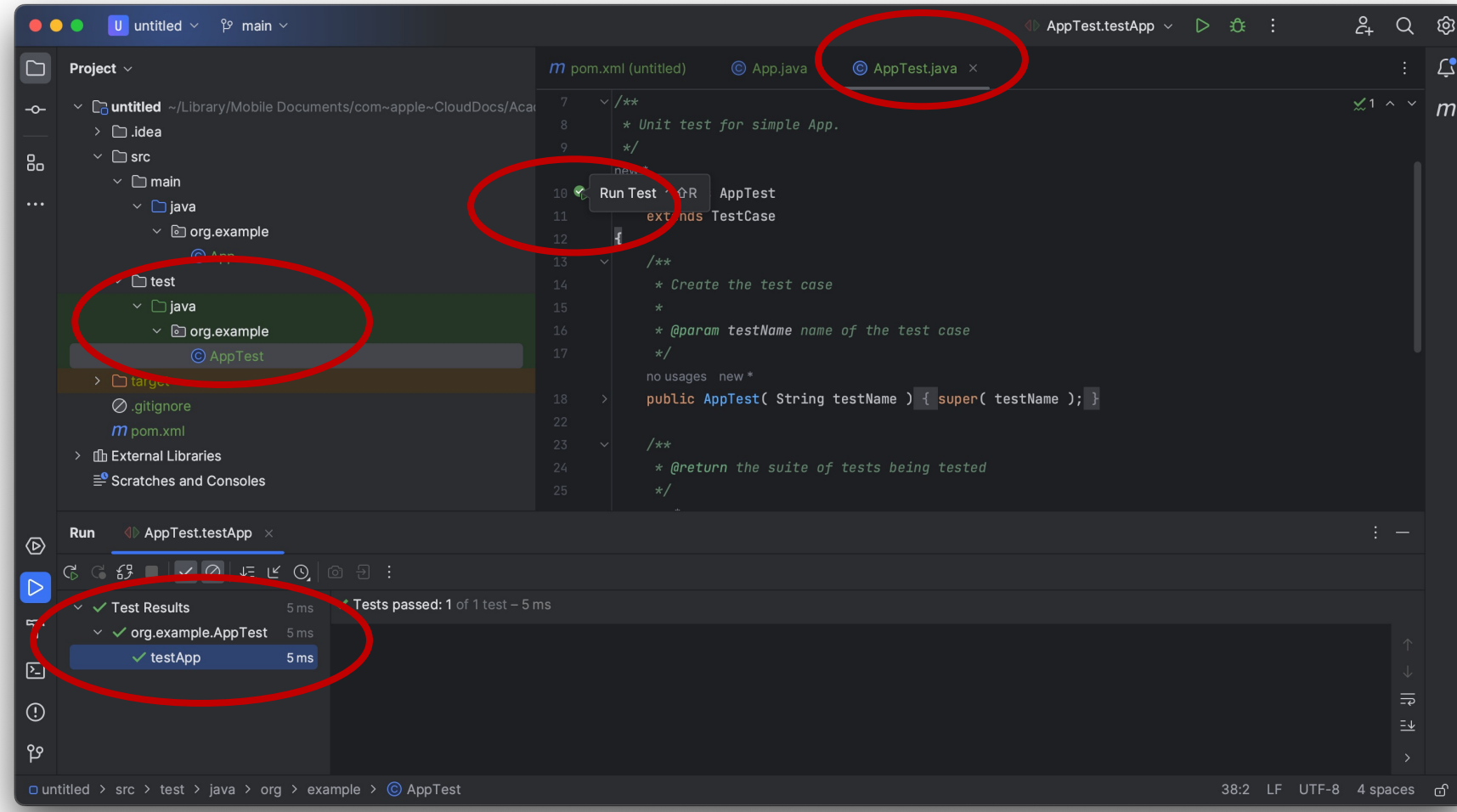
JUnit

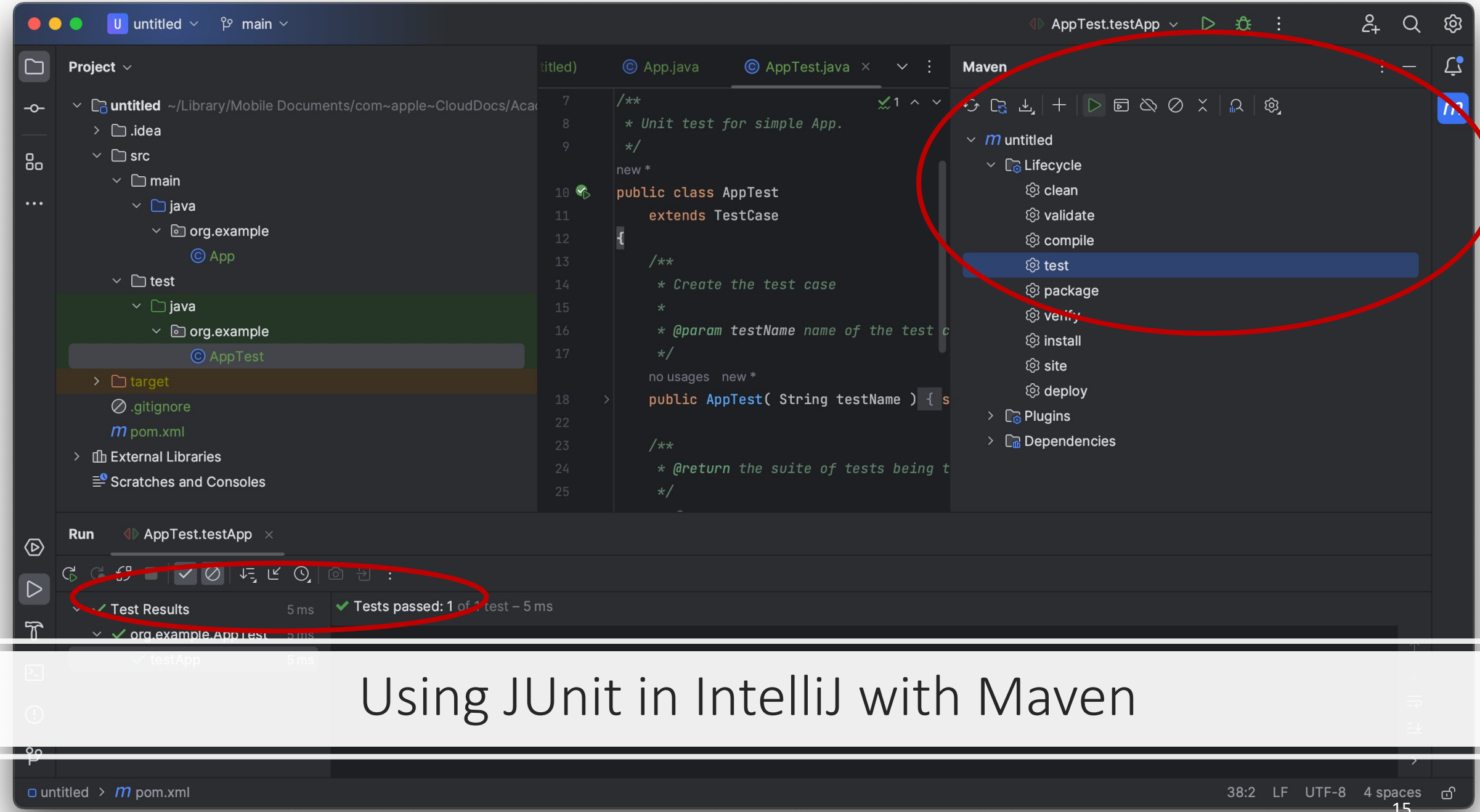
https://junit.org/junit4/

- One of many popular unit testing frameworks for Java
- Will be using JUnit 4.13 in this course
  - Specified and configured in the Maven `pom.xml` file
  - No need for you to manually download and install Junit
- *Note:* JUnit 5 also exists

- I will be writing your unit tests, so you do not create JUnit code yourself

Using JUnit in IntelliJ





Using JUnit in IntelliJ with Maven

Submitting Assignments

GitHub Classroom

Prerequisite:

- A GitHub account – sign up for one if needed!
  - Probably not necessary, but consider adding your WCUPA email address as one of the emails associated with the account

No prior Version Control experience necessary!

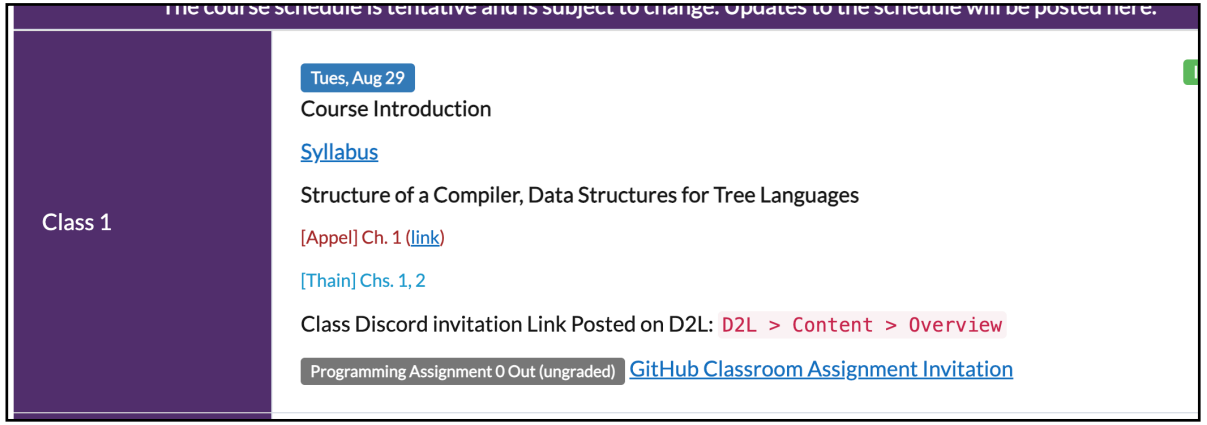
In CSC416/565:

- You will accept “assignments” that are hosted in GitHub Classroom.
- Immediately GitHub Classroom will prompt you to link your GitHub account with your identifier in the class roster
  - *(Thus, I will know who you are, despite whatever your GitHub is.)*

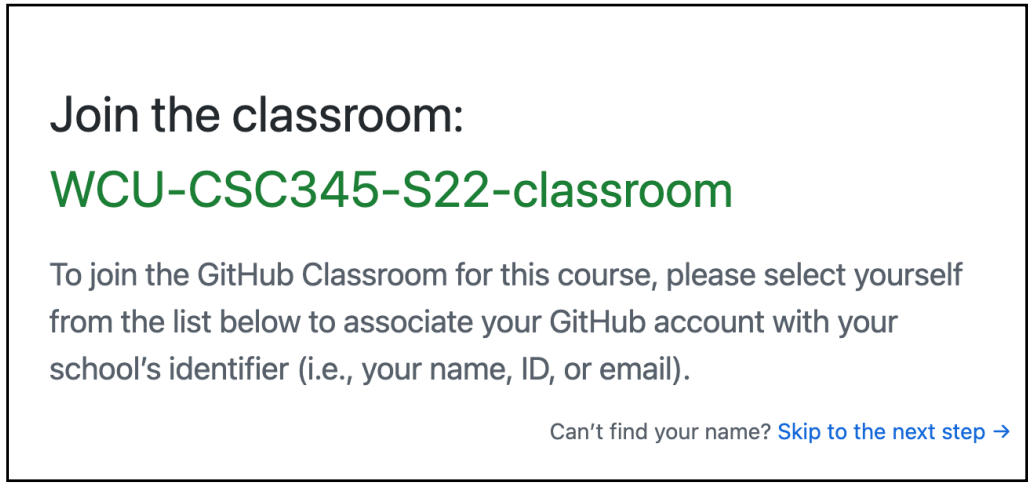
“Programming Assignment 0”

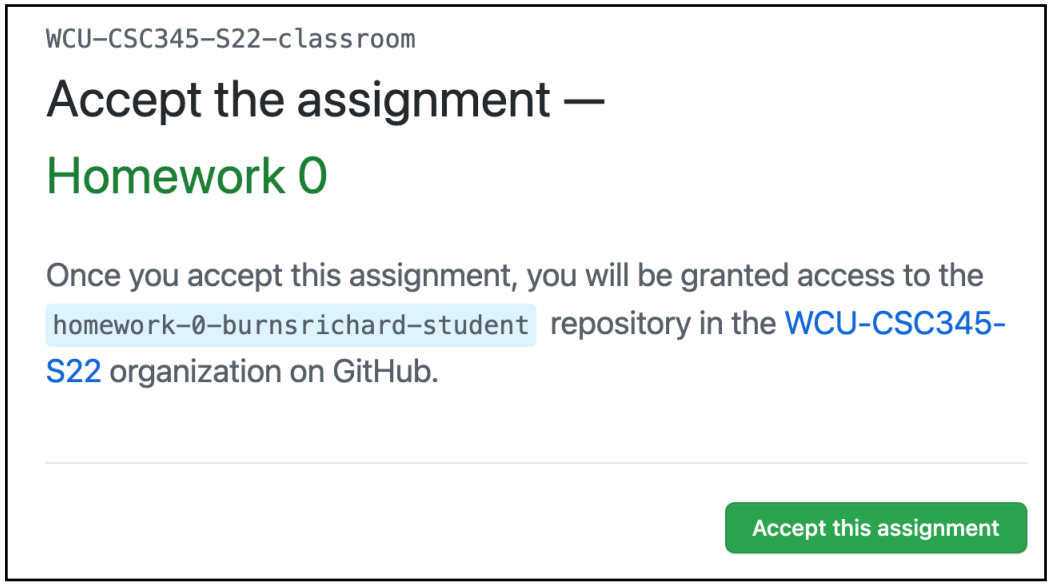
- Before we use GitHub in real, graded assignments, we’ll practice it in a “Programming Assignment 0”
- Like a real HW:
  - There are small tasks to do
  - There is a deadline
  - The assignment is auto marked correct or incorrect w/ unit tests
  - There is Java/IntelliJ/Maven/JUnit/Git/GH practice
- But:
  - It’s not graded. (Doesn’t count for anything.)

Step 1: Login to GitHub



Step 2: Follow “GitHub Classroom Assignment Invite” link





You accepted the assignment, **Homework 0** . We're configuring your repository now. This may take a few minutes to complete. Refresh this page to see updates.

Your assignment is due by **Feb 3, 2022, 23:59 EST**

You're ready to go!

You accepted the assignment, **Homework 0**.

Your assignment repository has been created:

`https://github.com/WCU-CSC345-S22/homework-0-burnsrichard-student`

We've configured the repository associated with this assignment (update).

Your assignment is due by **Feb 3, 2022, 23:59 EST**

programming-assignment-0-burnsrichard-student Private

Watch 0

main 1 branch 0 tags Go to file Add file Code

burnsrichard tests updates a16a543 2 hours ago 14 commits

.github	tests updates	2 hours ago
src	Update App.java	14 hours ago
.gitignore	Initial commit	14 hours ago
README.md	Initial commit	14 hours ago
pom.xml	Initial commit	14 hours ago

README.md

Programming Assignment 0

Date Out: Tuesday, August 29

Due Date: Tuesday, September 12 (start of class)

This introductory programming assignment will help familiarize you with the development environment of this course:

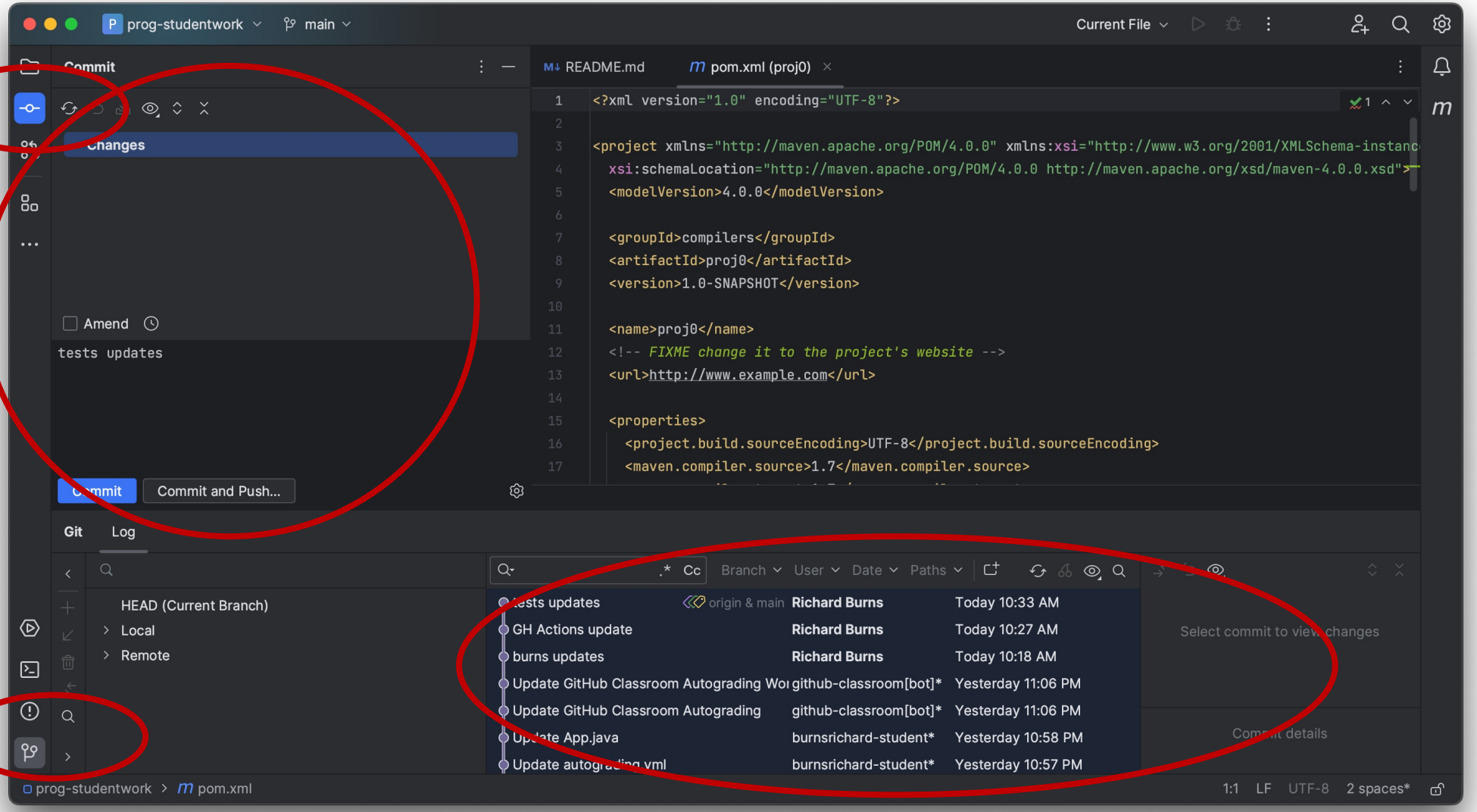
- Java for programming
- IntelliJ CE (community edition) IDE for coding, debugging, and more!
- Maven, a project manager for specifying project structure and dependencies

Next Steps

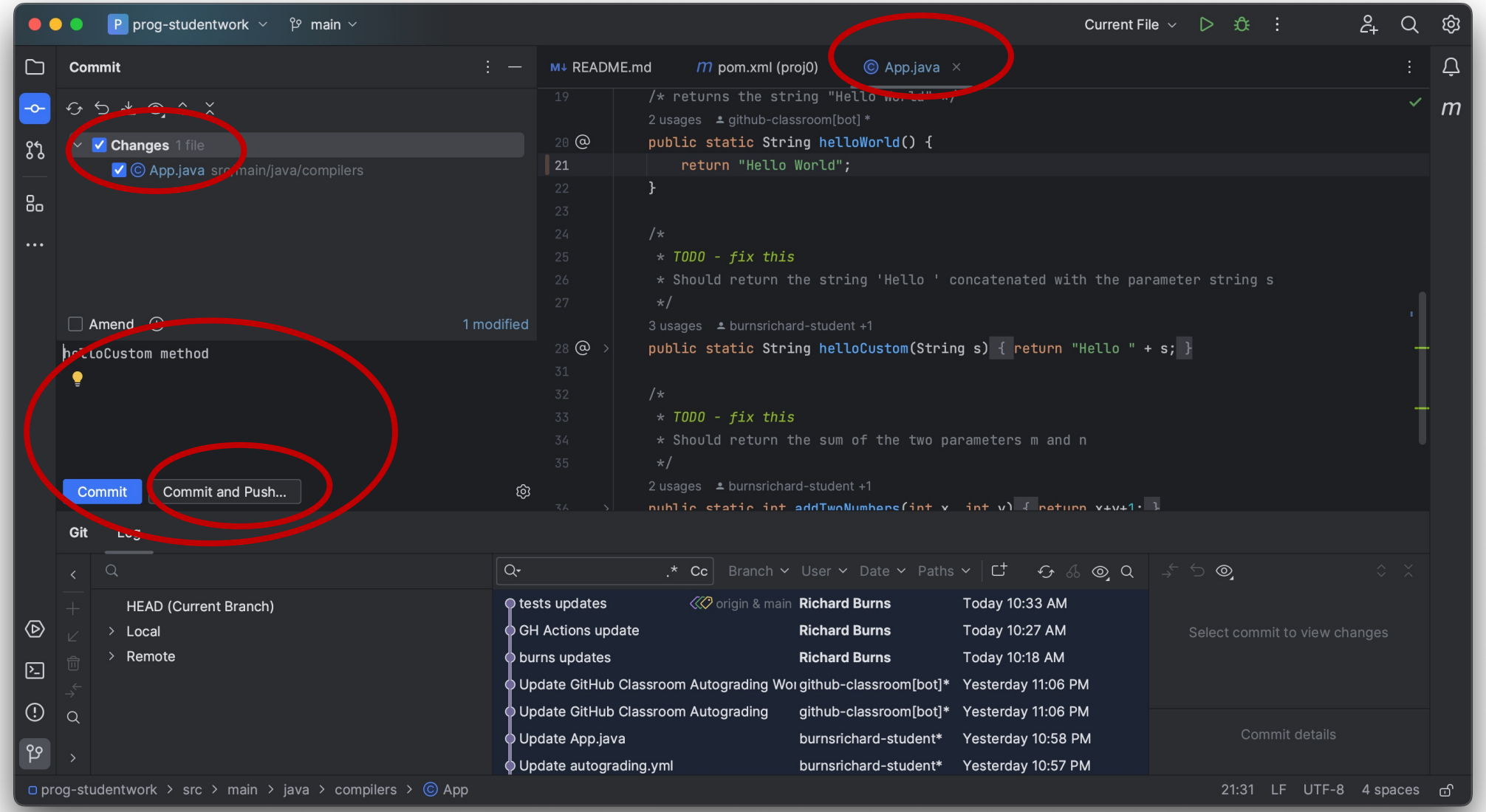
See the README.md (assignment description)

1. Open IntelliJ. Create a “New” -> “Project from Version Control”.
2. This will *clone* your remote GH repo onto your computer.
3. Write / test the code.
4. Upload/commit the code to your GitHub Classroom repository.
5. Look at and inspect the results of your GitHub auto-graded actions.

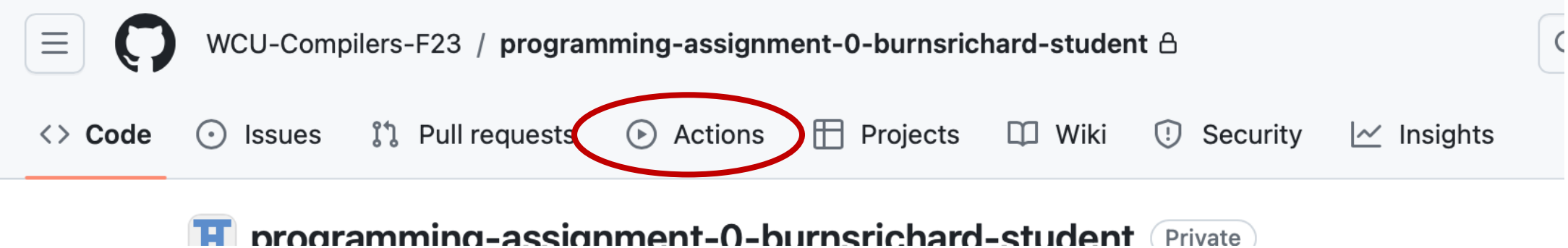
Support for Git and version control is already integrated with IntelliJ



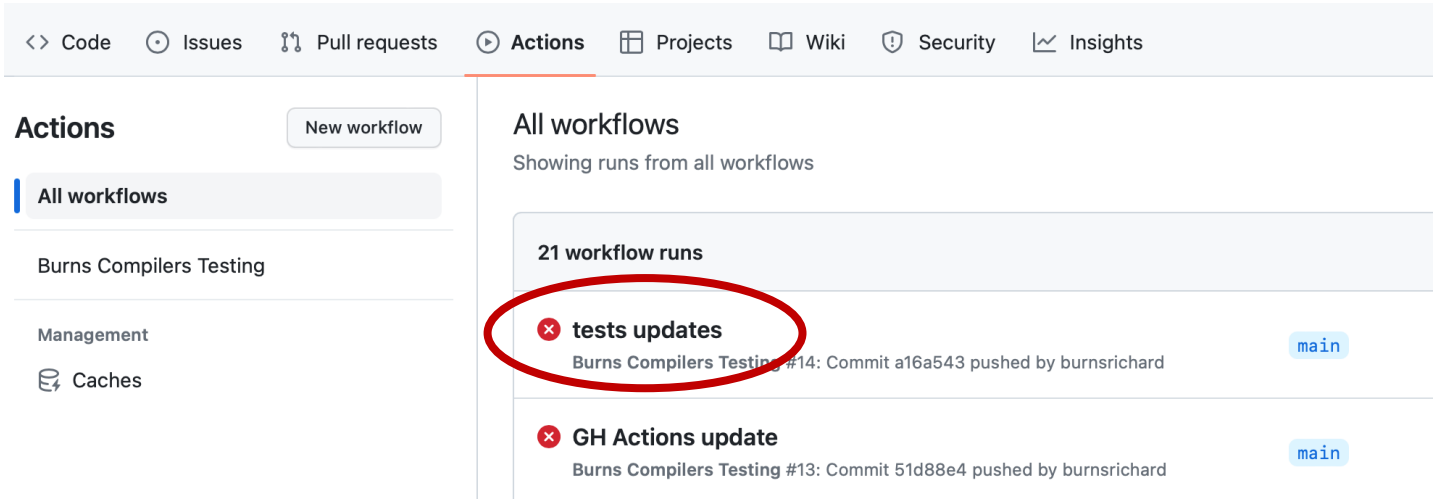
To save changes, commit to your local repo, and also push to your remote GH repo



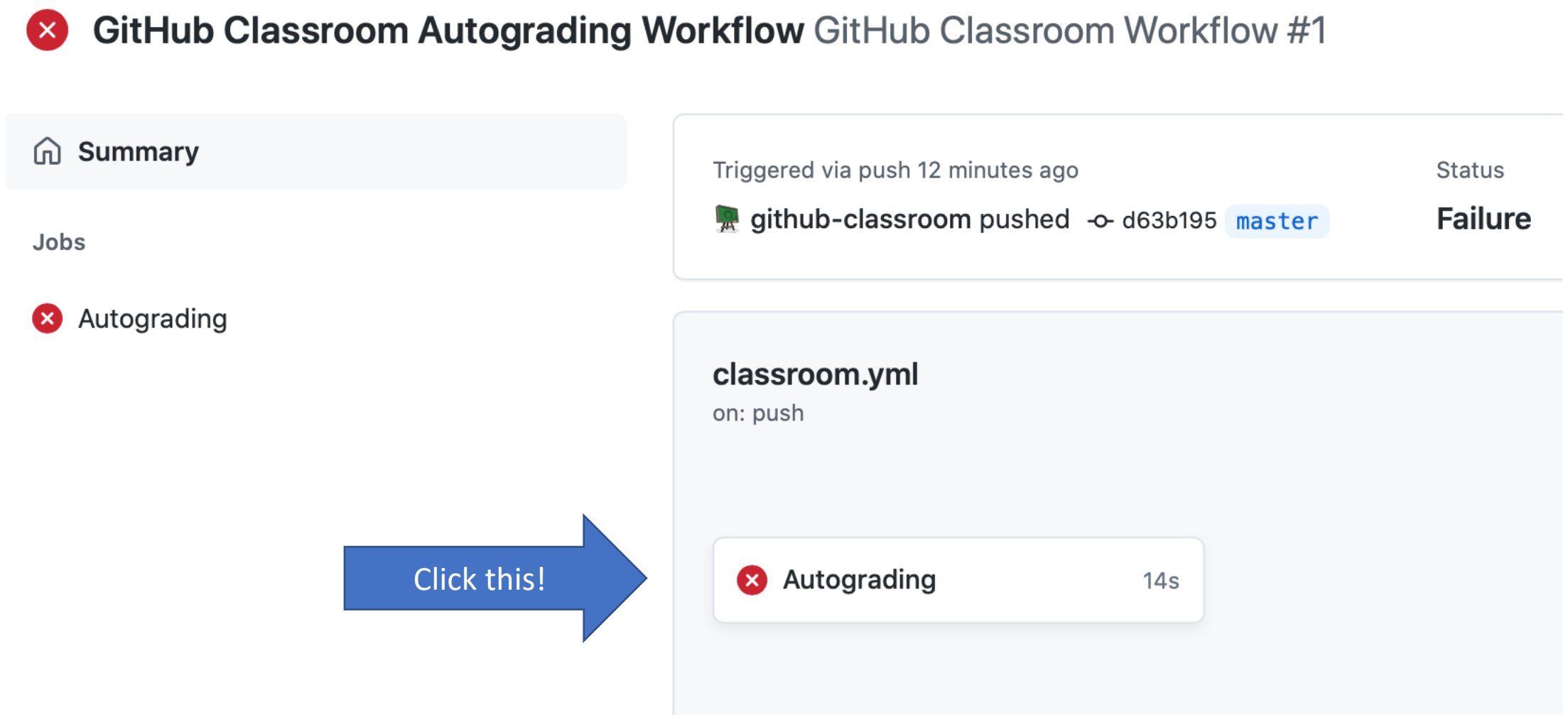
Autograded Tests and GitHub Actions



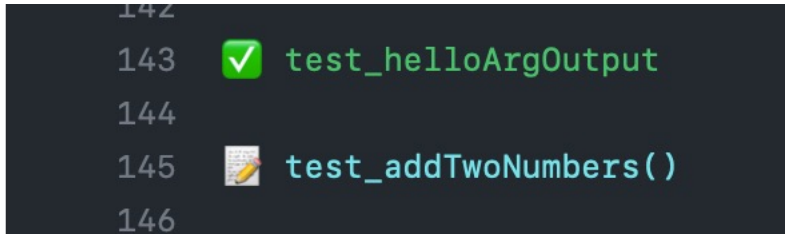
- Initially, GH tests your starter code when it is created and some unit tests fail
- Expand the workflow by clicking on “GitHub Classroom Autograding Workflow”



- Then click on “Autograding”

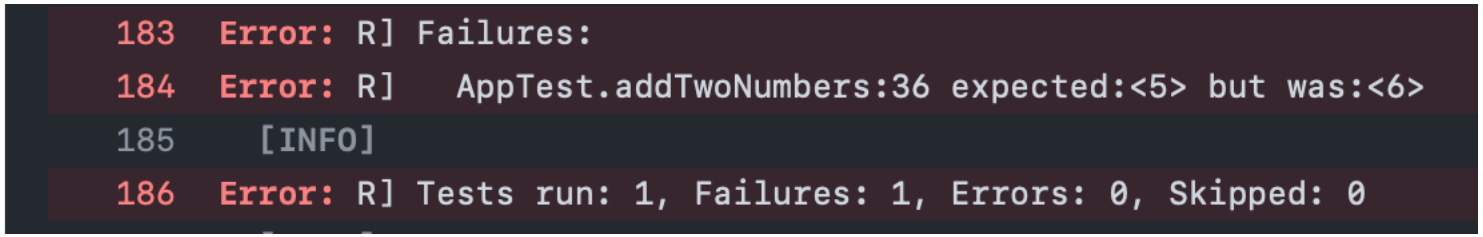


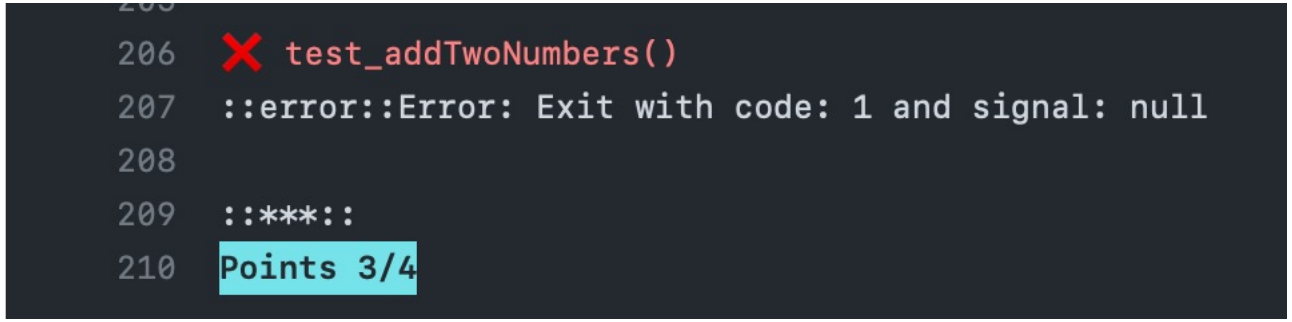
Autograding Output



There is a lot of info here, but if you scroll down:

- Some tests may pass
- Some tests may fail
- Overall score



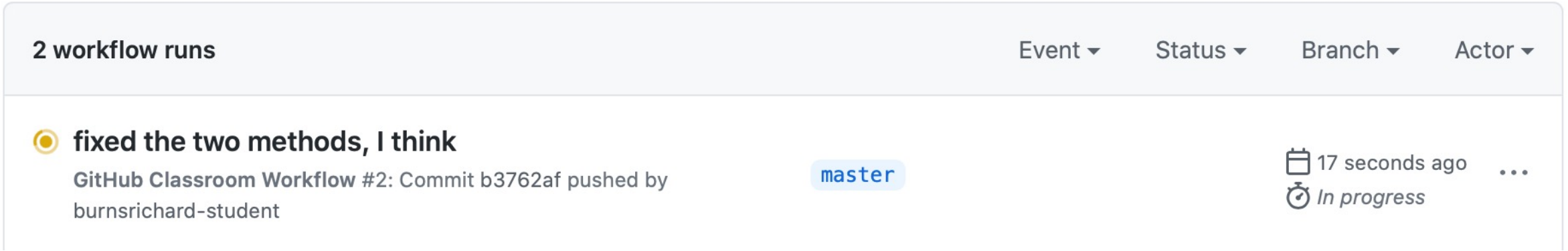


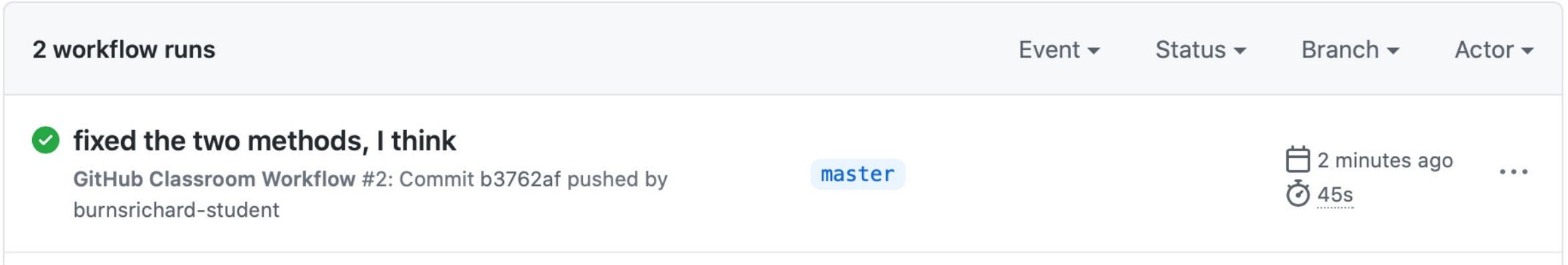
Ideally, code/test your code locally first. But then verify that after pushing that your GH tests are also passing.

Consult the testing directory in Maven which contains the unit tests.

Be Patient!
Autograding sometimes takes a few minutes

- Under “Actions”, your latest commit should may say “Queued” or “In progress”
  - This means your code will be auto unit tested very soon





Scores are automatically populated to me in my admin view on GitHub Classroom