

Gradescope Autograder

Designed for CSSE220 | [GitHub Link](#) | Written by [Canon Maranda](#)

Assignment Requirements

All assignments must meet the following requirements for the autograder to work:

- The assignment is graded with Junit tests
- Students are editing a known list of files, excluding unit test files
- Every unit test directly contributes to the student's grade
 - o This means extra credit problems are not yet supported

Here are a few examples of assignments that **do not** work with the autograder software:

- Java swing graphics
- Assignments where students create their own unit tests

Getting Started

Download the latest version of Gradescope Autograder from [GitHub](#). Follow the instructions in the README file. If a pre-made ZIP file already exists for the assignment, you may download that instead and ignore the above instructions.

Preparing Assignment in Gradescope

Create a new programming assignment in the course. Set the number of autograder points to the maximum score obtainable from unit tests. If commenting or code quality factor into the grade, check the box next to Enable Manual Grading. After the assignment is created, open the assignment settings page. Under Submission Methods Enabled, make sure Upload is the only option checked. Follow the following format for the Ignored Files textbox:

*

!NameOfFile.java

Where `NameOfFile` is the name of any file you want the student to submit. Use a new line for each accepted file, following the given format. This process prevents students from submitting files that can potentially overwrite autograder code. If you want to learn more, search for `.gitignore` format.

Click on Configure Autograder. Use the Replace Autograder button to select your ZIP file for the assignment. Set the Base Image OS to Ubuntu, Base Image Version to 22.04, and Base Image Variant to JDK 17. Click Update Autograder and wait for the process to finish.

Additional Feature: MOSS

Gradescope will automatically perform similarity checks on code using MOSS. Click Review Similarity, select Java as the language, and upload instructor-provided code. Note that this tool flags comments that are formatted differently from the original file. To fix this, open all the files in Eclipse and run the code formatter, then upload both versions of each file to the similarity checker.