

Use autograding

In this article

About autograding

Grading methods

Configuring autograding tests for an assignment

Viewing and downloading results from autograding tests

Further reading

You can automatically provide feedback on code submissions from your students by configuring tests to run in the assignment repository.

Who can use this feature

Organization owners who are admins for a classroom can set up and use autograding on assignments in a classroom. For more information on classroom admins, see "[Manage classrooms](#)."

About autograding

You can use autograding to automatically check a student's work for an assignment on GitHub Classroom. You configure tests for an assignment, and the tests run immediately every time a student pushes to an assignment repository on GitHub.com. The student can view the test results, make changes, and push to see new results.

After a student accepts an assignment, on every push to the assignment repository, GitHub Actions runs the commands for your autograding test in a Linux environment containing the student's newest code. GitHub Classroom creates the necessary workflows for GitHub Actions. You don't need experience with GitHub Actions to use autograding. For more information on workflows and GitHub Actions, see "[About continuous integration](#)."

You can use a testing framework, run a custom command, write input/output tests, or combine different testing methods. The Linux environment for autograding contains many popular software tools. For more information, see the details for the latest version of Ubuntu in "[Using GitHub-hosted runners](#)."

You can see an overview of which students are passing autograding tests by navigating to the assignment in GitHub Classroom. A green checkmark means that all tests are passing for the student, and a red X means that some or all tests are failing for the student. If you award points for one or more tests, then a bubble shows the score for the tests out of the maximum possible score for the assignment.

Grading methods

There are two grading methods: input/output tests and run command tests.

Input/output test

An input/output test optionally runs a setup command, then provides standard input to a test command. GitHub Classroom evaluates the test command's output against an expected result.

Setting	Description
Test name	The name of the test, to identify the test in logs
Setup command	<i>Optional.</i> A command to run before tests, such as compilation or installation
Run command	The command to run the test and generate standard output for evaluation
Inputs	Standard input for run command
Expected output	The output that you want to see as standard output from the run command
Comparison	<p>The type of comparison between the run command's output and the expected output</p> <ul style="list-style-type: none">• Included: Passes when the expected output appears anywhere in the standard output from the run command• Exact: Passes when the expected output is completely identical to the standard output from the run command• Regex: Passes if the regular expression in expected output matches against the standard output from the run command
Timeout	In minutes, how long a test should run before resulting in failure
Points	<i>Optional.</i> The number of points the test is worth toward a total score

Run command test

A run command test runs a setup command, then runs a test command. GitHub Classroom checks the exit status of the test command. An exit code of `0` results in success, and any other exit code results in failure.

GitHub Classroom provides presets for language-specific run command tests for a variety of programming languages. For example, the **Run node** test prefills the setup command with `npm install` and the test command with `npm test`.

Setting	Description
Test name	The name of the test, to identify the test in logs
Setup command	<i>Optional.</i> A command to run before tests, such as compilation or installation
Run command	The command to run the test and generate an exit code for evaluation
Timeout	In minutes, how long a test should run before




Points

Optional. The number of points the test is worth toward a total score

Configuring autograding tests for an assignment

You can add autograding tests during the creation of a new assignment. For more information, see "[Create an individual assignment](#)" or "[Create a group assignment](#)."

You can add, edit, or delete autograding tests for an existing assignment. All changes made via the Classroom UI will be pushed to the existing student repositories, so use caution when editing your tests.


- 1 Sign into [GitHub Classroom](#).
- 2 In the list of classrooms, click the classroom you want to view.
- 3 To the right of the assignment you want to edit, click .
- 4 In the left sidebar, click **Grading and feedback**.
- 5 Add, edit, or delete an autograding test.
 - To add a test, under "Add autograding tests", select the **Add test** dropdown menu, then click the grading method you want to use. Configure the test, then click **Save test case**.
 - To edit a test, to the right of the test name, click . Configure the test, then click **Save test case**.
 - To delete a test, to the right of the test name, click .
- 6 At the bottom of the page, click **Update assignment**.

Viewing and downloading results from autograding tests

Download autograding results

You can also download a CSV of your students' autograding scores via the "Download" button. This will generate and download a CSV containing a link to the student's repository, their GitHub handle, roster identifier, submission timestamp, and autograding score.

View individual logs

- 1 Sign into [GitHub Classroom](#).
- 2 In the list of classrooms, click the classroom you want to view.
- 3 In the list of assignments, click the assignment you want to view.
- 4 To the right of a submission, click .
- 5 Review the test output. For more information, see "[Using workflow run logs](#)."

Further reading

- [GitHub Actions documentation](#)

Legal