

Retrieving Graded Labs

Our goal is to give you feedback that is easy-to-find and easy-to-read. So in this class, we will provide grades and feedback on your lab assignment via git and GitLab in the form of a "Merge Request". Oftentimes, feedback from one week on the lab will help you improve in future labs, so it is important to go through your feedback!

For each lab that you submit, we will create a new "feedback" branch in your repository. In that branch, your submitted programs will be supplemented with grader feedback. Feedback will contain high-level comments from all graders, the grade you received on the lab, and suggestions for improvement. When you look in the .java files that you submitted, there will be new comments, preceded by //\\$ (or /*\\$ */ for multi-line comments) to indicate that the comment is from a grader. In other words, all comments that a grader leaves inside your Java files will be easily distinguishable from code comments that were part of your original submission.

The instructors will have issued a "Merge Request", which easily allows you to see what has been added. You have the option of merging the comments into your own master branch or keep it as a separate branch. You might try first reviewing the comments without merging and then merge and review again to see which approach you prefer.

1. Open the Project in GitLab by logging in to <https://evolene.cs.williams.edu> and clicking on the 'Projects' button in the upper-left corner. Select the Lab project for which you'd like to view the feedback.
2. Open the Project in gitlab and navigate to the Merge Requests option in the side bar:

The screenshot shows the GitLab project 'ephelia1'. The sidebar on the left is highlighted with a red oval around the 'Merge Requests' link, which has a value of '1' next to it. The main content area shows the 'Details' page for a merge request. At the top, it says 'task 1 complete' and 'cs134 authored 7 minutes ago'. Below this, there are buttons for 'README', 'Add CHANGELOG', 'Add CONTRIBUTING', 'Enable Auto DevOps', and 'Add Kubernetes cluster'. At the bottom, there is a table showing file commits:

Name	Last commit	Last update
README.md	updated link to PDF	1 week ago
hello.py	task 1 complete	7 minutes ago
honorcode.txt	updating	1 week ago
README.md		

3. Select the "instructor feedback" request:

The screenshot shows a list of merge requests. One specific request titled "instructor feedback" is highlighted with a red circle. The request was opened 34 minutes ago by user cs136. It has 0 comments and was updated 34 minutes ago.

4. Select the "Changes" option to view the diff between the two branches:

The screenshot shows the "Changes" tab for a merge request. It displays a summary of changes: 1 commit and 1 merge commit will be added to master. There is an option to "Modify merge commit". Below this, there is a note: "You can merge this merge request manually using the command line". At the bottom, there are like, dislike, and smiley icons, and a discussion section with 0 comments, 1 commit, and 2 changes. The "Changes" tab is circled in red.

5. You can choose to view either "Inline" or "Side-by-Side" (everything that has been added shows up with a +next to it in green):

The screenshot shows the "Changes" tab with the "Inline" view selected. It displays the differences between the latest version and the master branch for two files: GradeSheet-lab1.txt and hello.py. In GradeSheet-lab1.txt, there are 20 additions (green) and 1 deletion (red). In hello.py, there are 12 additions (green) and 1 deletion (red). The code snippets show the changes made to each file.