
Resubmissions

You have 2 resubmissions at your disposal. You may resubmit the midterm exam and any lab except the final lab. A high-quality resubmission recoups up to 50% of the missing points.

Resubmissions must be given to me before the end of the final exam reading period.

We really do read every one of these. Please take your time and do your work carefully and neatly so that we can understand what you mean. We want to give you back the maximum amount of points, so make it easy for us to do so.

Resubmission must include:

1. the original solution,
2. the updated solution, and
3. a document explaining your mistakes.

This document must be typed or it will not be accepted.

Goal

Critically, your explanation document must describe, in plain English, *what* you did wrong, *why* you made the mistake, and *how* your new solution fixes the problem. If you learned something from your mistake, tell me. In short, the goal of this exercise is to give you a second chance to demonstrate that you learned a concept.

How to Resubmit

For a midterm resubmission:

Your resubmission should be *on paper*. You should give me:

1. your original exam, and
2. a *typed document* explaining your mistakes that includes reworked solutions.

I recommend using a *paper clip* to bind the two documents together so that I can pull them apart and compare them side-by-side. Please put this resubmission either *directly into my hands* or *into my department mailbox*. If you put the resubmission into my department mailbox, please send me an email to let me know that you put it there.

For code resubmissions: For code, you are going to create a new *branch* in your `git` repository, commit your changes, and then push those changes to `evolene` so that we can see them.

To do that, `cd` to your assignment's code directory, and run the following:

1. Run `git checkout -b resubmission`, which will create a new branch called `resubmission`.
2. Make corrections to your assignment's solution.
3. Run `git add` and `git commit` as appropriate to save your changes.
4. To push to `evolene`, run `git push --set-upstream origin resubmission`. We need to push differently than usual because the `resubmission` branch you just created does not exist on `evolene`. The above command tells `git` to create the branch on `evolene` if it does not already exist.
5. Go to your repository on `https://evolene.cs.williams.edu` and verify that your new `resubmission` branch appears in the web interface.

When you are done, fill out the form at the following URL so that we know you submitted a resubmission: `https://forms.gle/eszPFQrL62kUjauA8`.