

## Homework Self-Assessment Guidelines

In this offering of CS 332, we are asking you to complete **brief** (no more than one page of writing) self-assessments of your homework assignments within one week of having submitted them. The goals of these exercises are to:

- Give you an opportunity to engage with posted homework solutions carefully.
- Identify specific ways you can improve the content and clarity of your work.
- Gain experience critically evaluating (your own) technical work.
- Give you credit for something you should be doing anyway. :)

We are flexible about both the format and the content of the self-assessments you submit. What follows are just our suggestions for what we think could be both useful and efficiently implementable. But you should feel free to do what you think helps you learn best. Note that **this is not meant to be an onerous or time-consuming task**. If it becomes one, let us know, and we'll figure out how to fix it. This whole thing is new and experimental, and it could fail! As with all parts of this course, we welcome and encourage your feedback on whether you find these exercises worthwhile, whether you don't, or whether you have suggestions to change them.

If we have time, we will do our best to provide feedback on the self-assessments you submit. But they will only be "graded" on participation. As with the other participation components of this course, you can miss a few (15%) of these activities without penalty.

Here's what we envision as a workflow for carrying out a self-assessment exercise.

**Compare Solutions** For each problem on a given homework assignment:

1. Remind yourself of the problem statement. If you were asked to prove something, check that you understand all of the hypotheses (conditions that you're allowed to assume), the desired conclusion, the direction of the implication, and all of the stated quantifiers and their order.
2. Reread your own solution to the problem. Think about the following questions:
  - (a) Does your solution interpret and address the problem correctly?
  - (b) Are the main components (the skeleton) of the arguments present and clearly stated?
  - (c) Are the smaller supporting statements and calculations well-justified? Does each statement follow clearly from the previous ones? Or are there any unwarranted claims or assumptions?
  - (d) Is notation clear and unambiguous? Does it all type check?
3. Now read the provided sample solution to the problem. Map out the structure of the argument. What are the main components, and what are the supporting statements? Is there anything about the solution that you don't understand, or that you aren't sure is correct? (If so, ask us!)

4. Think about what the two solutions did similarly, and what they did differently. Were there major differences in the plan of attack? (There might be, and that's great! There can be many different ways to solve the same problem.) Were there places where one solution provided more detail than the other – if so, do you think this was necessary? Why or why not? Are there differences in the precision or clarity of the writing? Does your evaluation of your solution along any of the dimensions in Item 2 change? In hindsight, was there a definition or a concept that didn't quite click the first time around? Does this comparison give you ideas for how you could improve either solution?

**Report Your Findings** Record your evaluation and insights however is efficient and useful to you. Some ideas for doing this include:

1. Annotate the PDF of your submitted solution with comments reflecting your evaluation, what you learned, and suggestions for improvement. You don't have to do this for every problem – just the ones for which you found something significant to report.
2. Put together a brief narrative summary capturing the most important points from your evaluation of all of the problems. For instance, you can discuss which ideas and concepts your submitted solutions showed you understood, which ones you're still working on, and what this exercise taught you about them. You can also describe some of the correctness / clarity gaps you may have found in your solutions, and what you learned that could help improve them.
3. Mix-and-match both: You can annotate some of your solutions, while providing a narrative summary to capture the others.
4. If the solutions you submitted were perfect the first time around and you didn't learn anything, you can report that too. We'd also appreciate critiques of our solutions.