

CS130(0) Style Guide

Introduction

Every assignment you hand in for this class should be written with the usability for the grader in mind. **Your assignments should feel like engaging portfolio-ready pieces that someone outside of this class can easily understand** - this means the reader should be able to understand the premise of the project without reading the assignment handout. To be 'portfolio-ready', the material should also be interesting and engaging enough that a viewer would go out of their way to read it - i.e. if they saw your handin as an article on Medium, they would click on it to read it. This content style guide offers some tips on how to achieve that.

Why is presentation important?

In industry, you'll often be tasked with communicating the value of your work. You might get questions from developers on how to implement your designs, or a product manager might not think your idea is worth executing. It's important to present your work nicely to make sure the high-quality content and hard effort you put in it is not lost in communication.

Additionally, we want you to get practice presenting your work outside the context of the assignment and therefore are engaging as works on their own. In industry, you'll also have to be able to present your projects as stand-alone pieces where a reader without context of the assignment you had would be able to understand your thought process and design choices.

How can I ensure my work is presented well?

Below are some tips for good presentation. Please note that following all of these does not necessarily guarantee a perfect score, and an assignment can receive a perfect score even if it does not follow each of these. Ultimately, the presentation portion of your score will be based on the rule of thumb of whether or not it's portfolio-ready.

For images:

- Don't use blurry images.
- For paper sketches, make high quality scans instead of taking photos by phone.
- For digital illustrations (i.e. most wireframes and prototypes), save them as vector-based formats like svg, ai, or eps whenever possible. It's generally not advised to save your illustrations as png

or jpg before embedding in your pdf.

- For normal photographs or native raster images (e.g. a profile photo of a persona, image of a waterfall), formats like png and jpg images are fine. Jpgs are lossy (the quality degrades when edited) and should be used only for photographs.
- Keep text separate from your images as much as possible. If you must include text inside an image, make the font size large enough to read.
- Use colors helpfully, but not gratuitously.
- If images are referenced in the text, label them helpfully with names like "Figure 1".

For writing:

- **Make sure the reader would be able to understand the assignment if they didn't read the handout of the assignment beforehand.** This means don't reference the name of the assignment in any titles or text within the assignment and be sure to explain the premise of the project.
- Adhere to the 2-page limit. This applies to text only -- labeled figures may be added to the end of your hand-in. It is okay to add text in your visuals in moderation.
- Proofread! Use spell-check.
- Be concise, and don't repeat information. To show that something is important, you can underline or italicize it.
- Break long paragraphs into smaller chunks of text.
- Using big words might not be helpful. Aim not to impress your reader with your vocabulary, but with the quality of your work. If simpler words make your work stand out more, use those.
- Example:
 - "This user was discombobulated, riddled, and puzzled by this interface." → "This user was confused by the back button's placement."

For code:

- Your code should be written in a way that **allows another developer to quickly understand what's going on and add new features if necessary even if they haven't read the handout for the assignment.**
- Any code taken from online or elsewhere must be clearly cited in either a comment or a README or you will be in violation of the collaboration policy.
- Use descriptive names for variables and methods.
- Avoid repeating code.
- Comment helpfully!