

Code Review

Instructions for Reviewees

1. Choose a single file to have reviewed. It would preferably be a “model”, the way you represent the key internal objects that you work with.
2. Make sure to comment the file and structure it as best you can.
3. Do NOT include any identifiable elements (so no names, emails etc).
4. Upload your file to Moodle. Instructions to follow.
5. When looking at the reviews, keep in mind that they are objective evaluations of the code, not judgements on your coding skills. In other words, don't take the reviews personally.

Instructions for Reviewers

When doing code reviews of your peers, here are the items to address as you look at a single module file:

1. Does the top of the file contain a comment explaining what the module's role is?
 - Is that role clear?
2. Is the public interface for the file clearly shown somewhere (typically near the end of the file)?
 - Does the public interface seem right? Are there methods that are missing, or methods whose usefulness is not clear?
3. For each public function or constructor in the file:
 - Is there a comment describing what the function does?
 - Is it clear from just the comment what the use of the function would be?
4. For each function in the file:
 - Does the name of the function describe what the function's intent is?
 - Are the names of the function parameters and local variables clear and descriptive of what roles these variables play in the function?
 - Is the overall flow of the function clear to follow?
 - Is consistent formatting used?
 - Does the function “do too much”? Should some part of the function perhaps be separated into its own function?
 - Are comments within the function useful in clarifying any tricky parts of the function code?
 - Are there comments within the function that feel superfluous and don't really add much insight?
 - Are there parts of the code that are not clear to you? It is *not* your job to struggle to understand what the code is doing. It is the programmer's job to make the code as clear to readers as possible.

- You don't have to tell them how to "fix it", but simply note that these parts are hard for someone else to process. The way to improve it might end up being using different variable names, refactoring a part into a separate function, or adding comments, that is up to them how to proceed.