



Code Review Rubric

Professional (Be Kind and Helpful)

Criteria	Low Proficiency	Average Proficiency	High Proficiency
Giving Feedback	Comments are unclear, insulting, or hurtful. Comments do not explain what might be wrong.	Comments explain what might be wrong but could be clearer.	Comments are clear, kind, and explain what might be wrong.
Suggesting Fixes	Comments don't offer ways to make things better, OR comments give unhelpful suggestions.	Comments sometimes suggests ways to improve the code, sometimes are off-topic.	Comments always offer helpful ideas to make the code better.
Considering Feelings	Comments use a lot of "you" statements. Comments use blame or point fingers, focusing on the person instead of focusing on the code.	Comments are usually kind, avoid blame, and focus on the code instead of the person.	Comments are always kind, consider others' feelings, and use "I" statements. Comments focus on the code, not the person who wrote it.

Technical (Make Sure Everything Matches and Works Correctly)

Criteria	Low Proficiency	Average Proficiency	High Proficiency
Following the Rules	Often forgets, or doesn't know, correct syntax and code conventions	Remembers most of the rules but makes occasional errors	Always remembers and follows the coding rules, makes very few errors
Using Helpful Tools	Doesn't use or know about the tools that can help check the code.	Uses tools sometimes but might not always catch mistakes.	Always uses tools effectively to check the code first.
Looking at the Whole Picture	Only looks at the small details (i.e. syntax errors) but forgets about the problem or how all the code fits together.	Looks at both small details (i.e. syntax errors) and the whole code (i.e. how functions are being used)	Always checks how the new code fits in and thinks about how the parts of the code interact with each other.

		but might miss important connections	Considers how code is related to the problem being solved.
--	--	--------------------------------------	--

Collaborative (Share and Work Together)

Criteria	Low Proficiency	Average Proficiency	High Proficiency
Everyone Gets a Turn	Doesn't let others review their code. Ignores feedback.	Lets others review their code. Listens to and implements some feedback.	Always lets others review their code. Listens to and accepts feedback.
Keep It Short	Shows really long or confusing pieces of code for review.	Tries to keep code shorter but sometimes it's still too long.	Always shows short, clear pieces of code for easy review.
Work Side by Side	Refuses to work with others to do code reviews.	Works well with others; may need reminders on best ways to share ideas or solve problems together.	Works well with others. Is skilled at sharing ideas, and solving problems with a teammate or small group.