

Peer Evaluation

One of the goals of this course is to encourage good software development practices, especially when building a large software system (such as a compiler). For each project submission, you will be assigned two other random students in the course. You must review their code and offer constructive feedback. In industry, this process is known as a *code review* and is frequently used to improve software quality and to catch software defects early. This document lists the questions about the code that you should answer for each submission, giving examples of both *non-constructive* and *constructive* feedback (aim for the latter!) for each question.

Question	Example of Non-constructive Feedback	Example of Constructive Feedback
<i>What did you like about this submission?</i>	"It was good." "It was clean."	"The main loop in function X was clean and well-documented." "I liked the use of extra indentation to line up the array initializations in module X."
<i>Describe one significant difference between your own submission and this person's submission. Which approach is cleaner? Which approach is more efficient?</i>	"My code is faster." "My function X is shorter than this author's version. Mine is better."	"I chose to calculate the maximum value on every iteration of the outer loop while this author calculated it only once and cached the result. Their approach is more efficient, but my approach works even if the list is modified during iteration."
<i>Is the code well-formatted and well-documented? If not, suggest some specific improvements.</i>	"Not enough documentation." "I couldn't understand the code."	"The goal of function X was unclear; the author should add some documentation regarding its inputs and outputs." "The code is inconsistently formatted; the author should consider removing the extra empty lines in functions X and Y."
<i>Did you find any software defects? If so, briefly describe them.</i>	"Function X doesn't work." "I couldn't get it to run."	"Function X does not produce the correct output for this input: 'ABC'" "The program crashed with an IOException when I tried to run it on 'loops.decaf'."
<i>Do you have any other constructive comments for the author?</i>	"This code sucks and needs to be rewritten." "This code is perfect."	"Function X has redundant if-conditions; the last two could be consolidated." "The use of recursion in function Y to avoid ugly class-level data structures is very elegant."

In addition, please give each submission a numeric score in four categories (formatting, code style, elegance, and overall) according to this rubric:

Category	Score: 5	Score: 3	Score: 2	Score: 1
<i>Formatting</i>	consistent and helpful whitespace, indentation, and syntax throughout	mostly consistent whitespace, indentation, and syntax	inconsistent whitespace, indentation, and syntax	non-existent whitespace, inconsistent indentation or syntax
<i>Code Style</i>	clean and understandable code structures, insightful documentation throughout, very descriptive variable names	mostly clean and understandable code structures, some useful documentation, mostly-descriptive variable names	inappropriate or confusing code structures, useless documentation, some poor variable names	inappropriate and confusing code structures, useless or non-existent documentation, many poor variable names
<i>Elegance</i>	clean and efficient algorithms, easy-to-follow control flow, clear code dependencies	mostly clean algorithms, mostly-apparent control flow, mostly-clear code dependencies	unclear or inefficient algorithms, hard-to-follow control flow, mostly-unclear code dependencies	broken algorithms, impossible-to-follow control flow, unclear code dependencies
Overall	<i>model solution</i>	<i>good solution</i>	<i>ok, but flawed</i>	<i>unacceptable</i>

Submit your code review on Canvas, although you may wish to prepare your comments using a text editor or word processor beforehand to reduce the chance of data loss due to browser issues. Make sure you submit each review to the correct author!

Keep in mind that your review will also be compared with another person's review and my own assessment. In addition, your code reviews will be graded on the following scale:

Score	Description
5	Thorough, constructive criticism on all points with consistent numerical scores.
4	Constructive feedback for most points with consistent numerical scores.
3	Mixture of constructive and non-constructive (or non-existent) written feedback.
2	Mostly non-constructive written feedback; alternatively, a major discrepancy between written feedback and numerical scores.
1	No written feedback (only numeric scores).