LEVEL	1	2	3	4
names	names appear unreadable, meaningless or misleading	names accurately describe the intent of the code, but can be incomplete, lengthy, misspelled or inconsistent use of casing	names accurately describe the intent of the code, and are complete, distinctive, concise, correctly spelled and consistent use of casing	all names in the program use a consistent vocabulary
headers	headers are generally missing or descriptions are redundant or obsolete; use mixed languages or are misspelled	header comments are generally present; summarize the goal of parts of the program and how to use those; but may be somewhat inaccurate or incomplete	header comments are generally present; accurately summarize the role of parts of the program and how to use those; but may still be wordy	header comments are generally present; contain only essential explanations, information and references
comments	comments are generally missing, redundant or obsolete; use mixed languages or are misspelled	comments explain code and potential problems, but may be wordy	comments explain code and potential problems, are concise	comments are only present where strictly needed
layout	old commented out code is present or lines are generally too long to read	positioning of elements within source files is not optimized for readability	positioning of elements within source files is optimized for readability	positioning of elements is consistent between files and in line with platform conventions
formatting	formatting is missing or misleading	indentation, line breaks, spacing and brackets highlight the intended structure but erratically	indentation, line breaks, spacing and brackets consistently highlight the intended structure	formatting makes similar parts of code clearly identifiable
flow	there is deep nesting; code performs more than one task per line; unreachable code is present	flow is complex or contains many exceptions or jumps; parts of code are duplicate	flow is simple and contains few exceptions or jumps; duplication is very limited	in the case of exceptions or jumps, the most common path through the code is clearly visible
idiom	control structures are customized in a misleading way	choice of control structures is inapproriate	choice of control structures is appropriate; reuse of library functionality may be limited	reuse of library functionality and generic data structures where possible
expressions	expressions are repeated or contain unnamed constants	expressions are complex or long; data types are inappropriate	expressions are simple; data types are appropriate	expressions are all essential for control flow
decomposition	most code is in one or a few big routines; variables are reused for different purposes	most routines are limited in length but mix tasks; routines share many variables instead of having parameters	routines perform a limited set of tasks divided into parts; use of shared variables is limited	routines perform a very limited set of tasks and the number of parameters and shared variables is limited
modularization	most code is in one or a few large modules; or modules are artificially separated	modules have mixed responsibilities, contain many variables or contain many routines	modules have clearly defined responsibilities, contain few variables and a somewhat limited amount of routines	modules are defined such that communication between them is limited

- for each criterion, circle the level that is most representative of the features that are present
- no need to circle a level that is not relevant to the assignment
- level 2 implies that the features in level 1 are not present, level 4 implies that the features in level 3 are also present