LEVEL	1	2	3	4
	Documentation - is the code well-annotated to ensure rapid understanding?			
names	names appear unreadable, meaningless or misleading	names accurately describe the intent of the code, but can be incomplete, fuzzy, lengthy, misspelled	names accurately describe the intent of the code, and are complete, distinctive, concise, correctly spelled	all names in the program use a consistent vocabulary
headers	headers are missing or descriptions are redundant or obsolete; or use mixed languages	headers summarize the goal of parts of the program and how to use those, but may be incomplete or misspelled	headers accurately summarize the role of parts of the program and how to use those, are spelled correctly, may be wordy	headers contain only essential explanations, information and references
comments	comments are generally missing, redundant or obsolete; or use mixed languages	comments highlight important decisions and potential problems, but may be wordy or misspelled	comments highlight important decisions and potential problems, are concise and spelled correctly	comments are only present where strictly needed
	Presentation - is the code visually organized for a quick read?			
layout	old code is present	arrangement of code within source files is not optimized for readability	arrangement of code within source files is optimized for readability	arrangement of code is consistent between files
formatting	formatting is missing or misleading or lines are too long to read	indentation, line breaks, spacing and brackets highlight the intended structure but erratically	indentation, line breaks, spacing and brackets fully highlight the intended structure	formatting makes differences and similarities clearly visible
	Algorithms - is each part of the code as simple as possible?			
flow	there is deep nesting; code performs more than one task per line; control structures are customized in a misleading way	flow is complex or contains many exceptions; choice of control structures and libraries is inappropriate	flow is simple and contains few exceptions; choice of control structures and libraries is appropriate	flow prominently features the expected path
expressions	expressions are repeated or contain unnamed constants	expressions are complex; data types are inappropriate	expressions are simple; data types are appropriate	expressions are all essential for control flow
	Structure - is the code organized for quick understanding of parts and the whole?			
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; parts of code are repeated	routines perform a limited set of tasks divided into parts; shared variables are limited; code is unique	routines perform a very limited set of tasks and the number of parameters and shared variables is limited
modularization	modules are artificially separated	modules have vague subjects, contain many variables or contain many routines	modules have clearly defined subjects, contain few variables and a limited amount of routines	modules are defined such that communication between them is limited

- highlight features from all levels that are present in the code, starting at the lowest
- 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

Level 1: problematic features are present
Level 2: core quality goals not yet achieved
Level 3: core quality goals achieved
Level 4: achievement beyond core quality goals