

Area of evaluation	1	2	3	4	5
Testing	Tests exist, but not unit tests	Unit tests exist, but fail	Unit tests exist and pass, but don't cover expected scenarios	Unit tests exist and pass, cover expected scenarios	Unit tests exist and pass, cover expected scenarios, and/or are exceptional in some way (e.g. also covering edge cases)
Logical separation	No logical separation exists	An attempt made to logically separate out a concept, but it was incomplete or incorrect	At least one logical concept was properly separated out, but others were grouped together in an unideal way	Multiple logical concepts were properly separated out in a meaningful and effective way	All expected logical concepts are separated effectively and particular attention or efforts are made to address extensibility or scalability in a novel way.
Code organization	No attempt made to organize code logically	An attempt made to organize code logically but does not conform to well accepted standards, or is a failed implementation	An attempt was made to organize the code properly within a single file, without an effort to break out code which should live elsewhere (e.g. tests, for most languages)	Code is organized appropriately both within each file and as a project itself	Code is organized appropriately both within each file and as a project itself and conforms to accepted standards/norms for the language
Code quality	Code doesn't run/compile due to issues with the code itself or with instructions provided	Code runs/compiles but numerous obvious bugs exist which may result in unexpected behavior	Code runs/compiles but subtle logic errors exist which may result in unexpected behavior	Code runs/compiles and is absent of any bugs/errors related to logic, behaves as expected.	Code runs/compiles and is absent of any bugs/errors related to logic or otherwise.
Code cleanliness / readability	Code is completely unreadable, very difficult to tell what it's trying to do	Code is somewhat readable but unclear or confusing to understand	Code is easily readable, but contains redundant code, and/or code which is never executed	Code is readable, names of objects/methods and other components make sense	Code is exceptionally clean and readable, self-documenting, and/or contains meaningful documentation to assist with setup and execution