5 Star Scale

5 Star Scale

- Used to grade a program

- Correctness/Efficiency: _____/ 5 x 2 = _____/ 10

- Documentation: _____/ 5

- Structure/Complexity: _____/ 5

- Total: _____/ 20 (this ratio will be used to determine the applicable point score)
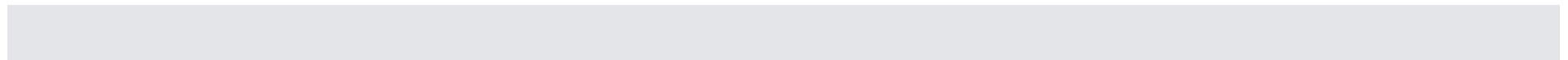
Zip Files

- When submitting source code, be sure to submit <u>all</u> files required to run your program, unless indicated otherwise in the lab manual

- Files must be compressed using the <u>zip</u> format; no other formats are accepted

- Files that cannot be viewed (e.g., due to a corrupt zip file or a non-zip format) will result in a 0 grade for the submission

- GitHub provides a reliable way to download a repository in zip format, but make sure the GitHub version is up-to-date

5 Stars

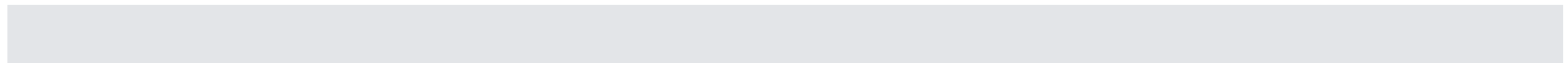| Correctness/ Efficiency	Documentation	Structure/Complexity
- passes all tests	- well-documented, allowing another programmer to use all functions based on the header comments alone	- well-engineered, consisting of a modular collection of simple, single-purpose functions
- code review reveals no faults	- responsibilities of all functions are described well, without giving implementation details	- constants are used whenever appropriate
- efficient (given the requirements)	- all parameters, return values, and side effects are explained	- globals are not used, unless unavoidable
- no redundant operations	- comments within all functions are helpful, without being distracting, making it easy to follow along	- all constants and variables are named appropriately
		- no layout abnormalities (eg, missing or improper indentation)
		- easily used and reused
		- no undesirable side effects, such as debug output

4 Stars

Correctness/ Efficiency	Documentation	Structure/Complexity
- passes nearly all tests	- occasionally, there are comments that are not complete, helpful, and/or true	- largely well-engineered, except for a few, minor issues
- a code review reveals nearly no faults; faults that are found, are minor	- could be improved by slightly reworded, slightly more, or slightly fewer comments	- a few, minor issues with constants, variables, or layout
- generally efficient, except in a few minor cases		- easily used and reused, except in a few, minor instances
- at most a few redundancies		- generally no undesirable side effects

3 Stars

Correctness/ Efficiency	Documentation	Structure/Complexity
- passes half the tests, or more, but the failure rate is too high for a 4-star rating	- in a number of cases, comments are cryptic, false, incomplete, misleading, missing, or redundant	- in need of reengineering due to a number of issues, none, or almost none of which, are major
- a code review reveals a number of faults, but none, or almost none of them, are major		- on a number of occasions, there are issues with constants, variables, or layout
- somewhat efficient, but there are a small number of major inefficiencies		- often easily used and reused, but there are a small number of major problems, none of which render the work unusable
- potentially many redundancies		

2 Stars

Correctness/ Efficiency	Documentation	Structure/Complexity
- fails more than half the tests	- only little relevant documentation	- a number of major issues, requiring major changes
- a code review reveals a high number of faults, including a number of major faults	- comments are often cryptic, false, incomplete, misleading, missing, or redundant	- not easily used and reused
- not efficient, although slow progress is being made		

1 Star

Correctness/ Efficiency	Documentation	Structure/Complexity
- fails nearly all the tests	- essentially no legitimate documentation	- only a few functions, many of which are responsible for too many things
- code review reveals a proliferation of major faults		- essentially not usable or reusable

0 Star

Correctness/Efficiency	Documentation	Structure/Complexity
- fails all tests and/or does not run	- no legitimate documentation and/or does not run	- no legitimate code and/or does not run
- source cannot be viewed	- source cannot be viewed	- source cannot be viewed