EECS 381 Example Code Quality Check List - C version

Explanation and guidance for some items is shown in italics

Student:_____ **AG score:**_____

General code quality

_____ (2) Appropriate commenting.

_____ Function prototypes first, functions in readable order

_____ Main function is very small

- represents top level of program process, calling functions that do the work.

_____ Program has well-chosen subfunctions to organize the code

_____ Program lacks redundant or awkward code.

- it could have been eliminated, or rewritten to make it simpler and clearer.

_____ Code is not duplicated excessively.

- don't code identical functionality with copy/paste! - write a function instead!

_____ Code reads well, and is not excessively verbose or convoluted.

_____ Good variable/symbol naming

_____ Code follows recommended C style practices (e.g. NULL, typedefs, macro names)

_____ Code uses C idioms

- testing with if(flag) instead of if(flag != 0), writing loops like for(i = 0; i < len; i++) when possible.

_____ (2) Program lacks egregious or gross inefficiency.

- egregious inefficiency: inefficiency without compensating quality improvement in return.

- gross inefficiency: extreme inefficiency that could have been easily avoided.

Specific code quality - differs depending on project

_____ Main function built around a switch statement that calls command-specific functions

_____ Format for scanf of strings disallows overrun of array

_____ No unnecessary memory allocations (e.g. for temporary buffers - local arrays used instead)

_____ Memory allocated and deallocated in responsible module's functions only

_____ Return value from malloc checked and program terminated if 0

_____ Memory for strings allocated to fit the data

_____ No apparent memory leaks.

_____ All allocated memory freed upon termination.

_____ Globals are used, and declared and defined in .h, .c files, following course guidelines.

_____ Globals are modifed only by appropriately responsible module

_____ Good choice of functions in Utility module (e.g. a string allocator/deallocator)

- functions used in more than one module, or could be used in very different projects.

_____ Standard Library facilities used appropriately (no reinvention of wheels)

Other Attributes

_____ Additional positive qualities:

_____ Additional negative qualities:

- Points deducted if serious failure to apply course content or project specifications.

E.g. On an important topic, the project looks like you aren't even in the course,

or you completely missed the point, or blew off the project goals.

_____ **Total**