DS2000 Homework & Quizzes Rubric

This rubric will be used to evaluate your homework assignments in DS2000. It will be applied to each homework you submit (most homework assignments have multiple Python programs). In addition to numeric scores in each rubric category, your grader will provide written feedback, if there is something helpful we can contribute to your learning.

Specific point values will vary per-assignment, but this outline will give you the general idea and regular weight break down. We may make small adjustments from assignment to assignment, but you can count on weights falling within 5% of what is outlined below.

Category	Weight	Excellent	Satisfactory	Unsatisfactory	Not Met
Program Correctness Is the program doing the right thing?	~60%	No errors, program always works correctly and meets the specifications.	Minor details of the program specification are not met, program functions incorrectly for some inputs.	Significant details of the specification are not met, program often exhibits incorrect behavior.	Program does not run successfully for any inputs, or not submitted.
Program Design Is the program structured so that code is appropriately reusable and program flow is easy to flow? ~30%	~30%	Program flow is easy to follow. Lines of code that complete a certain task are grouped together and it's easy to tell what they do. Code is modular. Functions are used where appropriate. Parameters are chosen to make functions flexible. No "do everything" functions.	Program flow could be improved with some substantive re-organization. Groupings of lines of code don't always make semantic sense. One or two functions do too much or too little, would be improved with different parameters/returns.	Program is difficult to follow. Tasks might be interwoven in unclear ways. Functions are strung together such that to understand the general idea of the program as a whole, an outside reader has to read all of the code.	Program has no overarching flow. No functions used when required/appropriate.
Style & Documentation Can someone else look at your code and understand at a high level the intent of your program?	~10%	Code is easy to understand from an outside perspective. - Variable and function names are clear and concise. - Comments throughout are clear and appropriate.	Minor issues. - Variable/function names are good in general but there are at least two places where we have to be investigators to figure out what they mean/do. - Two or more places that would benefit from comments are missing them **OR** the code is overly commented.	Major issues. Variable/function names are not clear. Code is mostly not commented.	Goal not met. Variable/function names and organization not clear to the point of actively obscuring the code. No elements of the style guide incorporated. No comments.