Grading Rubric for Programming Assignments

Criteria	Exceptional 90-100%	Acceptable 80-89%	Amateur 70-79%	Unsatisfactory 0-69%	Grade / Comments:
Specifications	The program works and meets all of the specifications.	The program works and produces the correct results and displays them correctly. It also meets most of the other specifications.	The program produces correct results but does not display them correctly.	The program is producing incorrect results.	
Readability	The code is exceptionally well organized and very easy to follow. A uniform coding "standard" is followed throughout.	The code is fairly easy to read.	The code is readable only by someone who knows what it is supposed to be doing.	The code is poorly organized and very difficult to read. Variable and procedure names are ambiguous,	
Reusability	The code could be reused as a whole or each routine could be reused.	Most of the code could be reused in other programs.	Some parts of the code could be reused in other programs.	The code is not organized for reusability.	
Documentation	The documentation is well written and clearly explains what the code is accomplishing and how. A uniform documentation "standard" is followed throughout.	The documentation consists of embedded comment and some simple header documentation that is somewhat useful in understanding the code.	The documentation is simply comments embedded in the code with some simple header comments separating routines.	The documentation is simply comments embedded in the code and does not help the reader understand the code.	
Delivery	The program was delivered on time.	The program was delivered within a day of the due date.	The code was within 2 days of the due date.	The code was more than 2 days overdue.	
Efficiency	The code is extremely efficient without sacrificing readability and understanding.	The code is fairly efficient without sacrificing readability and understanding.	The code is brute force and unnecessarily long. The same code is repeated.	The code is huge and appears to be patched together.	
Overall Programming Style	Proper use of: modularization, use of syntax (e.g. if vs. select case), loops (for vs. while), variable names, function names, declaration, indentation, initialization, type casting, parameter passing, pass by value vs. pass by reference, global variables	The program exhibits 70% or more of the list shown (to the left)	The program exhibits less than 20% to 69% of the list shown (to the left)	Little or no attempt to use proper: modularization, use of syntax (e.g. if vs. select case), loops (for vs. while), variable names, function names, declaration, indentation, initialization, type casting, pass by value vs. pass by reference, global variables	

Adapted from Cal State Long Beach:
http://www.csulb.edu/colleges/coe/cecs/views/programs/undergrad/grade_prog.shtml