Programming Assignments Grading Rubric

	Unsatisfactory (≤ 40% of the points)	Satisfactory (60% of the points)	Good (80% of the points)	Excellent (100% of the points)
Requirements and Delivery **(50 points)**	• Completed less than 70% of the requirements. • Delivered on time but not in correct format.	• Completed between 70-80% of the requirements. • Delivered on time, and in correct format.	• Completed between 80-90% of the requirements. • Delivered on time, and in correct format.	• Completed between 90-100% of the requirements. • Delivered on time, and in correct format.
Coding Standards **(10 points)**	• No name, date, or assignment title included • Poor use of white space (indentation, blank lines). • Disorganized and messy • Poor use of variables (many global variables, ambiguous naming).	• Includes name, date, and assignment title. • White space makes program fairly easy to read. • Organized work. • Good use of variables (few global variables, unambiguous naming).	• Includes name, date, and assignment title. • Good use of white space. • Organized work. • Good use of variables (no global variables, unambiguous naming)	• Includes name, date, and assignment title. • Excellent use of white space. • Creatively organized work. • Excellent use of variables (no global variables, unambiguous naming).
Documentation **(10 points)**	• Very limited or no documentation included. • Documentation does not help the reader understand the code.	• Basic documentation has been completed including descriptions of all class variables. • Purpose is noted for each function.	• Clearly documented including descriptions of all class variables. • Specific purpose is noted for each function and control structure.	• Clearly and effectively documented including descriptions of all class variables. • Specific purpose noted for each function, control structure, input requirements, and output results.
Runtime **(20 points)**	• Does not execute due to errors. • User prompts are misleading or non-existent. • No testing has been completed, or no input validation.	• Executes without errors. • User prompts contain little information, poor design. • Some testing or input validation has been completed.	• Executes without errors. • User prompts are understandable, minimum use of symbols or spacing in output. • Most testing or input validation completed.	• Executes without errors excellent user prompts, good use of symbols, spacing in output. • Thorough and organized testing or input validation has been completed.
Efficiency **(10 points)**	• A difficult to understand and inefficient solution. Code is huge and appears to be patched together.	• A logical solution that is easy to follow but it is not the most efficient.	• The code is fairly efficient without sacrificing readability and understanding.	• Solution is efficient, easy to understand, and maintain.