This document describes the grading criteria and rubric for all programming assignments in the course. If you have any questions, please feel free to post your questions to the Virtual Office on discussion boards.

Grading Criteria

Correctness

The program should function correctly, including the following aspects:

- Meet the assignment specifications
- Have no logic errors, and produce correct output for a variety of input

Efficiency

Given a problem, there may exist different approaches to solving the problem and writing the program accordingly. In programming assignments, you should

- Make a smart choice on the data structure and algorithm in order to solve the given problem efficiently. For example, if you are asked to do a search on a sorted list, binary search should be chosen instead of sequential search.
- Write code in an efficient way, such as avoiding hard-coding, and using functions and loops appropriately to avoid unnecessary redundancy.

Readability

The program should be clean, readable and well-organized, including the following aspects:

- Use consistent indentation style, use whitespace when appropriate to separate code blocks, and keep formatting style consistent
- Use meaningful variable/class/function names, and keep naming scheme consistent
- Well-organize code into functions or classes when appropriate

Documentation

The program should be well-commented by

- Providing meaningful comments to major classes, functions, variables and statements to illustrate program logic
- Avoiding obvious and meaningless comments

Grading Rubric

Correctness	**60 points**	**59 ~ 48 points**	**47 ~ 36 points**	**35 ~ 1 points**	**0 point**
	Meet all assignment specifications, with no logic errors.	Meet all assignment specifications, but have minor logic errors.	Miss a few minor details of the assignment specifications, and have minor logic errors.	Miss major assignment specifications, or have major logic errors.	Miss major assignment specifications, and have major logic errors.
Efficiency	**15 points**	**14 ~ 12 points**	**11 ~ 9 points**	**8 ~ 1 points**	**0 point**
	The chosen data structure and algorithms can efficiently solve the given problem. The entire program is written in an efficient way.	The chosen data structure and algorithms can efficiently solve the given problem. There are a few places in the program that can be written in a more efficient way.	The chosen data structure and algorithms are not the efficient ones to solve the given problem. There are a few places in the program that can be written in a more efficient way.	The chosen data structure and algorithms can efficiently solve the given problem. Many places in the program can be obviously written in a more efficient way.	The chosen data structure and algorithms are not the efficient ones to solve the given problem. Many places in the program can be obviously written in a more efficient way.
Readability	**15 points**	**14 ~ 12 points**	**11 ~ 9 points**	**8 ~ 1 points**	**0 point**
	No issues with all aspects of readability. The program is clean, readable and well-organized.	Minor issues with three aspects of readability.	Major issues with one aspect of readability.	Major issues with two aspects of readability.	Major issues with all three aspects of readability. The program is poorly organized and difficulty to read.
Documentation	**10 points**	**8 ~ 9 points**	**7 ~ 6 points**	**5 ~ 1 points**	**0 point**
	The program is well-documented.	All major classes, functions, variables and statements have meaningful comments. However, the program is overly commented or has some meaningless comments.	Comments for a few places, such as classes and functions, are missing.	Major lack of meaningful comments that makes the program difficult to understand.	No comments.