

CS 247 Spring 2014
Assignment 2
Design and Programming-Style Marking Rubric

Q1: [50 marks (out of 70)] - Correctness (automarked)

Q1: [20 marks (out of 70)] - Design and Programming Style (subtractive marking scheme)

- 50 If the implementation is not hidden (that's right, -50 marks for not answering the question)
- 20 poor organization of code: data and operations should be packaged into classes / structs such that functions are associated with the data that they manipulate.
- 10 "Don't repeat yourself": Code that is at least 2 statements long, implements a coherent function, and appears in more than one place should be packaged as a helper function.
- 10 Variables or functions that are unnecessarily global (e.g., namespaces)
- 10 Poor or no documentation of routines, variables (unless self-documented)
- 10 Ineffective or incorrect use of dynamic memory allocation
- 5 Not correctly setting `ios::failbit` in `operator>>`
- 5 Not adhering to header file conventions (i.e., guard, no (namespace) using directives)
- 5 Missing constants: use mnemonic constants, `const` parameters, `const` member functions
- 5 Poor variable names (not self-documenting)
- 5 Poor or no indenting

Q2: [50 marks (out of 80)] - Correctness (automarked)

Q2: [5 marks (out of 80)] - Makefile

- 5 Compilation is not incremental

Q2: [25 marks (out of 80)] - Design and Programming Style (subtractive marking scheme)

- 20 poor organization of code: data and operations should be packaged into classes / structs and namespaces such that functions are associated with the data that they manipulate.
- 15 No or weak use of exceptions (e.g., not using exceptions to separate code that deals with bad input, not using exception classes)
- 10 "Don't repeat yourself": Code that is at least 2 statements long, implements a coherent function, and appears in more than one place should be packaged as a helper function.
- 10 Information hiding: all data members should be private; variables or functions that are unnecessarily global (e.g., namespaces)
- 10 Poor or no documentation of routines, variables (unless self-documented)
- 10 Ineffective or incorrect use of dynamic memory allocation
- 5 Not adhering to header file conventions (i.e., guard, no (namespace) using directives)
- 5 Missing constants: use mnemonic constants, `const` parameters, `const` member functions
- 5 Poor variable names (not self-documenting)
- 5 Poor or no indenting