

CS 247 Spring 2014

Assignment 3

Design and Programming-Style Marking Rubric

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

- 10 marks - Description of the problem. The description should be written in a way that the whole class would understand the problem to be implemented
- 10 marks - UML showing application of design pattern
- 30 marks - C++ implementation conforms to design pattern
- 15 marks - Bonus for doing decorator, iterator, factory pattern

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

- 20 marks for poor organization/modularity of code:
 - data and operations should be packaged into classes / structs such that functions are associated with the data that they manipulate.
 - declarations and definitions should be appropriately separated into header and implementation files
- 10 marks for poor information hiding (e.g., data members must be private, variables and functions should not be unnecessarily global, use of namespaces)
- 10 marks for repetitive code: 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 marks for ineffective or incorrect use of dynamic memory allocation
- 10 marks for poor or no documentation of routines, variables (unless self-documented)
- 5 marks for not adhering to header file conventions (i.e., guard, no using directives)
- 5 marks for missing constants: use mnemonic constants, const parameters, const member functions
- 5 marks for poor variable names (not self-documenting)
- 5 marks for poor or no indenting

Q3: [40 marks (out of 60)] - Correctness -- Automarked

Q3: [20 marks (out of 60)] - Design and Programming Style (subtractive marking scheme)

- 20 marks for not using STL algorithms and containers
- 10 marks for poor information hiding (e.g., data members must be private, variables and functions should not be unnecessarily global, use of namespaces)
- 10 marks for repetitive code: 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 marks for ineffective or incorrect use of dynamic memory allocation
- 10 marks for poor or no documentation of routines, variables (unless self-documented)
- 5 marks for not adhering to header file conventions (i.e., guard, no using directives)
- 5 marks for missing constants: use mnemonic constants, const parameters, const member functions
- 5 marks for poor variable names (not self-documenting)
- 5 marks for poor or no indenting