Observations Report

Drew Cabales

Challenge: Technical Review

National Weather Service/OHD
Science Infusion and Software Engineering Process Group (SISEPG) – C++ Coding Standards and Guidelines Peer Review Checklist

C++ Coding Standards and Guidelines Peer Review Checklist

Last Updated: 25 April 2016

Reviewer's Name:	Peer Review Date:	
Project Name:	Project ID: Enter if applicable	
Developer's Name:	Project Lead:	
Review Files & Source code		
Code Approved		

The following check list is to be used in the assessment of C++ source code during a peer review. Items which represent the code being reviewed should be checked.

1. General Programming Standards and Guidelines

Refer to the OHD General Programming Standards and Guidelines Peer Review Checklist to assess the adherence to the OHD General Programming Standards and Guidelines.

2. C++ Programming Standards

2.1	Readability and Maintainability
/	_Consistent indentation (3 or 4 spaces)
\checkmark	_Consistent use of braces
$_{\times}$	_No tabs used
2.2	File Names
_/	_Header files and namespace files use suffixes: .h, .H, .hh, .hpp, or .hxx
	Source files use suffixes: .C, .cc, .cpp, or .cxx
_<	_UpperMixedCase is used for class or namespace file names
×	lowerMixedCase is used for function file names

Private class data member names are prepended with the underscore, the rest is

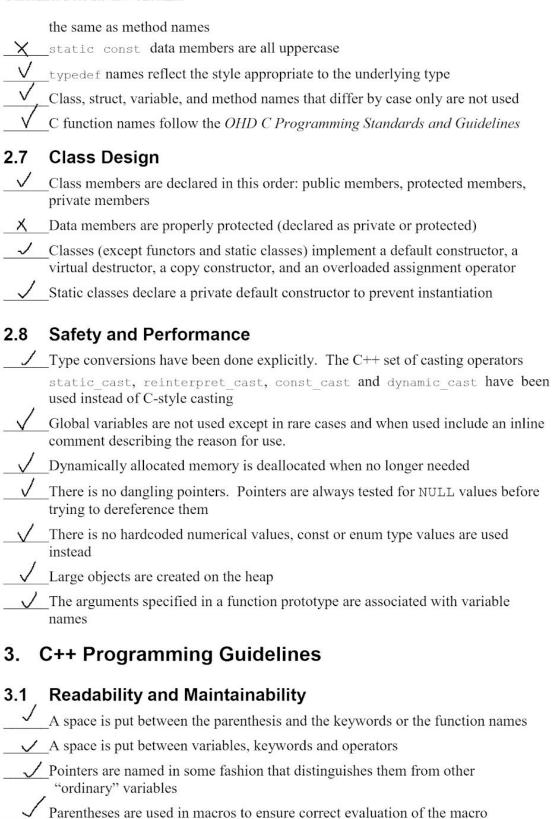
Science Infusion and Software Engineering Process Group (SISEPG) - C++ Coding Standards and

National Weather Service/OHD

2 11/16/2006 Version 1.9

National Weather Service/OHD

Science Infusion and Software Engineering Process Group (SISEPG) – C++ Coding Standards and Guidelines Peer Review Checklist



3

11/16/2006 Version 1.9

Science	al Weather Service/OHD Infusion and Software Engineering Process Group (SISEPG) – C++ Coding Standards and nes Peer Review Checklist
	The goto statement is used very sparingly
3.2	User Defined Types
	_static const members are used instead of #defined constants
/	Proper typedefs are used instead of using templates directly
	_enum is used to define a collection of integral constants
3.3	Variables
/	_const correctness has been practiced
	_All variables are correctly initialized
	Local variables are declared near their first use.
/	The copy constructor is used to construct an object instead of the assignment operator (=)
3.4	Performance
$\sqrt{}$	_inline functions are used instead of Macros
/	The prefix form (++i ori) is used instead of postfix form (i++ or i)
/_	Pointer arithmetic has been avoided
	Repetitive computations are reduced by only doing them once and saving the result in a temporary variable for future access
3.5	Class Design
	Parts-of relation inheritance has been avoided

Questions:

- In the Code File of the challenge, there was a potential for a major memory leak in the "UberNode". What was it? UberNode's deconstructor isn't called when it's deleted
- 2. Static Testing encapsulates a couple software testing methods that include Static Analysis and Code Reviews. Static testing is automated testing that checks for code validation without executing the code. Static Testing also covers Code Reviews, which is done by software developers and engineers to find what errors are occurring, and how often. They look into potential mitigations for their vulnerabilities and record what errors are common within their codebase.
- 3. We can write tests that can conclude there's a chance that a specific section of code has errors based on common design patterns and implementations of classes and libraries, but to prove that it's correct or not is too complex for an automated system to detect.

I learned about the core differences between static and dynamic testing, and how both are important to produce clean, efficient code that will hopefully include minimal bugs. Since there's a long list of well known exploits that are common amongst many companies' code in the past and present, these static analysis tools are important in recognizing flaws that software engineers might not have knowledge of. There's a limit to automated testing, however it proves to be a good defense measure after human analysis and the code compiler.