

Clang

 This page was automatically generated and should not be edited.

 The information on this page was provided by outside contributors and has not been verified by SEI CERT.

 The table below can be re-ordered, by clicking column headers.

Tool Version: 3.9

Checker	Guideline
-analyzer-checker=cplusplus	EXP51-CPP. Do not delete an array through a pointer of the incorrect type
-Wdangling-initializer-list	EXP54-CPP. Do not access an object outside of its lifetime
-Wdelete-incomplete	EXP57-CPP. Do not cast or delete pointers to incomplete classes
-Wdelete-non-virtual-dtor	OOP52-CPP. Do not delete a polymorphic object without a virtual destructor
-Wdynamic-class-memaccess	EXP60-CPP. Do not pass a nonstandard-layout type object across execution boundaries
-Wexceptions	ERR53-CPP. Do not reference base classes or class data members in a constructor or destructor function-try-block handler
-Wexceptions	ERR54-CPP. Catch handlers should order their parameter types from most derived to least derived
-Winvalid-noreturn	MSC53-CPP. Do not return from a function declared <code>[[noreturn]]</code>
-Winvalid-offsetof	EXP59-CPP. Use <code>offsetof()</code> on valid types and members
-Wmismatched-new-delete	MEM51-CPP. Properly deallocate dynamically allocated resources
-Wreorder	OOP53-CPP. Write constructor member initializers in the canonical order
-Wreserved-id-macro	DCL51-CPP. Do not declare or define a reserved identifier
-Wreturn-type	MSC52-CPP. Value-returning functions must return a value from all exit paths
-Wunevaluated-expression	EXP52-CPP. Do not rely on side effects in unevaluated operands
-Wuninitialized	EXP53-CPP. Do not read uninitialized memory
-Wunsequenced	EXP50-CPP. Do not depend on the order of evaluation for side effects
-Wuser-defined-literals	DCL51-CPP. Do not declare or define a reserved identifier
-Wvarargs	EXP58-CPP. Pass an object of the correct type to <code>va_start</code>
-Wvexing-parse	DCL53-CPP. Do not write syntactically ambiguous declarations
cert-dcl50-cpp	DCL50-CPP. Do not define a C-style variadic function
cert-dcl59-cpp	DCL59-CPP. Do not define an unnamed namespace in a header file
cert-err34-c	ERR62-CPP. Detect errors when converting a string to a number
cert-err52-cpp	ERR52-CPP. Do not use <code>setjmp()</code> or <code>longjmp()</code>
cert-err58-cpp	ERR58-CPP. Handle all exceptions thrown before <code>main()</code> begins executing
cert-err60-cpp	ERR60-CPP. Exception objects must be nothrow copy constructible
cert-err61-cpp	ERR61-CPP. Catch exceptions by lvalue reference
cert-msc50-cpp	MSC50-CPP. Do not use <code>std::rand()</code> for generating pseudorandom numbers
cert-oop54-cpp	OOP54-CPP. Gracefully handle self-copy assignment
Clang checks for violations of this rule and produces an error without the need to specify any special flags or options.	DCL52-CPP. Never qualify a reference type with <code>const</code> or <code>volatile</code>
clang-analyzer-alpha.cplusplus.VirtualCall	OOP50-CPP. Do not invoke virtual functions from constructors or destructors
clang-analyzer-alpha.security.ArrayBoundV2	MEM50-CPP. Do not access freed memory
clang-analyzer-core.UndefinedBinaryOperatorResult	EXP53-CPP. Do not read uninitialized memory

clang-analyzer-cplusplus.NewDelete	MEM50-CPP. Do not access freed memory
clang-analyzer-cplusplus.NewDeleteLeaks	MEM51-CPP. Properly deallocate dynamically allocated resources
clang-analyzer-unix.MismatchedDeallocator	MEM51-CPP. Properly deallocate dynamically allocated resources
misc-new-delete-overloads	DCL54-CPP. Overload allocation and deallocation functions as a pair in the same scope