C++ static code analysis

Unique rules to find Bugs, Vulnerabilities, Security Hotspots, and Code Smells in your C++ code

All rules 578 | Vulnerability 13 | Bug 111 | Security Hotspot 18 | Code Smell 436 | Quick Fix 68

Tags

Search by name...

Unary prefix operators should not be repeated

Analyze your code

Bug Major

The needless repetition of an operator is usually a typo. After all, why write `!!!i` when `!i` will do?

On the other hand, the repetition of increment and decrement operators may have been done on purpose, but doing so obfuscates the meaning, and should be simplified.

This rule raises an issue for sequences of: `!`, `~`, `-`, and `+`, and in C++ for repetitions of the increment and decrement operators.

Noncompliant Code Example

```
int i = 1;

int j = - - -i;  // Noncompliant; just use -i
int k = ~~i;     // Noncompliant; same as i
int m = + +i;    // Noncompliant; operators are useless here

bool b = false;
bool c = !!!b;   // Noncompliant
```

Compliant Solution

```
int i =  1;

int j = -i;
int k =  i;
int m =  i;

bool b = false;
bool c = !b;
```

Exceptions

Boolean normalization `!!` is ignored.

Available In:

sonarlint | sonarcloud | sonarqube Developer Edition

Bug

"std::move" and "std::forward" should not be confused

Bug

A call to "wait()" on a "std::condition_variable" should have a condition

Bug

A pointer to a virtual base class shall only be cast to a pointer to a derived class by means of dynamic_cast

Bug

Functions with "noreturn" attribute should not return

Bug

RAII objects should not be temporary

Bug

"memcmp" should only be called with pointers to trivially copyable types with no padding

Bug

"memcpy", "memmove", and "memset" should only be called with pointers to trivially copyable types

Bug

"std::auto_ptr" should not be used

Bug

Destructors should be "noexcept"

Bug