rules.sonarsource.com

C++ static code analysis: The comma operator, "&&", and "||" should not be overloaded

2 minutes

Overloaded versions of the comma and logical conjunction operators have the semantics of function calls whose sequence point and ordering semantics are different from those of the built-in versions. It may not be clear at the point of use that these operators are overloaded, and so developers may be unaware which semantics apply.

Exception: Starting from *C++17*, the order of evaluation of the comma operator is defined and identical for the builtin and the overloaded versions. In such circumstances, the comma operator can safely be overloaded.

Noncompliant Code Example

```
#include "util.h"
class A
{
public:
  UtilType getValue ( );
  UtilType setValue ( UtilType const & );
};
void f1 ( A & a1, A & a2 )
```

```
{
  a1.getValue ( ) && a2.setValue ( 0 );    // Short circuiting
may occur
}
bool operator && ( UtilType const &, UtilType const & ); //
Noncompliant
void f2 ( A & a1, A & a2 )
{
  a1.getValue ( ) && a2.setValue ( 0 ); // Both operands
evaluated if type returned has overloaded operator&&
}
```

See

- MISRA C++ 2008, 5-2-11 - The comma operator, && operator and the || operator shall not be overloaded.