

"pthread_mutex_t" should be properly

"pthread_mutex_t" should not be consecutively locked or unlocked

initialized and destroyed

📆 Bug





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



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



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



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"

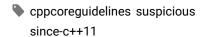
📆 Bug

"Forwarding references" parameters should be used only to forward parameters

Analyze your code

Code Smell





Forwarding references are a special kind of references that both ignore and preserve the value category of a function argument, making it possible to forward it by means of std::forward.

Any code using such a reference for any other purpose than forwarding is actually ignoring rvalue-ness and const-ness of the associated parameter.

Noncompliant Code Example

```
#include <utility>
#include <string>
#include <iostream>
template<typename TP> void f( TP&& arg ) {
    std::string s(arg);
}
int main() {
    std::string s("test");
    f(std::move(s));
    std::cout<<"f:"<<s<<std::endl; // output is "f:test"</pre>
    return 0;
}
```

Compliant Solution

```
#include <utility>
#include <string>
#include <iostream>
template<typename TP> void f( TP&& arg ) {
    std::string s(std::forward<TP>(arg));
}
int main() {
    std::string s("test");
    f(std::move(s));
    std::cout<<"f:"<<s<<std::endl; // output is "f:"
    return 0;
}
```

See

• C++ Core Guidelines F.19 - For "forward" parameters, pass by TP&& and only std::forward the parameter

Available In:

sonarlint ⊖ | sonarcloud ☆ | sonarqube Developer Edition

© 2008-2022 SonarSource S.A., Switzerland. All content is copyright protected. SONAR, SONARSOURCE, SONARLINT, SONARQUBE and SONARCLOUD are trademarks of SonarSource S.A. All other trademarks and copyrights are the property of their respective owners. All rights are expressly reserved. Privacy Policy