

C++ static code analysis: "nodiscard" attributes on functions should include explanations

1 minute

The `nodiscard` attribute can be used with or without explanations, but explaining why a result should not be discarded can only improve one's understanding of the code, and would prevent developers from wasting time figuring those things out by themselves.

This rule raises an issue when `nodiscard` is used on function without any explanation.

Noncompliant Code Example

```
[[nodiscard]] std::vector<int> generateRandomValues(int count); //  
Noncompliant  
  
generateRandomValues(100);
```

Compliant Solution

```
[[nodiscard("Computation of values is expensive")]] std::vector<int>  
generateRandomValues(int count);  
  
generateRandomValues(100);
```