C static code analysis

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

Methods should not be empty

Analyze your code

Code Smell   Critical   suspicious

There are several reasons for a method not to have a method body:

- It is an unintentional omission, and should be fixed to prevent an unexpected behavior in production.
- It is not yet, or never will be, supported. In this case an exception should be thrown in languages where that mechanism is available.
- The method is an intentionally-blank override. In this case a nested comment should explain the reason for the blank override.

Noncompliant Code Example

```
void fun(int p1) {
}
```

Compliant Solution

```
void fun(int p1) {
  int a = doSomething(p1);
  int threshold = 42;
  if (a > threshold) {
    // ...
  }
}
```

or

```
void fun(int p1) {
  // Intentionally unimplemented...
}
```

Exceptions

This rule doesn't raise an issue for empty class constructors or destructors. For instance this is the only way to define user-defined default constructors.

Available In:

sonarlint   sonarcloud   sonarqube Developer Edition

Stack allocated memory and non-owned memory should not be freed

Bug

Closed resources should not be accessed

Bug

Dynamically allocated memory should be released

Bug

Freed memory should not be used