rules.sonarsource.com

C++ static code analysis: Function declarations that look like variable declarations should not be used

1-2 minutes

Like a clever insect posing as a leaf, there are constructs in C++ which look like variable declarations, but which are actually interpreted by the compiler as function or function pointer declarations. Beyond the problem of confusing maintainers, it’s highly likely in such cases that what the coder intended is not what the compiler will do.

Noncompliant Code Example

```
void doWork(Status status) {
  Lock lock();  // Noncompliant; declares function named "lock"
  ...
  Form form(ProgressBar(status)); // Noncompliant; declares function named "form" with "status" parameter
  ...
}
```

Compliant Solution

```
void doWork(Status status) {
  Lock lock;  // remove the parentheses to declare a variable
  ...
  Form form((ProgressBar(status))); // add a pair of parentheses to declare a variable
  ...
}
```

Since C++11 you can also use direct initialization to declare a variable:

```
void doWork(Status status) {
  Lock lock{};
  ...
  Form form{ProgressBar{status}};
  ...
}
```

See

- CERT, DCL53-CPP. - Do not write syntactically ambiguous declarations