C++ static code analysis: Functions that are not used in a project should be removed

1-2 minutes

Unless you are in a library codebase context, functions that are declared in your program but never executed are dead code that should be removed. Cleaning out dead code decreases the size of the maintained codebase, making it easier to understand the program and preventing bugs from being introduced.

Note: {rule:cpp:S1144} is a subset of this rule; hence, it should be deactivated when this rule is activated.

Noncompliant Code Example

```
void unusedStaticFunction() { // Noncompliant
}

void unusedGlobalFunction() { // Noncompliant: it is not called in the entire program
}

void undefinedGlobalFunction(); // Noncompliant

class A {
public:
  A(int i) : i(i){} // Compliant it is called in "main"
  void startUsed() { // Compliant, the member function "startUsed" is called in "main"
  }
  void startUnused() { // Noncompliant, the member function "startUnused" is not called in the program
  }
private:
  A(): i(0) { // Compliant, private and deleted constructor are an exception
  }
  int i;
};

int main() // Compliant, "main" is an exception.
{
  A a{2};
  a.startUsed();
}
```