

# C++ static code analysis: Control flow statements "if", "for", "while", "switch" and "try" should not be nested too deeply

2 minutes

---

Nested `if`, `for`, `do`, `while`, `switch` and `try` statements is a key ingredient for making what's known as "Spaghetti code".

Such code is hard to read, refactor and therefore maintain.

## Noncompliant Code Example

With the default threshold of 3:

```
if (condition1) {                // Compliant; depth = 1
    /* ... */
    if (condition2) {            // Compliant; depth = 2
        /* ... */
        for(int i = 0; i < 10; i++) { // Compliant; depth = 3, not
exceeding the limit
            /* ... */
            if (condition4) {      // Noncompliant; depth = 4
                if (condition5) {  // Depth = 5, exceeding the
limit, but issues are only reported on depth = 4
                    /* ... */
                }
            }
        }
    }
}
```

```
        return;
    }
}
}
}
```

## Exceptions

Each use of a macro containing control flow statements is counted as one nesting level, even if the macro contains more than one control flow statement.

```
#define FOREACH(V,ARR) if(ARR!=nullptr) for(int V=0;
V<(sizeof(ARR)/sizeof(ARR[0])); V++)
```

```
if (condition1) {    // Compliant; depth = 1
    if (condition2) { // Compliant; depth = 2
        FOREACH(i, arr) { // Compliant; depth = 3 (not 4)
            if (condition3) { // Noncompliant; depth = 4
                /* ... */
            }
        }
    }
}
```