rules.sonarsource.com

C static code analysis: Switch cases should end with an unconditional "break" statement

3 minutes

When the execution is not explicitly terminated at the end of a switch case, it continues to execute the statements of the following case. While this is sometimes intentional, it often is a mistake which leads to unexpected behavior.

Noncompliant Code Example

```
switch (myVariable) {
  case 1:
    foo();
    break;
  case 2:  // Both 'doSomething()' and 'doSomethingElse()' will be executed. Is it on purpose ?
    doSomething();
  default:
    doSomethingElse();
    break;
}
```

Compliant Solution

```
switch (myVariable) {
  case 1:
    foo();
    break;
  case 2:
    doSomething();
    break;
  default:
    doSomethingElse();
    break;
}
```

Exceptions

This rule is relaxed in the following cases:

```
switch (myVariable) {
  case 0:                                // Empty case used to specify the
same behavior for a group of cases.
  case 1:
    doSomething();
    break;
  case 2:                                // Use of return statement
    return;
  case 3:                                // Use of throw statement
    throw 1;
  case 4:                                // Use of an attribute to make explicit
the fact that we want to fall through the next case
    doSomething();
    [[fallthrough]];
  case 5:                                // Use of continue statement, if the
switch is inside a loop
    continue;
  default:                               // For the last case, use of break
statement is optional
    doSomethingElse();
}
```

See

- MISRA C:2004, 15.0 - The MISRA C *switch* syntax shall be used.
- MISRA C:2004, 15.2 - An unconditional break statement shall terminate every non-empty switch clause
- MISRA C++:2008, 6-4-3 - A switch statement shall be a well-formed switch statement.
- MISRA C++:2008, 6-4-5 - An unconditional throw or break statement shall terminate every non-empty switch-clause
- MISRA C:2012, 16.1 - All switch statements shall be well-formed

- MISRA C:2012, 16.3 - An unconditional break statement shall terminate every switch-clause
- MITRE, CWE-484 - Omitted Break Statement in Switch
- CERT, MSC17-C. - Finish every set of statements associated with a case label with a break statement
- CERT, MSC52-J. - Finish every set of statements associated with a case label with a break statement