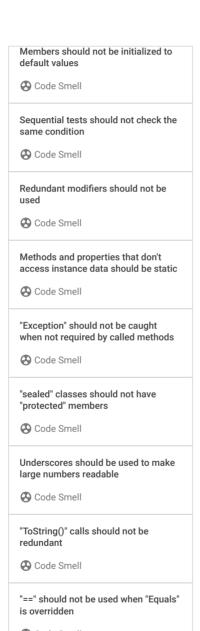


C# static code analysis Unique rules to find Bugs, Vulnerabilities, Security Hotspots, and Code Smells in your C# code Security O Quick 52 Fix (28) **₩** Bug (76) Hotspot



An abstract class should have both

abstract and concrete methods

Multiple variables should not be

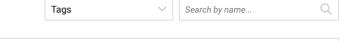
Culture should be specified for "string"

declared on the same line

Code Smell

Code Smell

operations



```
Integral numbers should not be
shifted by zero or more than their
number of bits-1
```

Analyze your code

👬 Bug 🕚 Minor 🕜

Shifting an integral number by 0 is equivalent to doing nothing but makes the code

If the first operand is an int or uint (32-bit quantity), the shift count is given by the low-order five bits of the second operand. That is, the actual shift count is 0 to 31

Note that integral number with a less than 32-bit quantity (e.g. short, ushort...) are implicitly converted to int before the shifting operation and so the rule for

If the first operand is a long or ulong (64-bit quantity), the shift count is given by the low-order six bits of the second operand. That is, the actual shift count is 0 to 63

Noncompliant Code Example

```
public void Main()
    short s = 1:
    short shortShift1 = (short)(s << 0); // Noncompliant</pre>
    short shortShift1 = (short)(s << 16); // Compliant as sh
    short shortShift3 = (short)(s << 32); // Noncompliant, t</pre>
    int i = 1:
    int intShift1 = i << 0; // Noncompliant</pre>
    int intShift2 = i << 32; // Noncompliant, this is equiva</pre>
    long lg = 1;
    long longShift1 = lg << 0; // Noncompliant</pre>
    long longShift2 = lg << 64; // Noncompliant, this is equ</pre>
```

Compliant Solution

```
public void Main()
    short s = 1:
    short shortShift1 = s;
    short shortShift1 = (short)(s << 16);</pre>
    short shortShift3 = (short)(s << 1);</pre>
    int i = 1;
    var intShift1 = i:
    var intShift2 = i << 1;</pre>
    long lg = 1;
    var longShift1 = lg;
    var longShift2 = lg << 1;</pre>
```

Code Smell

"switch" statements should have at least 3 "case" clauses

Code Smell

break statements should not be used except for switch cases

Code Smell

String literals should not be duplicated

Code Smell

Files should contain an empty newline at the end

Code Smell

Exceptions

This rule doesn't raise an issue when the shift by zero is obviously for cosmetic reasons:

- When the value shifted is a literal.
- When there is a similar shift at the same position on line before or after. E.g.:

```
bytes[loc+0] = (byte)(value >> 8);
bytes[loc+1] = (byte)(value >> 0);
```

See

• Microsoft documentation - Bitwise and shift operators

Available In:

sonarlint ⊖ | sonarcloud & | sonarqube

© 2008-2022 SonarSource S.A., Switzerland. All content is copyright protected. SONAR, SONARSOURCE, SONARLINT, SONARQUBE and SONARCLOUD are trademarks of SonarSource S.A. All other trademarks and copyrights are the property of their respective owners. All rights are expressly reserved.

Privacy Policy