^=

[Compound Operators]

Description

The compound bitwise XOR operator ^= is often used with a variable and a constant to toggle (invert) particular bits in a variable.

A review of the Bitwise XOR ^ operator:

0 0 1 1 operand1

0 1 0 1 operand2

----------

0 1 1 0 (operand1 ^ operand2) - returned result

Syntax

x ^= y; // equivalent to x = x ^ y;

Parameters

x: variable. Allowed data types: char, int, long.  
y: variable or constant. Allowed data types: char, int, long.

Example Code

Bits that are "bitwise XORed" with 0 are left unchanged. So if myByte is a byte variable,

myByte ^ B00000000 = myByte;

Bits that are "bitwise XORed" with 1 are toggled so:

myByte ^ B11111111 = ~myByte;

Notes and Warnings

Because we are dealing with bits in a bitwise operator - it is convenient to use the binary formatter with constants. The numbers are still the same value in other representations, they are just not as easy to understand. Also, B00000000 is shown for clarity, but zero in any number format is zero.

Consequently - to toggle bits 0 & 1 of a variable, while leaving the rest of the variable unchanged, use the compound bitwise XOR operator (^=) with the constant B00000011

1 0 1 0 1 0 1 0 variable

0 0 0 0 0 0 1 1 mask

----------------------

1 0 1 0 1 0 0 1

bits unchanged

bits toggled

Here is the same representation with the variables bits replaced with the symbol x. ~x represents the complement of x.

x x x x x x x x variable

0 0 0 0 0 0 1 1 mask

----------------------

x x x x x x ~x ~x

bits unchanged

bits set

So if:

myByte = B10101010;

myByte ^= B00000011 == B10101001;