

C Sharp Report - Class May 2, 2023

How Work Bitwise for Negative Numbers in C Sharp

The bitwise representation of negative numbers in C# is based on the two's complement system. In this system, the most significant bit (the leftmost bit) is used to represent the sign of the number. If the most significant bit is 0, the number is positive. If the most significant bit is 1, the number is negative.

Why in C Sharp ~2 is equal to -3?

The `~` operator in C# is the bitwise NOT operator. It flips the bits of the input value, changing 0s to 1s and 1s to 0s. When you apply the `~` operator to a number `x`, the result is `-x - 1`.

Bit e.g: `2 = 00000010` `~2 = 11111101 = -3`

Which is the difference between `Dictionary[index] = value` and `Dictionary.Add(index, value)` in C Sharp?

The difference between `Dictionary[index] = value` and `Dictionary.Add(index, value)` in C# is how they handle existing keys.

For `Dictionary[index] = value`, if the key `index` already exists in the dictionary, the value associated with that key is updated to `value`. If the key does not exist, a new key-value pair is added to the dictionary.

For `Dictionary.Add(index, value)`, if the key `index` already exists in the dictionary, an exception is thrown. If the key does not exist, a new key-value pair is added to the dictionary.