C# Fundamentals

By: Mosh Hamedani

What are Logical Operations?

Logical operations are part of Boolean algebra, which is often taught to students of computer science and electronic engineering at University. So, if you're not familiar with Boolean algebra, here is a brief introduction.

In Boolean algebra, the value of variables can only be **true** or **false**, also denoted 1 and 0 respectively. Unlike elementary algebra, where the main operations are addition, subtraction, etc, the main operations of Boolean algebra are **conjunction** (AND), **disjunction** (OR) and **negation** (NOT).

Logical AND

Let's assume we have two variables: **x** and **y**. In C#, the logical AND operator is indicated by &&. We can define a Boolean expression as follows:

```
z = x && y
```

In this expression, **z** is true if *both* **x** and **y** are true; otherwise, it'll be false.

What is a real-world example of this in programming? Imagine you're developing a loan application. The provider only offers loans to applicants who are over 18 and are citizen of the given country. In this example, we have two variables:

x = applicant being over 18
y = application being a citizen

z = is eligible to apply for loan = x && y

If *both* **x** and **y** are true, the applicant is eligible to apply for a loan.

Later, when we get to conditional statements, you can check to see if the above expression evaluates to true or false, and then, can change the flow of your application.

So, here is the rule of thumb with logical AND: if *both* **x** and **y** are true, **x && y** will be true; otherwise, it'll be false.

Logical OR

In C#, logical OR is indicated by two vertical lines (ll). Considering the following expression:

```
z = x || y
```

z will be true, if *either* **x** or **y** is true.

What is a real-world example of this? Imagine you're building software for a recruiter. For a given job application, applicants can apply if they have a degree in computing, or more than 5 years of experience in the field. You can model this using a Boolean expression as follows:

x = applicant has a degree in computing
y = applicant has more than 5 years of experience

z = application is eligible = x || y

If either x or y is true, z will be true.

So, unlike the logical AND, where both variables must be true, with logical OR, if at least one of them is true, the result will be true.

Logical NOT

The NOT operator in C# is indicated by an exclamation mark (!) and it reverses the value of a given variable or expression:

```
y = !x
```

So, here, if **x** is true, **y** will be false, and if **x** is false, **y** will be true.

Clean Coding

In all examples here, I used variables **x** and **y**, mainly to relate programming to Boolean algebra. But when it comes to coding, you should avoid using variable names such as x, y, z as they don't give a clue to other developers reading your code (or even yourself). Instead, use meaningful names. For instance, in the first example, you can replace **x**, **y** and **z** as follows:

```
x:  isOver18
y:  isCitizen
z:  isEligible
```

Often, it's a good practice to prefix Boolean names with IS or HAS (if possible).