Difference Between & and &&

www.differencebetween.com

Key Difference - & vs &&

In programming, there are situations to perform mathematical computations. An operator is a symbol to perform specific logical or mathematical functions on a value or a variable. The value or the variables in which the operations are happening are known as operands. There are various operators in programming languages. Some of them are arithmetic operators, relational operators, logical operators, bitwise operators and assignment operators. Arithmetic operators support mathematical operations such as addition, subtraction, multiplication etc. The relational operators are useful for finding the relationship of operands. Bitwise operators perform operations on bit level. One main bitwise operator is bitwise AND. It is represented using &. The logical operators help to analyze multiple conditions to make a decision. One main logical operator is logical AND. It is represented using &&. This article discusses the difference between & and &&. The **key difference** between & and && is that **& is a bitwise operator while && is a logical operator.**

What is &?

& is a bitwise operator. The programs are written by the programmer. These programs are understandable by humans but not understandable by the machine or the computer. Therefore, it is necessary to convert the human readable program into the machine-understandable format. The machine recognizes binaries; zeros and ones. Each binary is a bit. Bit-level processing is useful for increasing the speed. In bitwise operators such as &, the operator works on bits and perform bit by bit operation.

If a and b are variables and a contains 0 and b contains 1, then the bitwise AND is 0. If a is having value 1 and B is having value 0, then the output is 0. If a is having value 0 and B is having value 1, then the output is 0. If both a and b contain 1, then the output is 1. This 1 denotes true, and 0 denotes false. Assume that x is 4 and y is 5. The binary of 4 is 100. The binary of 5 is 101. When performing bit by bit operation, the bitwise AND is 100. When taking the AND operation of two different values will give 0. When both values are 1, then the output is 1.

main.cpp | *main.cpp

```
1   #include <iostream>
2
3   using namespace std;
4
5   int main()
6   {
7       int x= 4;
8       int y= 5;
9       int z;
10      z= x & y;
11      cout << z;
12      return 0;
13  }
14
15
16
```

Figure 01: Program using & Operator

According to the above program, variable x has the value 4. The binary value of 4 is 100. Variable y has the value 5. The binary value of 5 is 101. The z variable has the result of bitwise AND of x and y. The answer is 100. It is 4. Therefore, the output of the program will display 4.

What is &&?

It is a logical operator. It is used to make a decision based on multiple conditions. The && symbol represents the logical AND. In logical AND, if both operands are non zero, then the condition becomes true. When variable x is holding the value 1 and variable y is holding the value 0, the logical AND that is (x && y) is false or 0. One example of && is as follows.

*main.cpp

```cpp
#include <iostream>
using namespace std;
int main()
{
    int mark = 65;
    char grade;

    if (mark >= 75){
        grade = 'A';
    }
    else if(mark >= 60 && mark<75){
        grade = 'B';
    }
    else if(mark >= 45 && mark < 60)
    {
        grade = 'C';
    }
    else{
        grade ='F';
    }
    cout << grade;
    return 0;
}
```

Figure 02: A program using && Operator

According to the above program, the mark is a variable. It is assigned a value 65. In else if blocks the mark is compared. The && operator is used to refer the AND operation. In the else if (mark >= 60 && mark < 75), checks whether the mark variable is between 60 and 75. If the mark is less than 75 and it is greater than or equals to 60, the grade will be 'B'. In else if (mark >= 45 && mark < 60), the compiler will check whether the mark is between 45 and 60. If the mark is greater than or equal to 45 and mark is less than 60, then the grade is 'C'. These two statements include logical AND (&&).

What is the Similarity Between & and &&?

- Both are operators in programming.

What is the Difference Between & and &&?

& vs &&	
& is an operator in programming that performs bit by bit AND operations of the given operands.	&& is an operator in programming that performs logical AND operation on the multiple decisions.
Functionality	

& operator copies a bit to the result if it exists in both operands.	When using a && operator, if both operands are non-zero, then the condition becomes true.
Naming	
& is called as Bitwise AND	&& is called as Logical AND

Summary - & vs &&

Operators are used to perform mathematical and logical operations. Operators perform these operations on values or variables. They are known as operands. Some operators are arithmetic operators, assignment operators etc. Arithmetic operations contain addition, multiplication etc. Assignment operators, assign values from right side operands to left side operand. There are another two operators called bitwise operators and logical operators. Bitwise operators perform bit level operations. Logical operators make decisions based on multiple conditions. This article discussed the difference between & and &&. The difference between & and && is that & is a bitwise operator while && is a logical operator.

Reference:

1. tutorialspoint.com. “C Operators.” *The Point*. Available here

How to Cite this Article?

APA: Difference Between & and &&.(2018 February 07). Retrieved (date), from http://differencebetween.com/difference-between-and-vs/

MLA: "Difference Between & and &&" Difference Between.Com. 07 February 2018. Web.

Chicago: “Difference Between & and &&.” Difference Between.Com. http://differencebetween.com/difference-between-and-vs/accessed (accessed [date]).