Operators in C++

Once you have variables declared, you will do operations. An expression is the sequence of operands and operators to reduce in to one value. C++ provides you a lot of operators as shown below.

**-Assignment operator (=)**

Assignment operator is used to assign a value to a variable.   
Example:   
int x;   
int y;   
x=10; //assign 10 to x variable  
y=x; //assign x to y variable

**-Arithmetic operators (+, -,\*, /, %)**

Arithmetic operators are used to perform arithmetic operations on variables. 

|  |  |
| --- | --- |
| Sign | Meaning |
| + | Addition |
| - | Subtraction |
| \* | Multiplication |
| / | Division |
| % | Division of modulus |

    
Example:   
int x;   
int y;   
int z;   
int a;   
int m;   
x=10;   
z=y+x;   
a=z-(y=5);   
m=a%y; 

**-Compound assignation operators (+=, -=, \*=, /=, %=, <<=, >>=, &=, |=, ^=)**

Compound assignation operators are used to perform operations (+, -, \*, /,…) on variables with assignment operation.

|  |  |  |
| --- | --- | --- |
| Operator | Example | Equal to |
| += | i+=1 | i=i+1 |
| -= | i-=1 | i=i-1 |
| \*= | i\*=2 | i=i\*2 |
| /= | i/=2 | i=i/2 |
| %= | i%=2 | i=i%2 |
| <<= | i<<=2 | i=i<<2 |
| >>= | i>>=2 | i=i>>2 |
| &= | x&=3 | x=x&3 |
| |= | x|=y | x=x|y |
| ^= | x^=y | x=x^y |

**-Comparison Operators (==, >, <, >=, <=,!=)**

Comparison operators are used to compare variables. They return a Boolean value (true or false).

|  |  |  |
| --- | --- | --- |
| Operator | Meaning | Example |
| == | Equal to | (x==y) |
| > | Greater than | (x>y) |
| < | Less than | (x<y) |
| >= | Greater than or equal | (x>=y) |
| <= | Less than or equal | (x<=y) |
| != | Not equal to | (x!=10)&&(y!=10) |

**-Increment and Decrement operators (++, --, ++, --)**

These operators are used to make increment or decrement on a variable. 

|  |  |  |
| --- | --- | --- |
| Operator | Example | Equal to |
| ++ | x=++i | i=i+1  x=i |
| -- | x=--i | i=i-1  x=i |
| ++ | x=i++ | x=i  i=i+1 |
| -- | x=i-- | x=i  i=i-1 |

    
**-Logical Operators (&&, ||, !)**   
Logical operators are used to evaluate two expressions and return one result.

|  |  |  |
| --- | --- | --- |
| Operator | Meaning | Example |
| && | And | (x>y)&&(y<z) |
| || | Or | (x>y)||(x<3) |
| ! | Not | !(x==y) |

    
**-Bitwise operators (~, |, &, <<, >>, ^)**   
Bitwise operators are used to modify bits that represent the values that they store.

|  |  |  |
| --- | --- | --- |
| Operator | Meaning | Example |
| ~ | Reverse bits | x=~2 (result: x=-3) |
| | | Or | y=2|3(result: y=3) |
| & | And | y=2&3(result: y=2) |
| << | Shift bits to the left | y=5<<2 (result: y=20) |
| >> | Shift bits to the right | y=5>>2(result: y=1) |
| ^ | Excusive Or(XOR) | y=2^3(result: y=1) |