Programming In C++

Lecture 4 Operators in C++

Prepared By Dr. Ali Al-Sabaawi

Nineveh University- Faculty of IT- department of software

Operators

- There are four main classes of operators: arithmetic, relational, logical, and bitwise.

- Generally, there are three types of operators: unary, binary, and ternary. These terms reflect the number of operands an operator requires.

- Unary operators only require a single operand.

  -, ++, --

  for example: -5

  ++, it is increment operator

  --, it is decrement operator

Arithmetic Operators

- Binary operators work with two operands.

Operator	Meaning	Type	Example
+	Addition	Binary	`total = cost + tax;`
-	Subtraction	Binary	`cost = total - tax;`
*	Multiplication	Binary	`tax = cost * rate;`
/	Division	Binary	`salePrice = original / 2;`
%	Modulus	Binary	`remainder = value % 3;`

Arithmetic Operators

The precedence of the arithmetic operators

highest	++ −−
	– (unary minus)
	* / %
lowest	+ −

Expression	Value
5 + 2 * 4	13
10 / 2 - 3	2
8 + 12 * 2 - 4	28
4 + 17 % 2 - 1	4
6 - 3 * 2 + 7 - 1	6

Arithmetic Operators

- Binary operators work with two operands.

x++; is the same as x = x + 1;

x--; is the same as x = x – 1;

Note: x++ and ++x are different

For example, cout<<x++ means the value of x will be used then the increment operator will be executed.

Whereas, cout<<++x means the increment operator is executed then the value of will be displayed.

Arithmetic Operators

```
int main()
 {
 int x,a,b,c;
a = 2; b = 4;
 c = 5;
 x = a-- + b++ - ++c;
 cout<<"x: "<<x;
 return 0;
}
```

The result is 0

```
#include <iostream>
 int main()
{
   int n1 = 1;
   int n2 = ++n1;
   int n3 = ++ ++n1;
   int n4 = n1++;
   //  int n5 = n1++ ++;   // error
  //  int n6 = n1 + ++n1; // undefined behavior
(we can one of n1 to ifferent variable)
      cout << "n1 = " << n1 << '\n'
           << "n2 = " << n2 << '\n'
           << "n3 = " << n3 << '\n'
           << "n4 = " << n4 << '\n';
}
```

The output

n1 = 5
n2 = 2
n3 = 4
n4 = 4

Relational and Logical Operators

Relational refers to the operator that tests or defines some kind of relation between two entities.

Logical refers to the ways these relationships can be connected.

p	q	p && q	p || q	!p
0	0	0	0	1
0	1	0	1	1
1	1	1	1	0
1	0	0	1	0

Relational and Logical Operators

Relational Operators

Operator	Action
>	Greater than
>=	Greater than or equal
<	Less than
<=	Less than or equal
= =	Equal
!=	Not equal

Logical Operators

Operator	Action
&&	AND
||	OR
!	NOT

Table 2-5. *Relational and Logical Operators*

Relational and Logical Operators

Highest	!
	> >= < <=
	== !=
	&&
Lowest	||

Relational and Logical Operators

Ex: x=15 y=6

Z= x==y ; // z=0

Z= x!=y; //z=1

Z= x<y ; //z=0

Z= x<=y; //z=0

Z= x>y; //z=1

Z= x>=y; // z=1

Relational and Logical Operators

Ex: x=y=9;

D=f=4;

Z= x==y && d==f; //z= 1 && 1; //z=1

f=4 d=3

Z= 1&&0 //z=0

z= 1 || 0 //z=1

Ex: int x=15, y=6,z;

Z= x==y; //z=0

Z= !(x==y); //z=1

Z= x>y; //z=1

Z= !(x>y); //z=0

Ex: and

Z= (x==y)&&(x>y); //z= 0&&1 z=0

Z= (x==y)&&(x<y); //z= 0&&0 z=0

Z= (x!=y)&&(x<y); //z= 1&&0 z=0

Z= (x!=y)&&(x>y); //z= 1&&1 z=1

Ex: or

Z= (x==y)||(x>y); //z= 0&&1 z=1

Z= (x==y)||(x<y); //z= 0&&0 z=0

Z= (x!=y)||(x<y); //z= 1&&0 z=1

Z= (x!=y)||(x>y); //z= 1&&1 z=1

Combined Assignment Operators

The combined assignment operators, also known as compound operators, and arithmetic assignment operators.

Operator	Example Usage	Equivalent to
+=	`x += 5;`	`x = x + 5;`
-=	`y -= 2;`	`y = y - 2;`
*=	`z *= 10;`	`z = z * 10;`
/=	`a /= b;`	`a = a / b;`
%=	`c %= 3;`	`c = c % 3;`

Combined Assignment Operators

Example Usage	Equivalent to
`x += b + 5;`	`x = x + (b + 5);`
`y -= a * 2;`	`y = y - (a * 2);`
`z *= 10 - c;`	`z = z * (10 - c);`
`a /= b + c;`	`a = a / (b + c);`
`c %= d - 3;`	`c = c % (d - 3);`

Precedence Summary

Highest	( ) [ ] –> .
	! ~ ++ –– (type) * & sizeof
	* / %
	+ –
	<< >>
	< <= > >=
	== !=
	&
	^
	|
	&&
	||
	?:
	= += –= *= /= etc.
Lowest	,

Example of operators

Ex: int i=14, j=5, k=3;

What is the result of the following statement:

Z1= (i+j) * 2 – i/3;

= (14+5)*2-14/3;

=(19)*2-4;

=38-4=34

Z2= i%j+k*5;

Z2= 14%5+3*5;

= 4+15=19

Example of operators

Ex: int i=2, j=3 , k=4;
Z= i*(7+(j+3)/2)-k;
Z= 2*(7+(3+3)/2)-4
Z=2*(7+6/2)-4
Z=2*(10)-4 Z=20-4=16

HW1: int i=3 , j=8;
Z=i+j+3<j+I !=30
Hw2: int m=12, n=5, k=20;
Z= m*12+(m*n%13+m/n)*k/10;
Hw3: int b=2, m=5, n=4;
Z=5+ m*n – (b=b*3+2)+((n- - + ++m)*10);

The End