C Language

Chapter # 9

Elements of C language

Lecture: 13

Today's Lecture

- What are operators?
- What is Expression?
- Different types of operators in C language.

Operators

- Operators are the symbols that are used to perform certain operations on data.

  C provides variety of operators.

1. Arithmetic operators
2. Relational operators
3. Logical operators
4. Assignment operator
5. Increment and Decrement operators
6. Compound Assignment operators

Expression

- A statement that evaluates to a value is called an expression.
- An expression gives a single value.
- An expression consist of operators and operands.

For example:

2 + 3   m/n   x+100

1. Arithmetic operators

- Arithmetic operator is a symbol that performs mathematical operations on data.

Arithmetic Operators	**Operation**	**Example**
+	Addition	10 + 2 = 12
–	Subtraction	10 – 2 = 8
*	Multiplication	10 * 2 = 20
/	Division	10 / 2 = 5
%	Modulus – It returns the remainder after the division	10 % 2 = 0 (Here remainder is zero). If it is 10 % 3 then it will be 1.

- **Some important points about modulus operator.**

1. Modulus operator is also called remainder operator.
2. The modulus operator works only with integer values.
3. If modulus operator is used with the division of 0, the result is always be 0, For example the expression 0 % 5 will give 0 as result.
4. In expression 3 % 5, 3 is not divisible by 5. its result is 3.

2. Relational operators

- Relational operators are used to specify conditions in programs. A relational operator compares two values. It produces result as true or false.

Relational Operators	**Operation**	**Example**
>	Greater than	4 > 9 (False)
<	Les than	3 < 4 (True)
==	Equal to	7 == 2 (False)
>=	Greater than or equal to	7 >= 7 (True)
<=	Less than or equal to	8 <= 10 (True)
!=	Not equal to	9 != 9 (False)

3. Logical operators

- Logical operators are used to evaluate compound conditions in programs.

Logical Operators	Operation	Example
&&	AND	4 > 9 && 7 > 3 (False)
| |	OR	3 > 8 | | 8 > 4 (True)
!	NOT	7 == 2 (False) !(7 == 2) (True)

4. Assignment operator

- Assignment operator is used to assign a value or computational result to a variable.

Assignment Operator	**Operation**	**Example**

=	Assignment	a = 20;
		s = a + b;
		x = c – d + 100;

Assignment Statement

- A statement that assigns a value to a variable is known as assignment statement.

For example:

```
a = 20;
s = a + b;
x = c – d + 100;
```

Compound Assignment Statement

- An assignment statement that assigns a value to many variable is known as compound assignment statement.

For example:

```
a = b = 20;
x = y = z = 70;
m = n = 100;
```

Lvalue and Rvalue

- An Lvalue is an operand that can be written on the left side of assignment operator. It must always be a single value.
- An Rvalue is an operand that can be written on the right side of assignment operator.
- All values can be used as Rvalues but all values cannot be used as Lvalues.

  For example:

  A constant can be used as Rvalue but cannot be used as Lvalue.

  The expression x = 5 is valid but the expression 5 = x is not valid.

Data Type of expression

- The data type of an expression depends on the types of operands. For example the result of an expression in int, if both operands are integers.
- An expression in which operands are of different data types is called mixed-type expression. In this case the result of an expression is evaluated to large data type in the expression.

Expression	**Data type of Expression**
int + float	float
int – long	long
int * double	double
float / long double	long double

5. Increment operator

- The increment operator is used to increase the value of a variable by 1. it is denoted by the symbol ++
- The increment operator cannot increment the value of constant and expression. For example A++ and X++ is valid statements, but 10++ is an invalid statement. Similarly, (a+b)++ or ++(a+b) are also invalid.
- Increment operator can be used in two forms.

1. Prefix Form
2. Postfix Form

1. Prefix Form

In Prefix form, the increment operator is written before the variable as follows: ++y

The above line increments the value by 1.

2. Postfix Form

In Postfix form, the increment operator is written after the variable as follows: y++

The above line increments the value by 1.

Difference between prefix and postfix increment

- When increment operator is used independently, prefix and postfix form work similarly. For example, the result of A++ and ++A is same.

- But when increment operator is used in a large expression with other operators, prefix and postfix form work differently. For example, the result of two expressions A = ++B and A = B++ are different.

Difference between prefix and postfix increment

In prefix form:

The statement A = ++B contains two operators i.e = and ++. It works in the following order.

1. It increment the value of B by 1.
2. It assign the value of B to A.

The above statement is equivalent to the following two statements.

++B

A = B

Difference between prefix and postfix increment

In postfix form:

The statement A = B++ contains two operators i.e = and ++. It works in the following order.

1. It assign the value of B to A.
2. It increment the value of B by 1.

The above statement is equivalent to the following two statements.

A = B

B++

5. Decrement operator

- The decrement operator is used to decrease the value of a variable by 1. it is denoted by the symbol --
- The decrement operator cannot decrement the value of constant and expression. For example A-- and X-- is valid statements, but 10-- is an invalid statement. Similarly, (a+b)-- or --(a+b) are also invalid.
- Decrement operator can be used in two forms.

1. Prefix Form
2. Postfix Form

1. Prefix Form

In Prefix form, the decrement operator is written before the variable as follows: --y

The above line decrements the value by 1.

2. Postfix Form

In Postfix form, the decrement operator is written after the variable as follows: y--

The above line decrements the value by 1.

Difference between prefix and postfix decrement

- When decrement operator is used independently, prefix and postfix form work similarly. For example, the result of A-- and --A is same.

- But when decrement operator is used in a large expression with other operators, prefix and postfix form work differently. For example, the result of two expressions A = --B and A = B-- are different.

Difference between prefix and postfix decrement

In prefix form:

The statement A = --B contains two operators i.e = and --. It works in the following order.

1. It decrement the value of B by 1.
2. It assign the value of B to A.

The above statement is equivalent to the following two statements.

--B

A = B

Difference between prefix and postfix decrement

In postfix form:

The statement A = B-- contains two operators i.e = and --. It works in the following order.

1. It assign the value of B to A.
2. It decrement the value of B by 1.

The above statement is equivalent to the following two statements.

A = B

B--

6. Compound Assignment operator

- Compound assignment operators combines assignment operator with arithmetic operator. Compound assignment operator are used to perform mathematical operations more easily.
- These operators are: +=, -=, *=, /=, %=

Syntax:

variable op= expression

For example:

n += 10   is equivalent to   n = n +10

Some examples of compound assignment are as follows:

n += 10	is equivalent to	n = n + 10
n -= 10	is equivalent to	n = n - 10
n *= 10	is equivalent to	n = n * 10
n /= 10	is equivalent to	n = n / 10
n %= 10	is equivalent to	n = n % 10

Unary and Binary operators

Unary operators: A type of operators that works with one operand is known as unary operators.

These operators are unary operators: -, ++, --

The above operators are used with one operand as follows.

-a, n++, --x

Binary operators: A type of operators that works with two operand is known as binary operators.

These operators are unary operators: +, -, *, /, %

The above operators are used with one operand as follows.

a + b, x / y

The End

Read this topic from your books and ask question if any confusion.

JAZAKALLAH