Vivekananda Institute of Professional Studies

Unit 1

OPERATORS IN C

Programming using C language
Dr. Nivedita Palia

An operator is a symbol which operates on a value or a variable. For example: + is an operator to perform addition.

C programming has wide range of operators to perform various operations. For better understanding of operators, these operators can be classified as:

Arithmetic Operators (+,-,*,/,%)
Increment and Decrement Operators (++,--)
Assignment Operators (=, +=,-=,*=,/=, %=)
Relational Operators (>,<,>=,<=,==,!=)
Logical Operators (&&,||,!)
Conditional Operator (?:)
Bitwise Operators (&,|,!)
Special Operators (, , sizeof)

Arithmetic operator

Operator	Meaning of Operator
+	addition or unary plus
-	subtraction or unary minus
*	multiplication
/	division (returns quotient)
%	remainder after division(returns remainder)

Increment and Decrement Operators

1. increment ++ and decrement -- change the value of an operand (constant or variable) by 1.
2. Increment ++ increases the value by 1 whereas decrement -- decreases the value by 1.
3. These two operators are unary operators, meaning they only operate on a single operand.

```
eg. int a=10, b=100
 ++a = 11
--b = 99
```

Assignment operator

An assignment operator is used for assigning a value to a variable. The most common assignment operator is =

int a=12,b=5;

Operator	Example	Same as
▶=	a = b	a = b // a=5
	b=a	b=a//b=12
▶+=	a += b	a = a+b //a=17
▶-=	a -= b	a = a-b //a=7
▶*=	a *= b	a = a*b //a=60
▶/=	a /= b	a = a/b //a=2 ,12/5=2
▶%=	a %= b	a = a%b //a=2 , 12%5

Relational operator

A relational operator checks the relationship between two operands. If the relation is true, it returns 1; if the relation is false, it returns value 0.

Relational operators are used in decision making and loops.

Operator	Meaning of Operator	Example
==	Equal to	5 == 3 returns 0
>	Greater than	5 > 3 returns 1
<	Less than	5 < 3 returns 0
!=	Not equal to	5 != 3 returns 1
>=	Greater than or equal to	5 >= 3 returns 1
<=	Less than or equal to	5 <= 3 returns 0

Vivekananda Institute
of Professional Studies

C Logical Operators

Logical Operator AND (&&)

C1 && C2 =

Where C1 , C2 are conditions.

Indicates whether both operands are true or not.
If both operands have nonzero values, the result has the value 1. Otherwise, the result has the value 0. The type of the result is `int`. Both operands must have an arithmetic or pointer type. The usual arithmetic conversions on each operand are performed.
The logical AND (&&) should not be confused with the bitwise AND (&) operator. For example:

`1 && 4` evaluates to `1` (`True && True = True`)

while

`1 & 4` (`0001 & 0100 = 0000`) evaluates to `0//bitwise operator`

Logical AND Operators

C1	c2	C1&&c2
F	F	F
F	T	F
T	F	F
T	T	T

C Logical Operators

Logical Operator OR (||)

Indicates whether either operand is true. If either of the operands has a nonzero value, the result has the value 1. Otherwise, the result has the value 0. The type of the result is `int`. Both operands must have a arithmetic or pointer type. The usual arithmetic conversions on each operand are performed.

The logical `OR` (`||`) should not be confused with the bitwise `OR` (`|`) operator. For example:

`1 || 4` evaluates to `1` (or `True || True = True`)
while `1 | 4` (`0001 | 0100 = 0101`) evaluates to `5`

Logical OROperators

c1	c2	C1||c2
F	F	F
F	T	T
T	F	T
T	T	T

C Logical Operators: NOT (toggle)

Logical `NOT` Operator (unary)

!

Logical not operator. Produces value 0 if its operand or expression is true (nonzero) and the value 1 if its operand or expression is false (0). The result has an `int` type. The operand must be an integral, floating, or pointer value.

```
Z=!(4 == 2)//!(0) ans is z=1
!x
```

Vivekananda Institute of Professional Studies

Logical NOT Operator (unary)

c1	!c1
F	T
T	F

C Conditional Operator

Conditional Operator (ternary)/ short cut of- if else

```
[?:]
This operator uses 3 operands.
(condition)? statement1:statements2
```

The first operand/expression is evaluated, and its value determines whether the second or third operand/expression is evaluated:
If the value is true, the second operand/expression is evaluated.
If the value is false, the third operand/expression is evaluated.
The result is the value of the second or third operand/expression. The syntax is:

```
First operand ? second operand : third operand
size != 0 ? size : 0
```

BITWISE OPERATOR

Used for manipulating data at bit level

Not applied to float or double

operator	Meaning
&	Bitwise AND
|	Bitwise OR
^	Bitwise exclusive OR
<<	Shift left
>>	Shift right

Special OPERATOR

Comma operator: used to link related expression together. A comma linked list of expression evaluated left to right and the value of right most expression is the value of combined expression

V= (x=10,y=5,x+y);

sizeof operator: it is compile time operator.

It returns number of bytes the operand occupies.

m=sizeof (sum);

- Q1 Swapping with or without third varible
- Q2 SI
- Q3 temp conversion
- Q4 greatest of two number, 3 numbers .

END