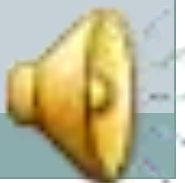


Basic of C Programming



CONCEPT OF OPERATORS - II



Logical Operators -



- An expression containing logical operator returns either 0 or 1 depending upon whether expression results true or false.
- Logical operators are commonly used in decision making in C programming.

Operator	Meaning	Example
&&	Logical AND. True only if all operands are true	If c = 5 and d = 2 then, expression ((c==5) && (d>5)) equals to 0.
	Logical OR. True only if either one operand is true	If c = 5 and d = 2 then, expression ((c==5) (d>5)) equals to 1.
!	Logical NOT. True only if the operand is 0	If c = 5 then, expression !(c==5) equals to 0.

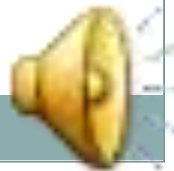


Bitwise Operators -



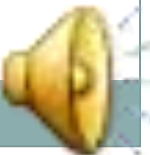
- During computation, mathematical operations like - addition, subtraction, multiplication, division, etc are converted to bit-level which makes processing faster and saves power.
- Bitwise operators are used in C programming to perform bit-level operations.

Operators	Meaning of operators
&	Bitwise AND
	Bitwise OR
^	Bitwise exclusive OR
~	Bitwise complement



Truth Table

a	b	a & b	a b	a ^ b
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0



Shift Operators -



- Shift Operators are also the part of Bitwise Operators.
- The bitwise shift operators move the bit values of a binary object.
- The left operand specifies the value to be shifted.
- The right operand specifies the number of positions that the bits in the value are to be shifted.

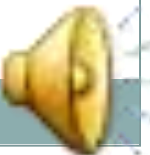
Operators	Meaning of operators
<<	Shift left
>>	Shift right

a = 0001000

b = 2

a << b = 0100000

a >> b = 0000010



Ternary Operators -



- It is also known as Conditional Operator.
- The conditional operators in C language are known by two more names
 - **Ternary Operator**
 - **? : Operator**
- It is actually the if condition that we use in C language decision making, but using conditional operator, we turn the if condition statement into a short and simple operator.
- **Syntax – expression 1 ? expression 2: expression 3**

```
int a = 30;  
int b = 20;  
(a > b) ? printf("a is greater") : printf("b is greater");
```

condition

condition is true

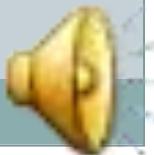
Condition is false



Special Operators -



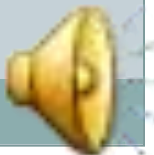
Special Operators	Meaning	Examples
*	Pointer Operator	*ptr
&	Address of a variable	&a
,	Comma operator	a , b
-	Negative value	-x
Type Casting	Forced Type Conversion	float(a)
sizeof	Size of the subsequent data types in bytes	sizeof(int)



Operator Precedence -



	Description	Represented By
1	Parenthesis	() []
1	Structure Access	. ->
2	Unary	! ~ ++ -- - * &
3	Multiply, Divide, Modulus	* / %
4	Add, Subtract	+ -
5	Shift Right, Left	>> <<
6	Greater, Less Than, etc	> < =
7	Equal, Not Equal	== !=
8	Bitwise AND	&
9	Bitwise Exclusive OR	^
10	Bitwise OR	
11	Logical AND	&&
12	Logical OR	
13	Conditional Expression	?:
14	Assignment	= += -= etc
15	Comma	,



Thank you...!

