TOPIC-6 Operators

Defination of operators:

operator is a symbol which tells the computer to perform certain mathematical and logical operator.

operators are used in C language program to operate on data and variables. It is used to combine various constants, variables and functions.

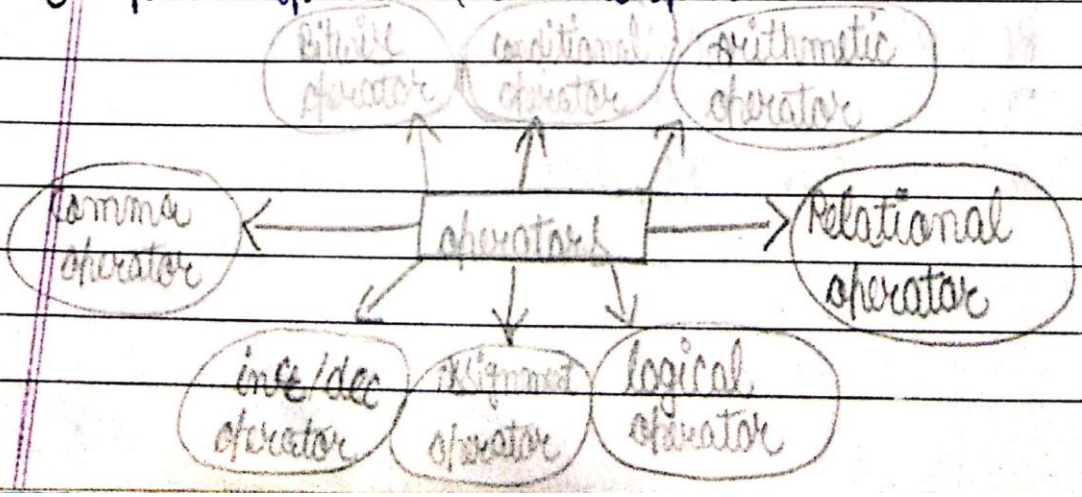
An operator is performed on operands.

eg: A+B;

In this example A, B are operants and + is operator.

types of operators:

1.) Arithmetic operators
2.) Relational operators
3.) Logical operators
4.) Assignment operators
5.) Inc/dec operator / unary operator
6.) Conditional operators
7.) Bitwise operators
8.) Special operators / comma operator



Page No
Date: / /201

1.) Arithmetic operators: An Arithmetic operator perform mathematical operations such as Addition, Substraction, Multiplication, Division. Arithmetic operator is a binary operator which requires atleast two operants to perform. These act upon numeric values.
eg: let a and b are two operants.
a=30
b=3

operators	Description	example
+	Addition	a+b = 10+3 = 13
-	Substraction	a-b = 10-3 = 7
*	Multiplication	a*b = 10*3 = 30
/	Division	a/b = 10/3 = 10/3

(2) Relational operators: Relational operator checks the relationship between the operants.
If the relation is true it returns 1.
If the relation is false it returns 0.

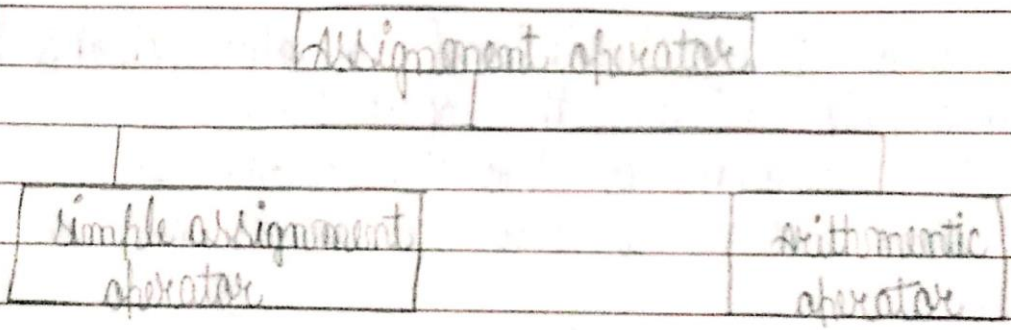
operator	description
>	greater than
>=	greater than equal to
<	less than
<=	less then equal to
==	equal to
!=	Not equal to

(3) logical operator: logical operators are used in decision making in C programming.

It is also used to compare two or more logical expressions.
The result of such operations is always logical: i.e. either true (1) or false (0).

operator	Meaning
&&	logical and
||	logical or
!	logical not

4. Assignment operators: The assignment operator assign new values to variables. The equal sign (=) is the fundamental assignment operator in C. Assignment operators are of two types.



Simple assignment operator
Arithmetic operator
eg: Assignment operator → expression
use of this operator is written in the general form as.

identifier = expression

a=10;
10 is stored in value a

Page No.
Date : / /201

5) Unary operator: Unary operators are the operators that act upon a single operand to produce a new value.
These operators are also called increment / decrement operator.

⇒ increment operator: These operators are used for incrementing value one by one. The symbol used for increment operator is (++).

⇒ decrement operator: These operators are used for decrementing value one by one. The symbol used for decrement operator is (--).

There are two ways of representing increment or decrement operator.

① ~~y++~~ postfix eg: y++ y--
② prefix eg: ++y --y

① postfix increment/decrement: Here first the value of variable is taken for operation then value of variable is increment / decrement.

eg: ① x = y++ where y = 10;
② a = b-- where b = 10;

after execution: x = 10;
y = 11;

after execution: a = 10;
b = 9;

② prefix increment/decrement: Here, first the value of variable is increment or decrement then the value of variable is taken for operation.

Page No.
Date: / /201

eg: x = ++y; where y = 10;
a = --b; where b = 10;

after execution: x = 10;
y = 11;
after execution: a = 10;
b = 9;

6.) conditional operator / ternary operator: conditional operator (?:) is ternary operator and is used in certain operation situation. It is represented by 2 symbols.
eg: ? :
It is worked on 3 operants, so it is called as ternary operator.
Syntax: The general form of the conditional expression is

exp 1 ? exp 2 : exp 3

where exp 1, exp 2, exp 3 are expressions.

7.) Bitwise operator: Bitwise operators are used for manipulating data at bit level. These operator used for testing the bits and shifting them from right to left.

C supports the 6 bit operators.

operators	Description
>>	Shift Right
<<	Shift left
~	1's complement
&	Bitwise and
|	Bitwise or
^	Bitwise exclusive (X) or

Page No.
Date : / /201

① Right shift operator: let $y = x >> 2$
eg: $x = 00011001$
$y = 10000110$
In right shift operator when we are shifted the values in left side then values are changed
eg: 0, 1 change into 1, 0

② left shift operator: let $y = x << 2$
eg: $x = 00011$
$y = 001100$
In left shift operator when we are shifted the values to right side then values are not changed:
eg: 1, 0 then result is 1, 0

③ one's complement operator: In this operator change all bit to 0 and each 0 bit to 1.

eg: $x = 11100100$
$y = 00011011$

④ Bitwise and operator:
eg: if $x = 2$ $y = 3$
$z = x \& y$ can be written as $z = x \& 3$
$x = 2 = 00000111$
$y = 3 = 00000110$

x	y	x&y
0	0	0
0	0	0
0	0	0
0	0	0
1	1	1
1	1	1
1	0	0

Page No.
Date : / /201

⑤ Bitwise OR operators

eg: let x=2 y=3 z=x|y

x=2 = 00010
y=3 = 00011

x	y	x|y
0	0	0
0	0	0
0	0	0
1	1	1
0	1	1

⑥ Bitwise xor operators

eg: let x=2 y=3 z=x^y

x=2 = 0000 111
y=3 = 0000 110

x	y	x^y
0	0	0
0	0	0
0	0	0
0	0	0
1	1	0
1	1	0
1	0	1

In this operator
1^1 = 0
when both are same it produce 0 when both are different it produce 1