OPERATIONS

- Operations means built in data type in c programming.
- Normal Operators are: =, +, -, *, /, <
- C supports a rich set of built in operators.
- An Operator is a symbol that tells the computer to perform mathematical or logical manipulations.
- C operators can be classified into a number of categories.

1. Arithmetic Operators
2. Relational Operators
3. Logical Operators
4. Assignment Operators
5. Increment & Decrement
6. Conditional Operators
7. Bitwise Operators
8. Special Operators

1. Arithmetic Operators

Operator	Meaning
+	Addition or Unary Plus
-	Subtraction or Unary minus
*	Multiplication
/	Division
%	Modulus Division

PALKULAM, KANYAKUMARI OBSERVE OPTIMIZE OUTSPREAD ROHINI COLLEGE OF ENGINEERING AND TECHNOLOGY

These operators built in data type allowed in C.

Example

a-b, a+b, a*b, a/b, a%b. Here a and b are variables and are known as operands.

Arithmetic operators have 3 types

1. Integer arithmetic operators
2. Real arithmetic operators
3. Mixed – mode arithmetic operators

1. Integer arithmetic operators

When both the operands in a single arithmetic expressions such as a+b are integers the expression is called an integer expression and the operation is called integer arithmetic.

If a and b are integers, then for a = 14 and b=4

a-b = 10
a+b = 18
a*b = 56
a%b =2 ->remainder

2. Real Arithmetic

- ❖ An arithmetic operation involving only real operands is called real arithmetic.
- ❖ A real operand may assume vales either in decimal or exponential notation.
- ❖ The operator % cannot be used with real operands.

Example:

a=10.5
b=15
Both modulus cannot be used

3. Mixed – mode Arithmetic:

- ❖ When one of the operands is real and the other is integer, the expression is called a mixed – mode arithmetic operation.
- ❖ If either operand of the real type, then only the real operation is performed and the result B always a real number. Then, 15/10.0 =1.5

2. Relational Operators

- ❖ We often compare two quantities and depending on their relation, take certain decision.
- ❖ For example, we may compare the age of two persons, or the price of two items and so on.
- ❖ The comparison can be done with the help of relational operators

Operator	Meaning
<	Is less than
<=	Is less than equal to
>	Is greater than
>=	Is less than
==	Is equal to

!=	Is not equal to

Example

4.5 <=10 True
4.5<10 False
3.5>=10 False
10<7 + 5 True
5 = 5 True

3. Logical Operators

It is a Binary value. In addition to the relational operators, c has the following three logical operators.

&& - Logical AND
// - Logical OR
! - Logical NOT

Op-1	Op-2	Op 1 && Op 2	Op 1 // Op 2
1	1	1	1
1	0	0	1
0	1	0	1
0	0	0	0

Example

If (age 55&& salary <5000) if condition True print will be done.

4. Assignment operators

Assignment operators are used to assign the result of an expression to a variable.

Assignment Operator	Short head Operator
a=a+1	a+=1
a=a-1	a-=1
a=a*/(m+n)	a*=m+n
a=a/(m+n)	a/=m+n

5. Increment and Decrement Operator:

C allow two very useful operators not generally found in other languages. There are the increment & decrement operator

++ and --

The operators ++ adds 1 to the operand, while subtracts 1. Both are unary operators and takes the following form,

```
++m or m++
--mm or m--
```

++mm is equivalent to m=m+1
--mm is equivalent to m=m-1

We use the increment and decrement statements in for and while loops extensively

6. Conditional Operators

A ternary operator "3" is available in c to construct conditional expressions of the form.

```
exp1? exp 2 : exp3
```

Where exp1, exp2, exp 3 are expressions.
Exp1 is evaluated first. If it is non zero then the expression exp2 is evaluated and becomes the value of the expression.

Example,

```
a=10  b=15
x=(a>b)?a:b
```

In this example X will be assigned the value of b,

```
if(a>b)            if(10>15)
 x=a;                 x=10
 else                 else
 x=b                  x=15
```

7. Bitwise Operator

C has a distinction of supporting special operators known as bitwise operators for manipulation of data at bit level.

These operators are used for testing the bits or shifting, then right or left.

Bitwise operators may not be applied to float or double.

OPERATOR	**MEANING**
&	Bitwise AND
/	Bitwise OR
^	Bitwise exclusive OR
<<	Shift Left
>>	Shift Right

8. Special Operators

C supports some special operators such as

1. Comma operators
2. Size of operator
3. Pointer Operator

4. Mambol Selection Operator(. and ->)

1. **Comma Operator**

   It can be used to link the related expression together. A comma linked list of expressions are evaluated. Left to right end the value of right.

Example,

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

2. **Size of Operator**

   The size is a compile time operator and where used with an operand. It returns the number of bytes the operand occupies. The operand may be a variable a constant or a data type qualifier.

Example,

   m= sizeof(sum);
   n=sizeof(longint);

3. **Mambol Selection Operator**

   del(.) operator – Used to access a member of a struct

   arrow operator(->) – Used to access a member of a struct which is referenced by the pointer in question.