Unit-4 OPERATORS

Operators

-An operators is a symbol used to perform some operations or action on one or more operands.

- Operators that requires two operands are **binary operators.**

- Operators that require one operand is **unary operators.**

- Operators that requires three operands is a **ternary operators.**

For example, in mathematical expression X + Y, where x and y are operands and + is a operator.

Types of operators in C Language:

1. Arithmetic Operators
2. Relational Operators
3. Equality Operators
4. Logical Operators (Boolean Operators)
5. Assignment Operators
6. Increment/Decrement Operators
7. Conditional Operators (Ternary Operators)
8. Bitwise Operators
9. Special Operators (Size of and Comma Operators)

1. Arithmetic Operators:

- Arithmetic operator is a symbol used for basic mathematical calculations.
- It usually takes two operands and returns the result of the mathematical calculations.
- C contains five arithmetic operators:

Operators	Purpose	Example
+	Addition	A+B
-	Subtraction	A-B
*	Multiplication	A*B
/	Division(only for integer and floating point division)	A/B
%	Remainder (only for integer division)	A%B

2. Relational Operators:

-Relational operator is a symbol that determines the relationship between different operands.

- Relational operator always give the final result in terms of true and false after its operations.
- C provides four relational operators:

Operators	Purpose	Example
<	Less than	A<B
<=	Less than or equal to	A<=B
>	Greater than	A>B
>=	Greater than or equal to	A>=B

3. Equality Operators:

- Equality operator is a symbol used to compare two operands if they are equal or not.
- It returns either true or false result.
- It is binary operator.
- C offers two equality operators:

Operator	Purpose	Example
==	Equals to	A==B
!=	Not equal to	A!=B

4. Logical Operators:

-Logical operator is a symbol that logically connects the logical expressions i.e. it is used to connect two or more expressions.

- C consists of three logical operators:

Operator	Purpose	Example	What it evaluates to
&&	AND	A && B	True(1) only if both A and B are true; otherwise it is false(0).
||	OR	A || B	True(1) if either A or B is true; false(0) only if both are false.
!	NOT	!A	False(0) if A is true; true(1) if A is false.

5. Assignment Operators:(=)

- Assignment operator is a symbol used to assign a value or a result of an expression to an identifier.

Simple Assignment Operators:

The most commonly used assignment operator is =.

Syntax: identifier=expression;

Example: x=40;   y=a + b;

Continue…

Compound Assignment:

The assignment operator(=) can be used with arithmetic operators as listed below:

Operator	Purpose	Example
+=	Add and assign	A+=B is same as A=A+B
-=	Subtract and assign	A-=B is same as A=A-B
*=	Multiply and assign	A*=B is same as A=A*B
/=	Divide and assign	A/=B is same as A=A/B
%=	Takes modulus and assign(only for integer division)	A%=B is same as A=A%B

Continue…

Multiple Assignment:

C also allow multiple assignment statements using =.

For example:

```
int a=b=c=20;
float x=y=4.5;
```

This kind of multiple assignment is only possible if all the variable types in the statement are the same type.

6. Increment/Decrement Operator:

- These operators are also called **unary operator**.

- An increment operator (++) is used to increment the value of variable by 1.

- A decrement operator (--) is used to decrement the value of variable by 1.

For example:

int a=5;

a++; → 6

a--; → 4

a++ is same as a=a+1.

a-- is same as a=a-1.

Continue…

Increment/Decrement

1. Pre-increment/ Pre-decrement: (++a/--a)

-means first increment/decrement then assign it to the another variable.

Example: int a=5;

x=++a; → 6

x=--a; → 4

2. Post-increment/Post-decrement: (a++/a--)

- means first assign it to the another variable and then increment/decrement.

Example: int a=5;

x=a++; → 5

x=a--; → 5

7. Conditional Operator (Ternary Operator)(?:)

- It stores a value depending upon a condition.

Syntax: condition ? expression1 : expression2;

If the condition is true then the expression 1 will be evaluated and if the condition is false then the expression 2 will be evaluated.

Example: marks>=40 ? “Pass” : “Fail”;

8. Bitwise Operator:

-Bitwise operators are used to performing calculations using binary digits(0 or 1) which can be performed using following operators.

There are six bitwise operators:

Operator	Purpose
&	Bitwise AND
|	Bitwise OR
^	Bitwise XOR
~	Bitwise Complement
<<	Left shift
>>	Right Shift

9. Special Operators:

- The Comma Operator:

C allows us to put multiple expressions in the same statement, separated by a comma. The expressions are evaluated in left to right order.

For example:

```
int x=5,y=10;
float a,b,c;
printf(“%d”, s);
scanf(“%d”, &a);
```

Continue…

- Size of operator:

- It is a compile time unary operator that returns the number of bytes the operand occupies.

Example:

```
int x;
printf(“Size of integer=%d bytes”, sizeof(x));
printf(“Size of integer=%d bytes”, sizeof(int));
```