FORTRAN - OPERATORS

An operator is a symbol that tells the compiler to perform specific mathematical or logical manipulations. Fortran provides the following types of operators:

- Arithmetic Operators
- Relational Operators
- Logical Operators

Let us look at all these types of operators one by one.

Arithmetic Operators

Following table shows all the arithmetic operators supported by Fortran. Assume variable **A** holds 5 and variable **B** holds 3 then:

Show Examples

Operator	Description	Example
+	Addition Operator, adds two operands.	A + B will give 8
-	Subtraction Operator, subtracts second operand from the first.	A - B will give 2
*	Multiplication Operator, multiplies both operands.	A * B will give 15
/	Division Operator, divides numerator by de-numerator.	A / B will give 1
**	Exponentiation Operator, raises one operand to the power of the other.	A ** B will give 125

Relational Operators

Following table shows all the relational operators supported by Fortran. Assume variable **A** holds 10 and variable **B** holds 20, then:

Show Examples

Operator	Equivalent	Description	Example
==	.eq.	Checks if the values of two operands are equal or not, if yes then condition becomes true.	$A == B$ is not true.
/=	.ne.	Checks if the values of two operands are equal or not, if values are not equal then condition becomes true.	$A != B$ is true.
>	.gt.	Checks if the value of left operand is greater than the value of right operand, if yes then condition becomes true.	$A > B$ is not true.
<	.lt.	Checks if the value of left operand is less than the value of right operand, if yes then condition becomes true.	$A < B$ is true.
>=	.ge.	Checks if the value of left operand is greater than or equal to the value of right operand, if yes then condition becomes true.	$A >= B$ is not true.

<=	.le.	Checks if the value of left operand is less than or equal to the value of right operand, if yes then condition becomes true.	$A <= B$ is true.

Logical Operators

Logical operators in Fortran work only on logical values .true. and .false.

The following table shows all the logical operators supported by Fortran. Assume variable A holds .true. and variable B holds .false. , then:

Show Examples

Operator	Description	Example
.and.	Called Logical AND operator. If both the operands are non-zero, then condition becomes true.	$A .and. B$ is false.
.or.	Called Logical OR Operator. If any of the two operands is non-zero, then condition becomes true.	$A .or. B$ is true.
.not.	Called Logical NOT Operator. Use to reverses the logical state of its operand. If a condition is true then Logical NOT operator will make false.	$!A .and. B$ is true.
.eqv.	Called Logical EQUIVALENT Operator. Used to check equivalence of two logical values.	$A .eqv. B$ is false.
.neqv.	Called Logical NON-EQUIVALENT Operator. Used to check non-equivalence of two logical values.	$A .neqv. B$ is true.

Operators Precedence in Fortran

Operator precedence determines the grouping of terms in an expression. This affects how an expression is evaluated. Certain operators have higher precedence than others; for example, the multiplication operator has higher precedence than the addition operator.

For example, x = 7 + 3 * 2; here, x is assigned 13, not 20 because operator * has higher precedence than +, so it first gets multiplied with 3*2 and then adds into 7.

Here, operators with the highest precedence appear at the top of the table, those with the lowest appear at the bottom. Within an expression, higher precedence operators will be evaluated first.

Show Examples

Category	Operator	Associativity
Logical NOT and negative sign	.not. –	Left to right
Exponentiation	**	Left to right
Multiplicative	* /	Left to right
Additive	+ -	Left to right
Relational	< <= > >=	Left to right
Equality	== /=	Left to right
Logical AND	.and.	Left to right
Logical OR	.or.	Left to right

Assignment	=	Right to left