



SNS COLLEGE OF TECHNOLOGY

Coimbatore-35
An Autonomous Institution

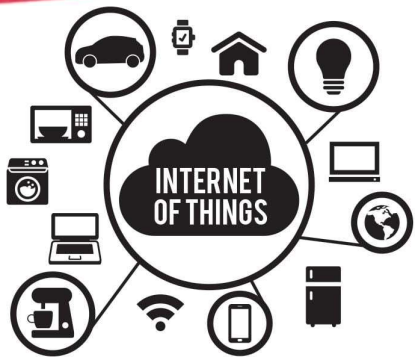


Department of Information Technology

23CST101 - PROBLEM SOLVING and C PROGRAMMING

I B.Tech. IT/ I SEMESTER

UNIT II : C PROGRAMMING BASICS



Introduction to 'C' Programming –Fundamental rules – Structure of a 'C' program – Compilation and Linking processes –Constants, Variables, keywords, Identifier, Delimiters – Declaring and Initializing variables – Data Types – Operators and Expressions –Managing Input and Output operations – Decision Making and Branching –Looping

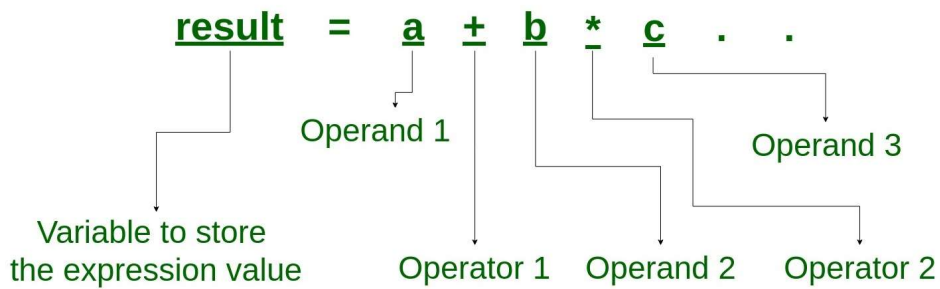


Operators and Expressions

2/17

- An **operator** is a symbol that tells the computer to perform certain mathematical or logical manipulations.
- Operators are used in programs to manipulate data and variables.
- They usually form a part of the mathematical or logical **expressions**.
- An **expression** is a sequence of ‘operands’ and ‘operators’ that reduces to a single value.
- For example, $10 + 15$ is an expression whose value is 25.

What is an Expression?



Algebraic Expression	C Expression
$a \times b - c \times d$	$a * b - c * d$
$(m + n) (a + b)$	$(m + n) * (a + b)$
$3x^2 + 2x + 5$	$3 * x * x + 2 * x + 5$
$\frac{a + b + c}{d + e}$	$(a + b + c) / (d + e)$
$\left[\frac{2BY}{d+1} - \frac{x}{3(z+y)} \right]$	$2 * b * y / (d + 1) - x / 3 * (z + y)$

Operators in C

	Operators	Type
Unary Operator →	++, --	Unary Operator
Binary Operator {	+, -, *, /, %	Arithmetic Operator
	<, <=, >, >=, ==, !=	Relational Operator
	&&, , !	Logical Operator
	&, , <<, >>, ~, ^	Bitwise Operator
	=, +=, -=, *=, /=, %=	Assignment Operator
Ternary Operator →	?:	Ternary or Conditional Operator

OPERATOR	TYPE	ASSOCIIVITY
() [] . ->		left-to-right
++ -- +- ! ~ (type) * & sizeof	Unary Operator	right-to-left
* / %	Arithmetic Operator	left-to-right
+ -	Arithmetic Operator	left-to-right
<< >>	Shift Operator	left-to-right
< <= > >=	Relational Operator	left-to-right
== !=	Relational Operator	left-to-right
&	Bitwise AND Operator	left-to-right
^	Bitwise EX-OR Operator	left-to-right
	Bitwise OR Operator	left-to-right
&&	Logical AND Operator	left-to-right
	Logical OR Operator	left-to-right
?:	Ternary Conditional Operator	right-to-left
= += -= *= /= %= &= ^= = <<= >>=	Assignment Operator	right-to-left
,	Comma	left-to-right



Operators and Expressions

6/17

Operator Precedence

- when two operator of the same priority are found in the expression precedence is given to the extreme left operator
- if there are more set of parenthesis in the expression the innermost parenthesis will be solved first followed by the second and so on

Comma and Conditional Operator

- Comma Operator
 - Used to separate two expression
 - Lowest priority among all operators
 - Example - $a=2, b=4, c=a+b;$
- Conditional Operator
 - Condition followed by two statements or value
 - Condition? (expression1): (expression2)

Arithmetic Operator

- Two types - Binary and Unary Operators

Arithmetic Operators in C

Arithmetic Operators	Meaning	Examples
+	Addition	$1008 + 108.8 = 1116.800000$
-	Subtraction	$1008 - 108.8 = 899.200000$
*	Multiplication	$1008 * 108.8 = 109670.400000$
/	Division	$1008 / 108.8 = 9.264706$
%	Modulus	$1008 \% 108.8 = 28.800000$

Operator	Symbol	Description
Unary minus	-	Returns opposite sign operand
Increment Operator	++	Increases the value of an operand by 1
Decrement Operator	--	Decrements the value of an operand by 1
NOT Operator	!	Reverses the logical state of an operand
Addressof Operator	&	Returns the address of an operand
sizeof () operator	sizeof ()	Returns the size used by the data type in bytes

Relational Operator

OPERATOR	MEANING	EXAMPLE	RESULT
<	Less than	$1 < 2$	True
>	Greater than	$1 > 2$	False
<=	Less than or equal to	$1 <= 2$	True
>=	Greater than or equal to	$1 >= 2$	False
==	Equal to	$1 == 2$	False
!=	Not equal to	$1 != 2$	True

Logical Operator

Logical Operators

Operator	Meaning	Example	Result
&&	Logical and	$(5 < 2) \&\& (5 > 3)$	False
	Logical or	$(5 < 2) \ \ (5 > 3)$	True
!	Logical not	$!(5 < 2)$	True

Bitwise Operator

Name	Symbol	Usage	What it does
Bitwise And	&	$a \& b$	Returns 1 only if both the bits are 1
Bitwise Or		$a b$	Returns 1 if one of the bits is 1
Bitwise Not	~	$\sim a$	Returns the complement of a bit
Bitwise Xor	^	$a \wedge b$	Returns 0 if both the bits are same else 1
Bitwise Left shift	<<	$a \ll n$	Shifts a towards left by n digits
Bitwise Right shift	>>	$a \gg n$	Shifts a towards right by n digits

Bitwise Operator

AND Truth Table

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

OR Truth Table

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

XOR Truth Table

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

NOT Truth Table

A	B
0	1
1	0



References

- <https://www.programiz.com/c-programming/c-operators>



Thank You!