Operators in C.

SigmaPi Academy

Operators are the one of the most important part of any C program. These are symbols that help us in forforming cortain tasks for on the variables, which are called of erands. They are 3 types of operators when grouped awarding to the number of operators to operators to operators.

(a) Uniary oferator: ++, -- (They require one

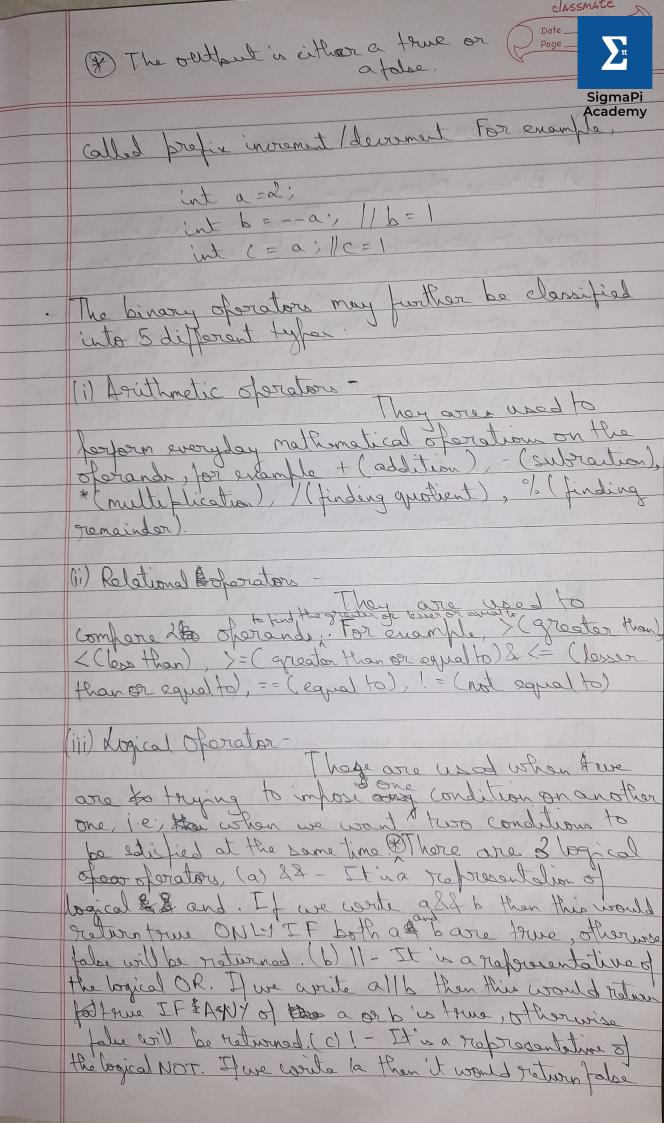
(C) Ternary oferator: ?: (They require 3 oferants.

8: also called Conditional
Operator)

The unary oferators either increase the value of the variable by I (++) or decrease it by I (--). When do they do this two change defends on the position of the variable. If the variable is placed before the operator i.e., a++ or a-+
then the value is used before changing. This is called postfix increment decrement. For example,

int a = 1;int b = a + t; 1/b = 1int c = a; 1/8 = 2.

If the variable is placed after the operator is ex + + a or -- a, then the value is schanged before using it. This is



Ja is true and vice-versa. \sum_{Π} (iv) Bituise Oforatore - Thou are used when we no martered buel tid emos wroped of thow the operands. When the mathematical calculation what the part half level they are faster than halfen at the bit bord that the operands are in the remainded. Here boths the operands are converted to but - level and then the ofercation is ferformed. For enample, & (AND), 1 COR), (XOR) << (Left Shift), >> (right shift) & & (NOT). They are used to

all adult shearesto at to jouler transfish resides

are abused to the other transfish resides

are sharesto at to one wateresto previous resto La VALUE, i.e., Variable Assignment operator to # VALVE. The value must be of the same data type as the variable of For example, = (equal to), += (similar to similar to a = q + x where x is a value), -= (similar to a=a-x), *= (similar to a = a *x), /= (similar to a=a/x) The conditional or formary operator?; is written of the form Expression! Expression? Expression 3. Il Expression is 2 true than Expression 2 will be executed also expression 3 will be All the operators mentioned above have a set UA.

The substance of the sub order is given by the OPERATOR PRECEDENCE CHART, given in table 3.

Precedence	Operator	Description	Associativity
1	++ () []> (type){list}	Suffix/postfix increment and decrement Function call Array subscripting Structure and union member access Structure and union member access through pointer Compound literal(C99)	Left-to-right
2	++ ! ~ (type) * & sizeof _Alignof	Prefix increment and decrement Unary plus and minus Logical NOT and bitwise NOT Type cast Indirection (dereference) Address-of Size-of Alignment requirement(C11)	Right-to-left SigmaPi Academy
3	* / %	Multiplication, division, and remainder	Left-to-right
4	+-	Addition and subtraction	
5	<< >>	Bitwise left shift and right shift	
6	< <= > >=	For relational operators < and ≤ respectively For relational operators > and ≥ respectively	
7	== !=	For relational = and = respectively	
8	&	Bitwise AND	
9	^	Bitwise XOR (exclusive or)	
10	I.	Bitwise OR (inclusive or)	
11	88	Logical AND	
12	11	Logical OR	
13	?:	Ternary conditional	Right-to-Left
14	= += -= *= /= %= <<= >>= &= ^= =	Simple assignment Assignment by sum and difference Assignment by product, quotient, and remainder Assignment by bitwise left shift and right shift Assignment by bitwise AND, XOR, and OR	
15	,	Comma	Left-to-right