Operators and Expression

Operators: Perform specific task Example Addition, subtraction, Multiplication, division, Modulus

Expression: is followed by an equal sign and can contain more than one operator.

C = A + B ---- > The equal sign made this an Expression

C = A + B ----→ The plus is an Operator. Performing Addition

In C, we have 4 types of operators

1. Arithmetic
2. Logical
3. Relational
4. Bitwise

Arithmetic: are expressed using arithmetic operators which are:

Operator	Result
+	Addition (also unary plus)
–	Subtraction (also unary minus)
*	Multiplication
/	Division
%	Modulus
++	Increment
+=	Addition assignment
– =	Subtraction assignment
*=	Multiplication assignment
/=	Division assignment
%=	Modulus assignment
– –	Decrement

Logical: are expressed using logical operators which includes

Operator	Description
>	greater than
<	less than
==	equal to
<=	less or equal to
>=	greater or equal to
!=	Inot equal to

Relational Operators

Relational Operators

Operator	Use	Description
`>`	`op1 > op2`	Returns `true` if `op1` is greater than `op2`
`>=`	`op1 >= op2`	Returns `true` if `op1` is greater than or equal to `op2`
`<`	`op1 < op2`	Returns `true` if `op1` is less than `op2`
`<=`	`op1 <= op2`	Returns `true` if `op1` is less than or equal to `op2`
`==`	`op1 == op2`	Returns `true` if `op1` and `op2` are equal
`!=`	`op1 != op2`	Returns `true` if `op1` and `op2` are not equal

Bitwise

Operators	Description	Use
&	Bitwise AND	op1 & op2
|	Bitwise OR	op1 | op2
^	Bitwise Exclusive OR	op1 ^ op2
~	Bitwise Complement	~op
<<	Bitwise Shift Left	op1 << op2
>>	Bitwise Shift Right	op1 >> op2
>>>	Bitwise Shift Right zero fill	op1 >>> op2

Precedence among operators

This deals with the way or the arrangement in which compiler execute instruction among operators. Some operations will have to be perform first before the other. The same principle holds

Precedence	Associativity	Operator	Description
18	Left-to-right	()	Parentheses (grouping)
17	Left-to-right	*f*(args...)	Function call
16	Left-to-right	*x*[index:index]	Slicing
15	Left-to-right	*x*[index]	Array Subscription
14	Right-to-left	**	Exponentiation
13	Left-to-right	~*x*	Bitwise not
12	Left-to-right	+*x* -*x*	Positive, Negative
11	Left-to-right	* / %	Multiplication Division Modulo
10	Left-to-right	+ -	Addition Subtraction
9	Left-to-right	<< >>	Bitwise left shift Bitwise right shift
8	Left-to-right	&	Bitwise AND
7	Left-to-right	^	Bitwise XOR
6	Left-to-right	|	Bitwise OR
5	Left-to-right	in, not in, is, is not, <, <=, >, >=, <>, == !=	Membership Relational Equality Inequality
4	Left-to-right	not *x*	Boolean NOT
3	Left-to-right	and	Boolean AND
2	Left-to-right	or	Boolean OR
1	Left-to-right	lambda	Lambda expression

A Compiler: Is a software that help us to execute our codes. There are different types of compliler for different programming language. For **C** and **C ++** you can use **DEV C++. you can google more on this, feel free.**

Which means that all our codes will be written and be executed on **Dev C++** User environment. Type on your browser ***download DEV C++ Make sure you download the latest version***