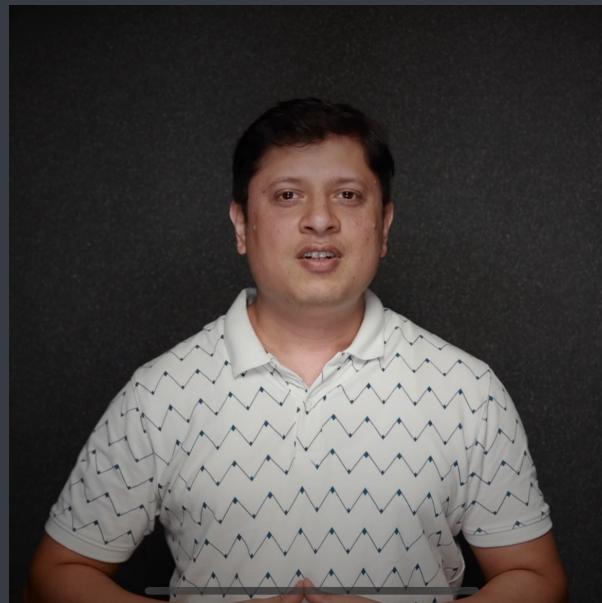


C Language

Operators in C Language



Saurabh Shukla (MySirG)

Agenda

- ① Arithmetic Instruction
- ② Classification of operators
- ③ Unary operators
- ④ Arithmetic operators
- ⑤ Bitwise operators
- ⑥ Relational operators
- ⑦ Logical operators
- ⑧ Assignment operators

Arithmetic Instruction

An instruction which is used to manipulate data using operators, is known as Arithmetic Instruction.

Operands (data)

↙ ↓
3 + 4
↑
operator

3 + 4 * 5
3 + 20
23

Classification of Operators

- ① Unary operators +, -, ++, --, sizeof()
- ② Arithmetic operators *, /, %, +, -
- ③ Bitwise Operators &, |, ^, ~, >>, <<
- ④ Relational Operators <, >, <=, >=, ==, !=
- ⑤ Logical Operators !, &&, ||
- ⑥ Conditional Operator ?:
- ⑦ Assignment Operators =, +=, -=, *=, /=, %=

operands

- ① unary
- ② Binary
- ③ Ternary

Unary Operators

$+, -, ++, --$

$+3, -5$

Increment Operator $++$

$\text{int } x=5;$ $\boxed{\begin{matrix} x \\ 786 \end{matrix}}$

$\text{printf}(" \%d", x);$ 5

$x++;$ \rightarrow post increment $\rightarrow x = x+1$

$\text{printf}(" \%d", x);$ 6

$++x;$ \rightarrow pre increment

$\text{printf}(" \%d", x);$ 7

$x+1$
 $x++$

$++x$ ↑
 $x++$ ↓

Decrement Operator $--$

$x--;$ post decrement \downarrow $x = x-1$

$--x;$ pre decrement \uparrow

Find Output of the following program?

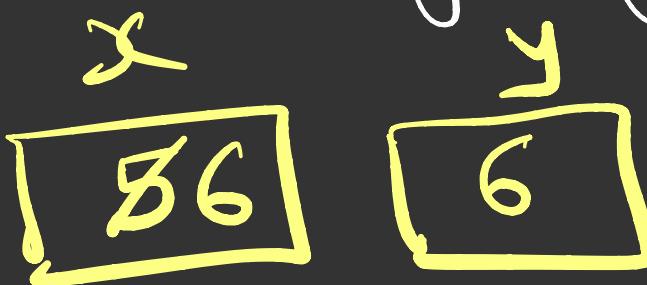
```
#include<stdio.h>
int main()
{
    int x=5, y;
    y=x++;
    printf("%d %d", x, y);
}
```



6 5

Find Output of the following program?

```
#include<stdio.h>
int main()
{
    int x=5, y;
    y=++x;
    printf("%d %d", x, y);
}
```



6 6

Unary Operators

sizeof()

- ① Data type ✓
- ② Variable ✓
- ③ Constant ✓

int x;

x = sizeof(float);

printf("%d", x); 4

x = sizeof(double);

printf("%d", x); 8

x = sizeof(char);

printf("%d", x); 1

```
int x, y;
float m;
char ch;
double d1;
```

```
8 x = sizeof(d1);
1 x = sizeof(ch);
4 x = sizeof(y);
4 x = sizeof(m);
```

```
4 x = sizeof(35);
8 x = sizeof(4.7);
4 x = sizeof('A');
```

Real constants are by default
double type
character constants are by default
of type int

Integer
constants are
by default of
type int
=

Arithmatic Operators (L to R)

* / %

$a \checkmark * b / c$

5 % . 2 1

+ -

$a \checkmark / b * c$

20 % . 4 0

3 + 4 7

$a + b \checkmark * c$

3 % . 4 3

3 - 4 -1

Integer operator Integer
= Integer

3 * 4 12

3 / 4 0

10 / 3 3

15 / 5 3

3.0 / 4 0.75

3 / 4.0 0.75

3.0 / 4.0 0.75

15.0 / 5 3.0

modulus operator cannot
be applied on real constants

3.5 % . 2 error

$a / 10 \rightarrow$ a without last digit

$a \% 10 \rightarrow$ last digit of a

Bitwise Operators

8 | ^ ~ , >>, <<
 AND 'OR' XOR NOT Right Shift Left Shift

Binary digit is called
 a bit.
 0 & 1 are bits

0 & 0 → 0	0 0 → 0	0 ^ 0 → 0	~ 0 → 1
0 & 1 → 0	0 1 → 1	0 ^ 1 → 1	~ 1 → 0
1 & 0 → 0	1 0 → 1	1 ^ 0 → 1	
1 & 1 → 1	1 1 → 1	1 ^ 1 → 0	

int x = 25 & 72;

25 = 00000000 00000000 00000000 00011001

72 = 00000000 00000000 00000000 01001000

8 = 00000000 00000000 00000000 00001000

$x = 35 \& 12;$

$$\begin{array}{r} 35 = 0010\ 0011 \\ 12 = 0000\ 1100 \\ \hline 0 = 0000\ 0000 \end{array}$$

$x = 23 \mid 47;$

$$\begin{array}{r} 23 = 0001\ 0111 \\ 47 = 0010\ 1111 \\ \hline 63 = 0011\ 1111 \end{array}$$

$x = 25 \gg 2;$

$25 = \underbrace{000000000}_{6} \underbrace{00000000}_{0} \underbrace{00000000}_{0} \underbrace{00011001}_{1}$

6

$x = 12 \ll 3; \quad 96$

$00000000 \quad 00000000 \quad 00000000 \quad 00001100$

$00000000 \quad 00000000 \quad 00000000 \quad 0000110000$

int x = ~5;

$$K = b_1 \leftarrow 2's$$
$$-K = b_2 \leftarrow 2's$$

5 = 00000000 00000000 00000000 00000101

$\sim 5 = \underline{1}1111111111111111111111111111010$

MSB 0 \rightarrow +ve 6 = 000...0110
sign bit 1 \rightarrow -ve -6

$$\begin{array}{r} 1101 \\ 0010 \\ +1 \\ \hline 0011 \end{array}$$

i's \rightarrow 1101
 \sim 0010
+ 1
 \hline
2's 0011

$$2S = b_1 \leftarrow 2's$$
$$-2S = b_2 \leftarrow 2's$$

Relational Operators (L to R)

<, >, <=, >=. (Comparison operators)

==, !=

True → 1
False → 0

x = 2 > 4; 0

x = 4 <= 4; 1

x = 4 == 3; 0

x = 5 > 4 > 3; 0

Logical Operators

! NOT(unary)

&& AND

|| OR

Every non-zero value
is True

zero is False

! True \rightarrow False

! False \rightarrow True

$x = !5 ;$ 0

$x = !5 > -2 ;$

!

$x > 0 \& \& y > 0$

Exp1 || Exp2

Exp1 && Exp2

T && T → T
T && F → F
F && X → F

F || F → F
F || T → T
T || X → T

Assignment Operators R to L

=

$x = 4;$

= Assignment

$4 = x;$ error

== equal to

variable = anything

$y = x = 3;$

int $x=5;$

x
5

$x = x + 3;$

↑
Container

↑
Content

Compound Assignment Operators

$+ = , - = , * = , / = , \% =$

int $x = 5;$

$x += 4;$

$x = x + 4$

$x -= 3;$

$x = x - 3$

$x *= 2;$

$x = x * 2$

$x /= 6;$

$x = x / 6$

$x \% = 5;$

$x = x \% 5$

int $x = 5;$
 $x *= 3 + 4;$
 $x *= 7$
 $x = 35$

int $x = 5;$
 $x = x * 3 + 4;$
 $x = 15 + 4$
 $x = 19$