CS & IT [redacted]

ENGINEERING

C Programming

C Tokens [redacted]

Lecture No.- 04

By- Satya sir

Recap of Previous Lecture

- Binary operators
  (Two operands)
  - Arithmetic ✓
  - Logical ✓
  - Bitwise & shift ✓
  - Relational
  - Assignment

Topics to be Covered

- Relational Operators
- Assignment operator
- Ternary operator
- Operator Predence & Associativity

Relational operators (or) Comparison operators

- These operators verify relation among 2 operands by comparing their values.

Operator	Meaning
<	Less than
<=	Less than or Equal to
>	Greater than
>=	Greater than or Equal to
!=	Not Equal to
==	Equal to

These operators compare both operands, returns True (or) 1 if relation satisfy

Otherwise, returns False (or) 0.

Examples :

```
a = -1,  b = 2,  c = 0,  d = 2

i =  a >= b ;     -1 >= 2   False/0
j =  c < d ;       0 < 2    True/1
k =  b <= d ;      2 <= 2   True/1
l =  a > c ;      -1 > 0    False/0
x =  a != c ;     -1 != 0   True/1
y =  b == c ;      2 == 0   False/0

Printf("%d %d %d %d %d %d", i, j, k, l, x, y);
```

O/P: 0 1 1 0 1 0

Assignment operator (=)

- It assigns (gives, allocates, makes), RHS value to LHS

Operand1 = operand 2 ;

Assigns op2 value to op1

- Operand 1 must be a variable
- Operand 2 can be any valid Expression.

$a = b+c$ Valid

$i = 5+4/3-2$ Valid

$x+y = z$ Invalid

Compound (or) Short-hand assignment

Operand 1 operator = Operand 2

⇒ Operand1 = operand1 Operator Operand2

Ex:

1) $a += b \Rightarrow a = a + b$
2) $x /= y \Rightarrow x = x/y$
3) $i <<= 3 \Rightarrow i = i << 3$
4) $p -= q+r \Rightarrow p = p-(q+r)$

$\Rightarrow p = p-q-r$

Ternary Operator ⇒ which Perform operation Using 3 Operands.

⇒ ?: Symbol is Called Ternary operator

⇒ Syntax: $\underset{\text{Operand 1}}{\underline{\text{Expression 1}}}$? $\underset{\text{operand 2}}{\underline{\text{Expression 2}}}$: $\underset{\text{operand 3}}{\underline{\text{Expression 3}}}$

- Operand 1 is Evaluated.

  If it's Result is TRUE — Operand 2 Evaluated, Operand 3 is ignored.

  Otherwise ⇒ FALSE — Operand 2 ignored, Operand 3 is Evaluated

- As operand 2 or operand3, which should be Evaluated is Conditional, ?: Operator is also known as Conditional Operator.
- ?: order will be Right to Left.

Examples

① a = 5, b = 9, c;

$c = a > b \; ? \; a : b;$

$5 > 9 \; ? $ FALSE

$C = b$

$\Rightarrow C = 9$

② $a = 1, \; b = 0, \; c = -2$

$d = \underbrace{a}_{\text{op1}} \; ? \; \underbrace{b > c}_{\text{op2}} : \underbrace{a < c}_{\text{op3}} ;$

$1 \; ? \Rightarrow 1$ is Non-Zero $\Rightarrow$ True

$d = b > c \Rightarrow d = 0 > -2$ (True)

$\Rightarrow \boxed{d = 1}$

③ $p = -3, \; q = 2, \; r = 0$

$x = p \; ? \; q \; ? \; r \; ? \; q : p : r : q ;$

p

p

$r \; ? \; q : p \Rightarrow 0 \; ?$ False $\Rightarrow p$

$q \; ? \; p : r \Rightarrow 2 \; ?$ True $\Rightarrow p$

$p \; ? \; p : q \Rightarrow -3 \; ?$ True $\Rightarrow p$

$x = p \Rightarrow \boxed{x = -3}$

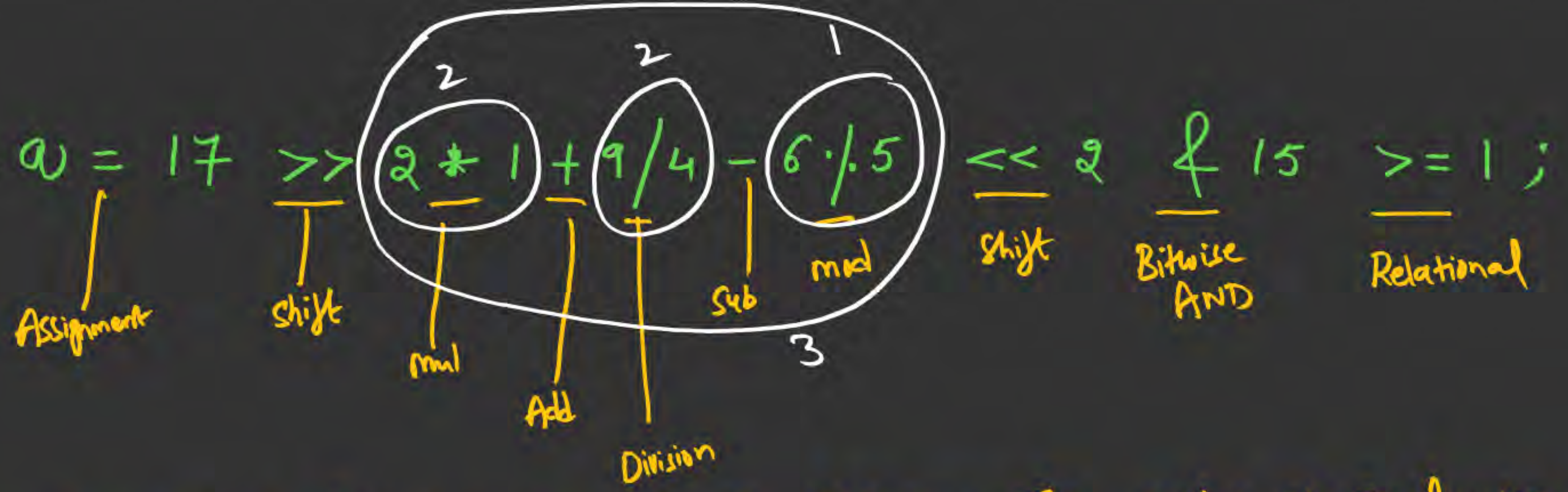
④ $a = 5, \; b = 7, \; c = 0, \; d = -2, \; e = 0$

$P = \underbrace{a <= b}_{\text{op1}} \; ? \; \underbrace{b != c}_{\text{op2}} : \overbrace{c == e \; ? \; a > d : c != a \; ? \; a < d : b != e}^{\text{op3}} ;$

$= 5 <= 7$ True $\Rightarrow$ op3 ignored

$P = b != c$

$P = 7 != 0 \Rightarrow P =$ True $\Rightarrow \boxed{P = 1}$



- When, Two or more operators Exist in the expression, Then, order of Evaluation is given by Operator Precedence (Pre-defined) [Priority or Rank]

- If Two or more operators, with Same Precedence, Then order of Evaluation is given by Associativity (Left to Right (or) Right to Left)

Ex:

$2 * 1 = 2$ $2 + 2 = 4$

$9/4 = 2$ $4 - 1 = 3$

$6 \% 5 = 1$ $17 >> 3 = 17/2^3 = 2$

$2 << 2 = 2 * 2^2 = 8$ $8 \& 1 = 0$

$15 >= 1$ True $= 1$

$\boxed{a = 0}$

Operator Precedence & Associativity Table

Operator	Name	Precedence	Associativity
( ), [ ], ++, --, ., ->	Brackets Postfix member access	1	L To R
++, -- +, -, *, &, !, ~, sizeof	[Prefix] Unary	2	R To L
* / %	Arithmetic	3	L To R
+ -		4	L To R
<<, >>	Shift	5	L To R
<, <=, >, >=	Relational	6	L To R
!=, ==		7	L To R
&	Bitwise AND	8	L To R
^	Bitwise XOR	9	
|	Bitwise OR	10	
&&	Logical AND	11	L To R
||	Logical OR	12	
? :	Ternary	13	R To L
=, +=, -=, *=, <<=, /=, ...	Assignment	14	R To L
,	Comma	15	L To R

2 mins Summary

- Special Symbols : The symbols that are not operators

  Ex: @, $, {, }, #

- Relational operators
- Assignment Operator
- Ternary
- Operator Precedence & Associativity

THANK - YOU