Computer Science & IT

C Programming

Data Types & Operators

Lecture No. 05

By- Abhishek Sir

Recap of Previous Lecture

Topic Relational operator

Topic Increment & Decrement operator

Topic Assignment operator

Topic Logical operator (AND, OR, NOT)

Topic

Slide

Topics to be Covered

Topic logical operator

Topic Concept of logical operator

Topic Bit wise operator

Topic Scope of variable

Topic

Slide

Toipc: Logical Operator

1. Arithmetic
2. Relational
3. Logical

precedence table

Slide

Toipc: Logical Operator &&

(++, --) post pre - unary

Logical Negation - unary

AND

x	y	x&&y
0	0	0
0	1	0
1	0	0
1	1	1 ✓

OR

x	y	x||y
0	0	0
0	1	1 ✓
1	0	1 ✓
1	1	1 ✓

x	!x
0	1
1	0

Slide

Toipc: Logical Operator (AND) &&

```
#include <stdio.h>
int main(void) {
    int a = 20; ✓
    int b = 30; ✓
    printf("%d", a > 10 && b > 10);
}
```

$$(\overset{1}{20>10} \;\&\&\; \overset{1}{30>10})$$

$\boxed{1}$

Slide

Toipc: Logical Operator (OR) ||

```
#include <stdio.h>
int main(void) {
  int a = 20;
  int b = 5;
  printf("%d", a > 10 || b > 10);

}
```

20>10 || 5>10

1 || 0 = 1

Slide

Toipc: Logical Operator (NOT) !

$(20>10) \,\&\&\, (5<4) \;||\; (4 \,!\!= 2)$

$(1 \,\&\&\, 0) \;||\; 1$

$0 \,||\, 1 = \boxed{1}$

! — Negation - High

&& : AND

|| : OR

Left to Right

example

$10 == 2 \;||\; 4 \,!\!= 5 \,\&\&\, 6 < 10$

$0 \,||\, 1 \,\&\&\, 1 = 0 \,||\, 1 = \boxed{1}$

Slide

Toipc: Logical Operator (NOT) !

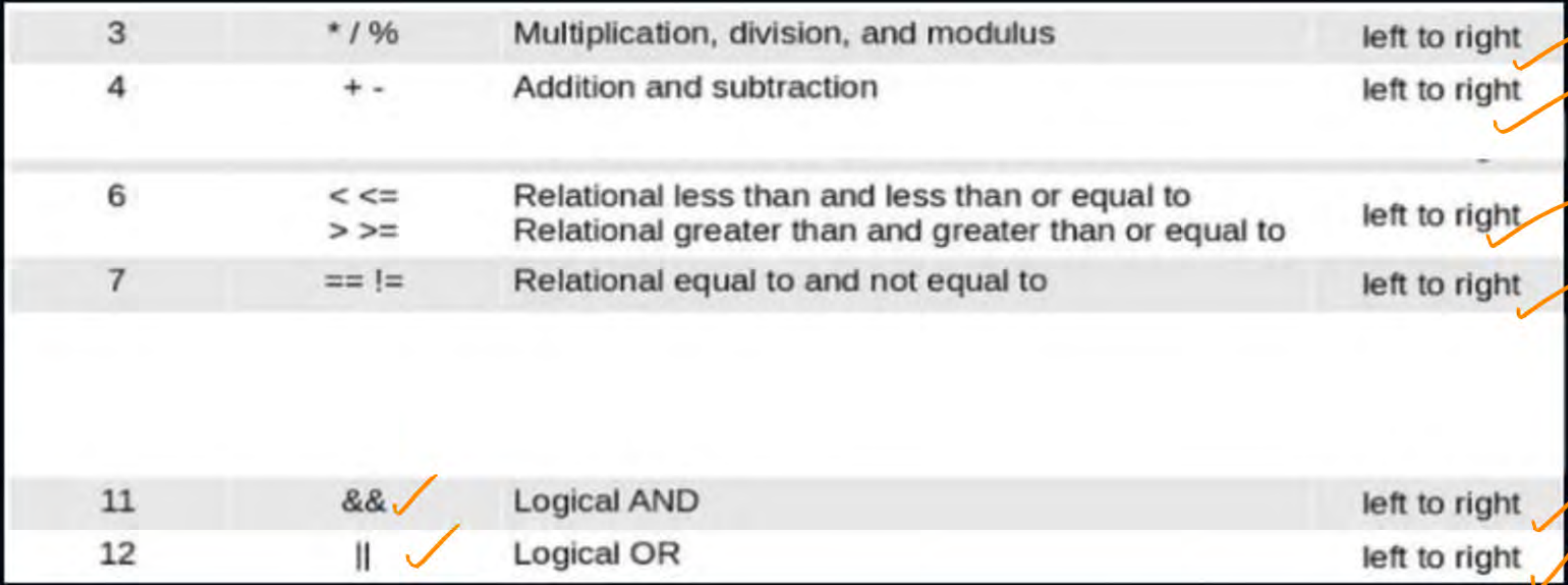
```
#include <stdio.h>
int main(void) {
  int a = 20;
  int b = 5;
  printf("%d", !(a < 10 ));
}
```

$!(20 < 10)$

$!(0) = \boxed{1}$

Slide

Toipc: Precedence



3	* / %	Multiplication, division, and modulus	left to right ✓
4	+ -	Addition and subtraction	left to right ✓
6	< <= > >=	Relational less than and less than or equal to Relational greater than and greater than or equal to	left to right ✓
7	== !=	Relational equal to and not equal to	left to right ✓
11	&& ✓	Logical AND	left to right ✓
12	|| ✓	Logical OR	left to right ✓

Slide

Toipc:Question

क्यो नही
why??

$0 \&\& (\quad) = 1$

```
#include <stdio.h>
  int main () {
 int a =(( 5+6!=10 )==8)&&((6*2/4>10)==(10>6));
  printf("%d", a);
 }
```

0 &&

$((11 != 10) == 8)$

The output of the program is ____

$(1 == 8)$

Slide

Toipc:Question

```
#include <stdio.h>
     int main ()  {
        int a =5+5!=10|| 6+4;
     printf("%d", a);
    }
```

Non Zero is true

b = !10;

b assigned with 0

= 10!=10||10

The output of the program ___

Non zero is True zero false

= 0||10

= 1

Slide

Toipc: Question

Short circuit code

```
#include <stdio.h>
    void main ()  {

        int x = 1, y = 0, z = 5;

        int a = (x && y) && ++z;

        printf("%d", z);


    }
```

(A) 6

(B) 5

(C) 0

(D) 1

AND operation

if one value is zero

the answer is zero

Second expression

Not execute

0 && (++z)

will Not execute

(1&&0) && ++z

&&

0||1 = 1

(5==6 && (3==3)) || 6>4

0 &&

0 || 6>4 = 1

Slide

Shoot circuit code

In evaluation of Logical AND if one value evaluates to 0 then Rest expression does not execute.

In evaluation of Logical OR if one value evalues to toue or 1 then Rest expression will Not execute.

Toipc: Question

```
#include <stdio.h>
    void main ()  {
        int x = 1, y = 1, z = 6;
                  1 ||++z
        int a = x && y || ++z;
                1 && 1
        printf("%d", z);
    }
```

(A) 6

(B) 5

(C) 0

(D) 1

Because x&&y is 1

|| ++z will always

evaluates to 1

Hence shoot Circuit

Code Rule will Not execute ++z.

Slide

Bit-wise Operator

t.me/Abhisheksharmapw

Bit wise operator

Bit wise AND	&
Bitwise OR	|
Bitwise Negation	~
Bitwise Leftshift	<<
Bitwise Rightsht	>>

Bit wise XOR ^

x y	$x \oplus y$
0 0	0
0 1	1
1 0	1
1 1	0

Slide

Bit-wise Operator

Operators	Meaning of operators
&	Bitwise AND ✓
|	Bitwise OR
^	Bitwise exclusive OR
~	Bitwise complement
<<	Shift left
>>	Shift right

Slide

Bit-wise Operator

What is the output the program

```
#include <stdio.h>
    int main ()  {
      int x = 5, y=17, z
      z = x&y;
      printf("%d", z);
 }
```

(A) 1 ✓

(B) 21

(C) 2

(D) -6

Sign

```
000101
010001
------
000001
```

```
2 | 17
2 | 8   1
2 | 4   0
2 | 2   0
2 | 1   0
    0   1
```

Slide

Bit-wise Operator

What is the output the program

```
#include <stdio.h>
    int main ()  {
      int x = 5, y=17, z
      z = x&y;
      printf("%d", z);
 }
```

(A) 1 (B) 21

(C) 2 (D) -6

Bit-wise Operator

What is the output the program

```
#include <stdio.h>
    int main ()  {
      int x = 5, y=17, z
      z = x|y;
      printf("%d", z);
 }
```

(A) 1 [✓ (B) 21]

(C) 2 (D) -6

$$\begin{array}{r} 000101 \\ 010001 \\ \hline 010101 \end{array}$$

$= 1\times2^4+0\times2^3+1\times2^2$
$+0\times2^1+1\times2^0$

$= 16+4+1 = \underline{21}$

Slide

Bit-wise Operator

What is the output the program

```
#include <stdio.h>
    int main ()  {
      int x = 5, y=17, z
      z = x|y;
      printf("%d", z);
 }
```

(A) 1 **(B) 21**

(C) 2 (D) -6

Slide

Bit-wise Operator

What is the output the program

```
#include <stdio.h>
    int main ()  {
      int x = 5, y=17, z
      z = x^y;
      printf("%d", z);
 }
```

(A) 1 (B) 21

(C) 20 (D) -6

$$\begin{array}{r} 000101 \\ 010001 \\ \hline 010100 \end{array}$$

$$1\times2^4+0\times2^3+1\times2^2+0\times2^1+0\times2^0$$

$$=16+4=20$$

Slide

Bit-wise Operator

What is the output the program

```
#include <stdio.h>
    int main ()  {
      int x = 5, y=17, z
      z = x^y;
      printf("%d", z);
 }
```

(A) 1 (B) 21

(C) 20 (D) -6

Slide

Bit-wise Operator

What is the output the program

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

2's complement

(A) 1

(B) 21

(C) 20

[✓ (D) -6]

0101 (5)

1010

$1\times -2^3 + 0\times 2^2 + 1\times 2^1 + 0\times 2^0$

$-8+0+2+0 = \boxed{-6}$

Bit-wise Operator

What is the output the program

```
#include <stdio.h>
    int main ()  {
        int x = ~~5~~, z
        z = ~x;10
        printf("%d", z);
    }
```

[redacted] 21

[redacted] -6

NAT

$5 \rightarrow -6$

$10 \rightarrow -11$

$0\,1\,0\,1\,0$ x is positive

$1\,0\,1\,0\,1$ $-(x+1)$

$1 \times -2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$

$-16 + 4 + 1 = -11$

HW (A) $x = -10$ $\sim x$?   (C) $x = -64$ $\sim x$?

(B) $x = -23$ $\sim x$?

Slide

Bit-wise Operator

```
#include<stdio.h>
  int main(){
        char a = 8;
        int k;
        k =a<<3;
        printf("%d", k);
        return 0;
  }
```

In case of update overflow may occur

(A) 1

✓ (B) 64

(C) 20

(D) -6

$x = 5$

$z = x << 1 \Rightarrow 5 \times 2^1 = 10$

$z = x << 2 \Rightarrow 5 \times 2^2 = 20$

$z = x << 3 \Rightarrow 5 \times 2^3 = 5 \times 8 = 40$

$K = a << 3 = 8 \times 2^3$

$= 8 \times 8 = 64$

8

if in Range

$a << K = \boxed{a \times 2^K}$

Slide

Bit-wise Operator

```
#include<stdio.h>
  int main(){
        char a = 8;
        int k;
        k =a<<3;
        printf("%d", k);
        return 0;
  }
```

(A) 1

[✓ (B) 64]

(C) 20

(D) -6

Slide

Bit-wise Operator

```
#include<stdio.h>
    int main(){
            char a = 64;
            int k;
            k =a>>3;
            printf("%d", k);
            return 0;
    }
```

(A) 1

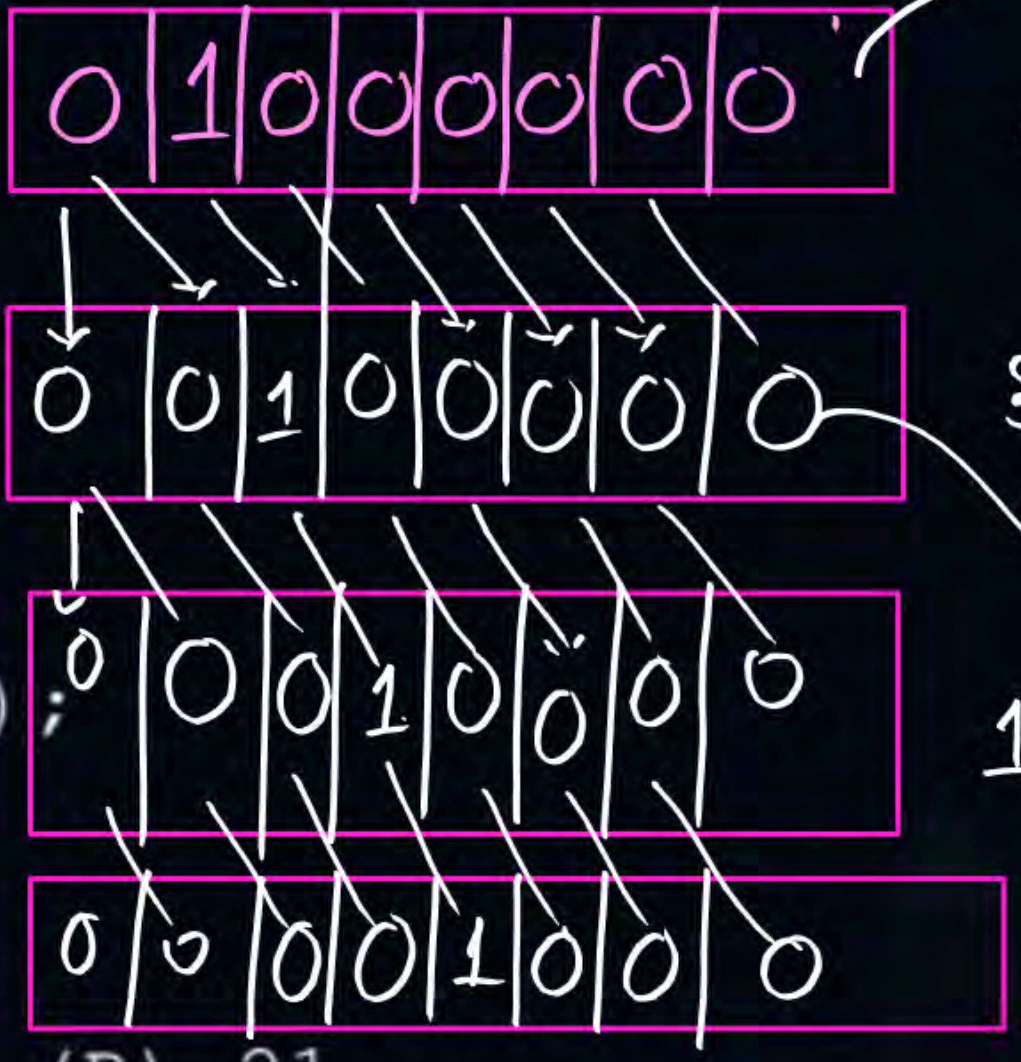
(B) 21

[✓ (C) 8]

(D) -6

a>>K

a is positive No = $\boxed{a \times 1/2^K}$



gone

64

32 >> 1 result

gone

16 >>2 result

8 >> 3 result

Slide

2 mins Summary

Topic

Logical operator

Topic

shoot circuit code

Topic

Bit wise operator

Topic

Topic

Slide

THANK - YOU