

FCCS202 COE V1 Section 3 x Meet - vbh-gdvy-iza x Something went wrong x Something went wrong x +

meet.google.com/vbh-gdvy-iza?authuser=0

REC R Ritu Sibbal is presenting M MAHIKA KUSHW... and 61 more 12:16 PM You

- C supports a full bitwise operators.
- Since C was designed to take the place of assembly language for most programming tasks, it needed to be able to support many operations that can be done in assembler, including operations on bits.
- *Bitwise operation* refers to testing, setting, or shifting the actual bits in a byte or word, which correspond to the standard **char** and **int** data types and variants
- Bitwise operations cannot be used on **float**, **double**, **long double**, **void**, or other more complex types

bcxgzslm1s ^

Raise hand Turn on captions Ritu Sibbal is presenting

Type here to search

12:16 04-01-2021

Participants: You, Ritu Sibbal, VISHAL VERMA, AMAN SONI, ROHIT SHARMA

FCC502 COE VI Section 3 Meet - vbh-gdy-iza Something went wrong Something went wrong

meet.google.com/vbh-gdy-iza?authuser=0

REC Ritu Sibbal is presenting MEGHNA and 61 more 68 12:16 PM You

# Bitwise Operators

Operator	Action
&	AND
	OR
^	Exclusive OR (XOR)
~	One's complement (NOT)
>>	Shift right
<<	Shift left

You R Ritu Sibbal V VISHAL VERMA A AMAN SONI ROHIT SHARMA

Type here to search 12:58 04-01-2021

Browser tabs: FCCS202 COE VI Section 3, Meet - vbh-gdvy-iza, Something went wrong, Something went wrong

Address bar: meet.google.com/vbh-gdvy-iza?authuser=0

Meeting controls: REC, Ritu Sibbal is presenting, PRATHAM JAIN and 62 more, 12:17 PM, You

Chat with everyone

- Bitwise operations most often find application in device drivers— such as modem programs, disk file routines, and printer routines
- Bitwise operations can be used to mask off certain bits, such as parity. (The parity bit confirms that the rest of the bits in the byte are unchanged. It is often the high-order bit in each byte.)

Participants: You, Ritu Sibbal, VISHAL VERMA, AMAN SONI, ROHIT SHARMA

Actions: Raise hand, Turn on captions, Ritu Sibbal is presenting

System tray: Type here to search, 12:17, 04-01-2021



FCCS202 COE VI Section 3

Meet - vbh-gdvy-iza

Something went wrong

Something went wrong

meet.google.com/vbh-gdvy-iza?authuser=0

Ritu Sibbal is presenting

HARSHIT SINGH and 62 more

12:26 PM

- In the following example, assume that **ch** had received the character A and had the parity bit set:

Parity bit

↓

1	1	0	0	0	0	0	1
0	1	1	1	1	1	1	1
<hr/>							
0	1	0	0	0	0	0	1

**ch** containing an "A" with parity set

127 in binary

do bitwise AND

"A" without parity

bcxgzslm1s

Microphone

Camera

Screen

Raise hand

Turn on captions

Ritu Sibbal is presenting

Type here to search

Taskbar icons

System tray

FCCS002 CDE VI Section 3

Meet - vbh-gdvy-iza

Something went wrong

Something went wrong

meet.google.com/vbh-gdvy-iza?authuser=0

REC

R

Ritu Sibbal is presenting

ABHISHEK KUM... and 61 more

68

12:29 PM

You

00:02:35

You

R

Ritu Sibbal

V

VAIBHAV SHARMA

V

VISHAL VERMA

S

SHASHANK MISHRA

- The bitwise OR, as the reverse of AND, can be used to set a bit. Any bit that is set to 1 in either operand causes the corresponding bit in the outcome to be set to 1
- For example, the following is  $128 \mid 3$ :

10000000

00000011

|

—————

10000011

128 in binary

3 in binary

bitwise OR

result

• An exclusive OR, usually abbreviated XOR, will set a bit on, if and only if the bits being compared are different. For example,  $127 \wedge 120$  is

0 1 1 1 1 1 1	127 in binary
0 1 1 1 1 0 0	120 in binary
$\wedge$ <hr/>	bitwise XOR
0 0 0 0 0 1 1	result

00:03:15

You

R

Ritu Sibbal

RITIK YADAV

V

VISHAL VERMA

S

SHASHANK MISHRA

FCFS002 COE V1 Section 3 Meet - vbh-gdy-iza Something went wrong Something went wrong

meet.google.com/vbh-gdy-iza?authuser=0

Ritu Sibbal is presenting

## Difference between Relational, logical and bitwise operators

- Relational and logical operators always produce a result that is either true or false,
- whereas the similar bitwise operations may produce any arbitrary value in accordance with the specific operation.
- Bitwise operations may produce values other than 0 or 1, while logical operators will always evaluate to 0 or 1.

bcxzgslms

People (68)

Chat

VINIT 12:11 PM  
is attendance done

You

Ritu Sibbal

ROHIT SHARMA

VISHAL VERMA

RAHUL SHARMA

bcxzgslms

Ritu Sibbal is presenting

logical XOR

Type here to search

12:33 04-01-2021



FCSS002 CDE VI Section 3 x Meet - vbh-gdvy-iza x Something went wrong x Something went wrong x +

meet.google.com/vbh-gdvy-iza?authuser=0

R Ritu Sibbal is presenting ABHIJEET SHAR... and 62 more 12:39 PM You

- The bit-shift operators,  $\gg$  and  $\ll$ , move all bits in a variable to the right or left as specified.
- The general form of the shift-right statement is  
*variable  $\gg$  number of bit positions*
- The general form of the shift-left statement is  
*variable  $\ll$  number of bit positions*
- As bits are shifted off one end, zeroes are brought in the other end.
- A shift is not a rotate. That is, the bits shifted off one end do not come back around to the other. The bits shifted off are lost

00:13:07 You V VISHAL VERMA R Ritu Sibbal AKHIL DUBEY K KAUSTAV SARIKAR

bcxzsimsimls ^

Microphone Mute Video Off Screen Share

FFCS002 CDE VI Section 3 x Meet - vbh-gdvy-iza x Something went wrong x Something went wrong x +

meet.google.com/vbh-gdvy-iza?authuser=0

R Ritu Sibbal is presenting ANANT and 62 more 12:40 PM You

## *The ? Operator*

- C contains a powerful and convenient operator that replaces certain statements of the if-then-else form. The ternary operator ? takes the general form  
$$Exp1 \text{ ? } Exp2 : Exp3;$$
- where  $Exp1$ ,  $Exp2$ , and  $Exp3$  are expressions.
- The ? operator works like this:  $Exp1$  is evaluated. If it is true,  $Exp2$  is evaluated and becomes the value of the expression. If  $Exp1$  is false,  $Exp3$  is evaluated, and its value becomes the value of the expression. For example, in  
$$x = 10;$$
$$y = x > 9 \text{ ? } 100 : 200;$$

00:13:57

You

V

VISHAL VERMA

R

Ritu Sibbal

N

NISHCHAY HASIJA

K

KAUSTAV SARIKAR

FCCS902 COE VI Section 3 x Meet - vbh-gdy-iza x Something went wrong x Something went wrong x +

meet.google.com/vbh-gdy-iza?authuser=0

R Ritu Sibbal is presenting V VISHAL VERMA and 61 more 12:44 PM You

y is assigned the value 100. If x had been less than 9, y would have received the value 200. The same code written using the **if-else** statement is

```
x = 10;  
if(x>9) y = 100;  
else y = 200;
```

You  
ARYAN CHAUHAN  
R Ritu Sibbal  
N NISHCHAY HASLIA  
K KAUSTAV SARKAR

FFCS002 CDE VI Section 3

Meet - vbh-gdvy-iza

Something went wrong

Something went wrong

+

meet.google.com/vbh-gdvy-iza?authuser=0

ARCHIT BANSAL and 62 more

12:46 PM

You

R

Ritu Sibbal is presenting

## The & and \* Pointer Operators

- A *pointer* is the memory address of an object.
- A *pointer variable* is a variable that is specifically declared to hold a pointer to an object of its specified type.
- Pointers are one of C's most powerful features, and they are used for a wide variety of purposes.
- For example, they can provide a fast means of referencing array elements.
- They allow functions to modify their calling parameters. They support linked lists, binary trees, and other
- dynamic data structures.
- We now briefly cover the two operators that are used to manipulate pointers.

00:19:26

You

ARYAN CHAUHAN

V

VISHAL VERMA

RITIK YADAV

R

Ritu Sibbal