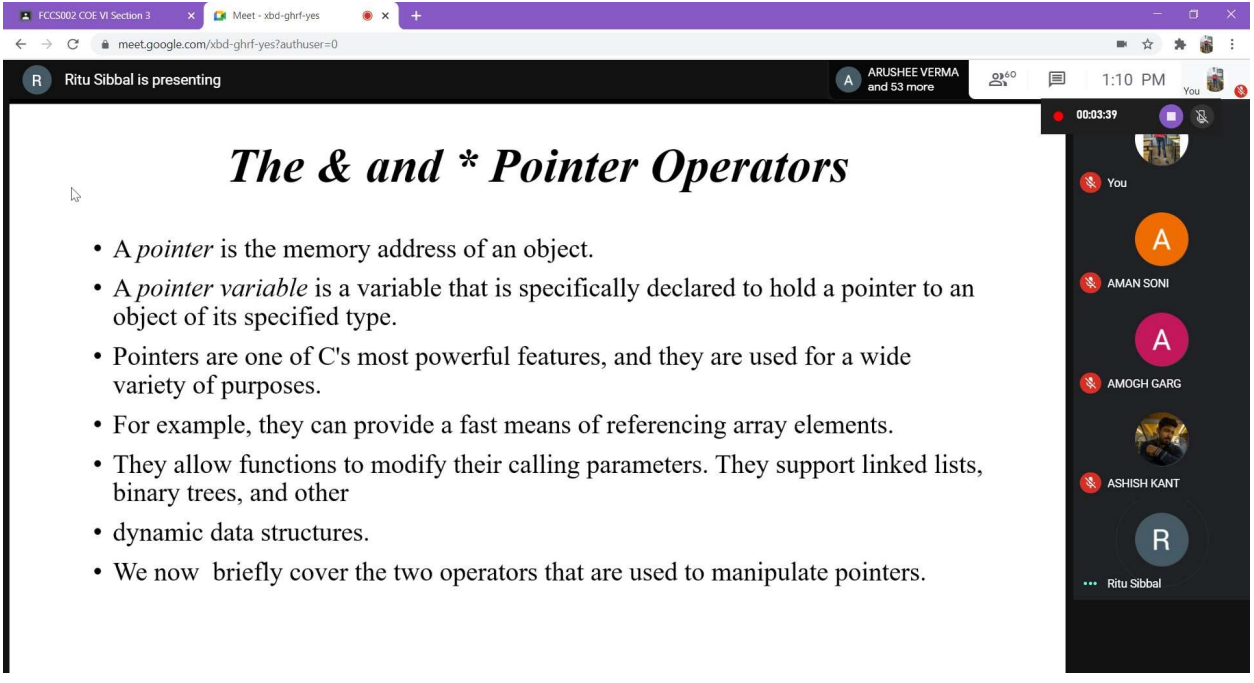
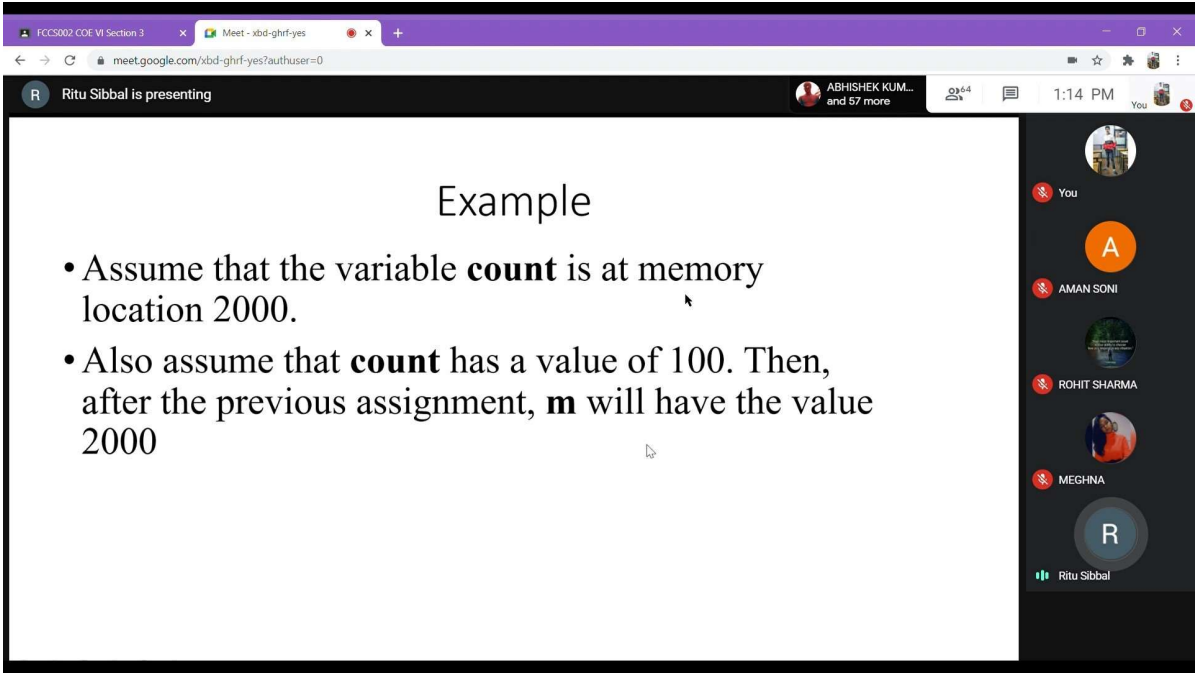
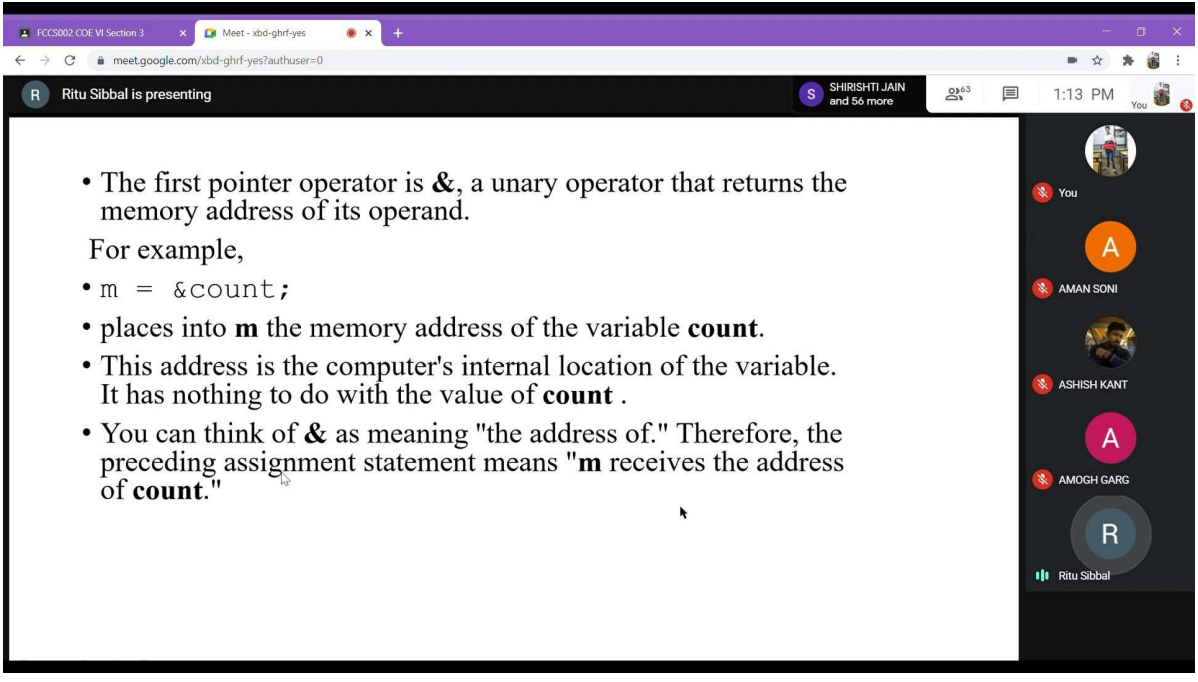
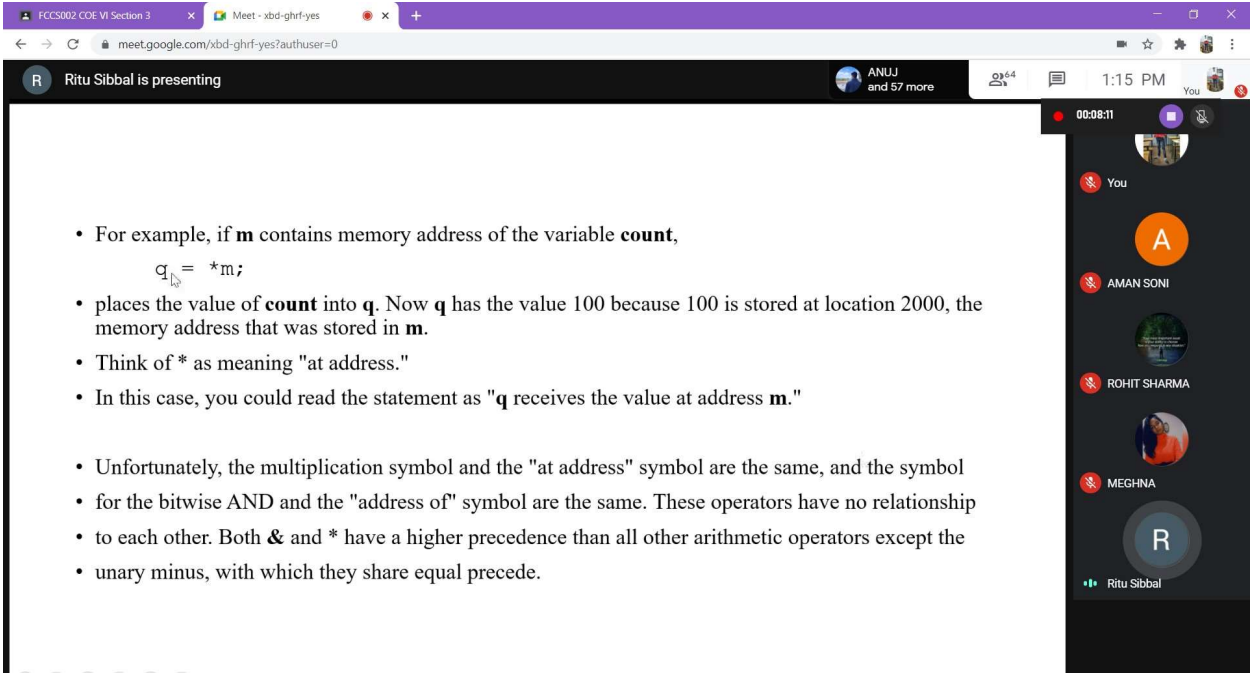
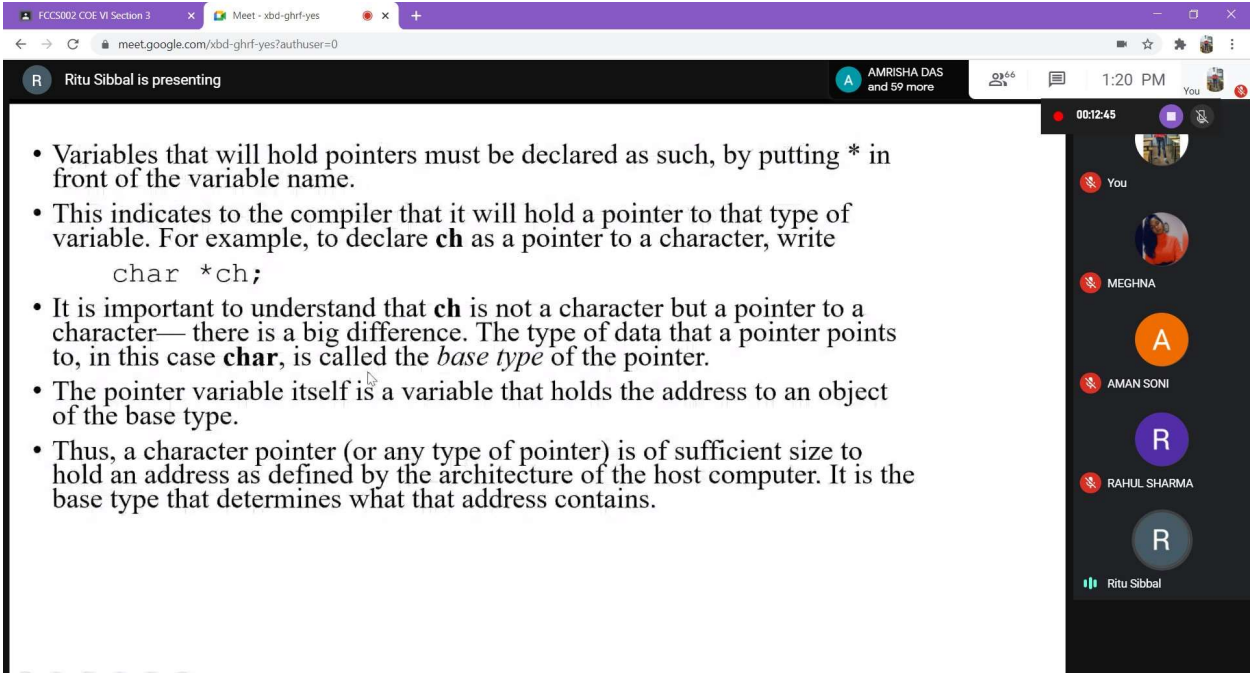
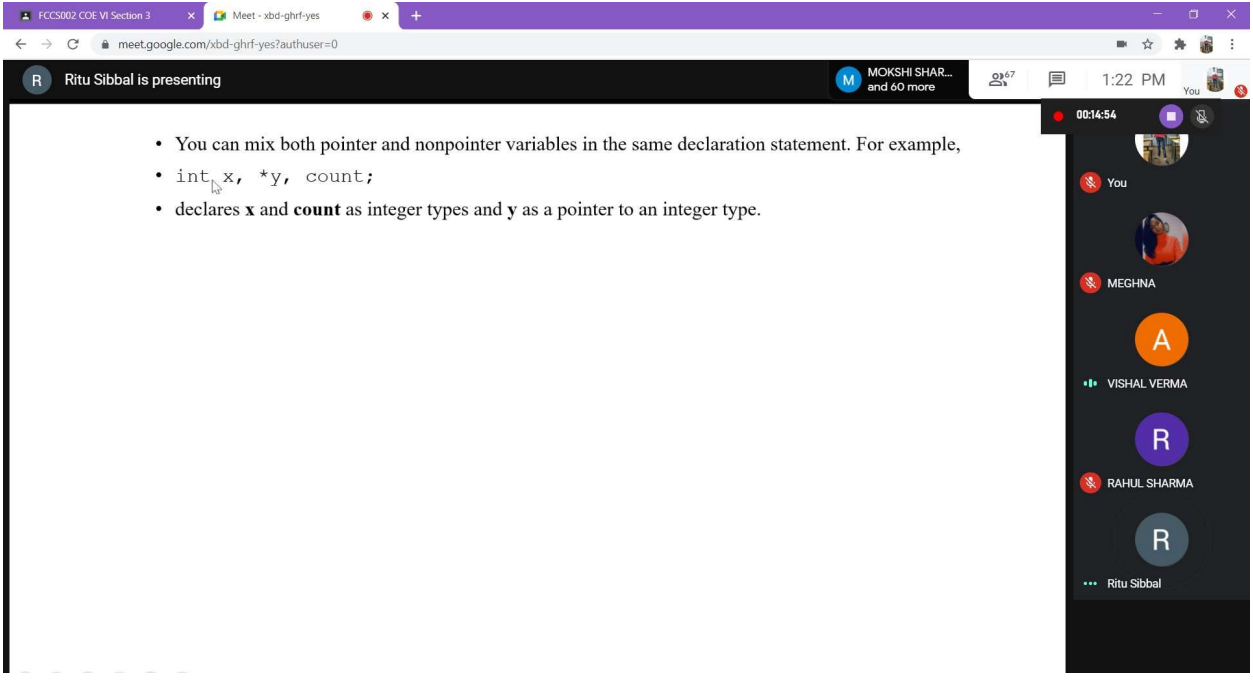

FCCS002 COE VI Section 3
Meet - xbd-ghrf-yes
meet.google.com/xbd-ghrf-yes?authuser=0
Ritu Sibbal is presenting
ARUSHEE VERMA and 53 more
1:10 PM
You
00:03:39
The & and * Pointer Operators
A pointer is the memory address of an object.
A pointer variable is a variable that is specifically declared to hold 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.
You
AMAN SONI
AMOGH GARG
ASHISH KANT
Ritu Sibbal

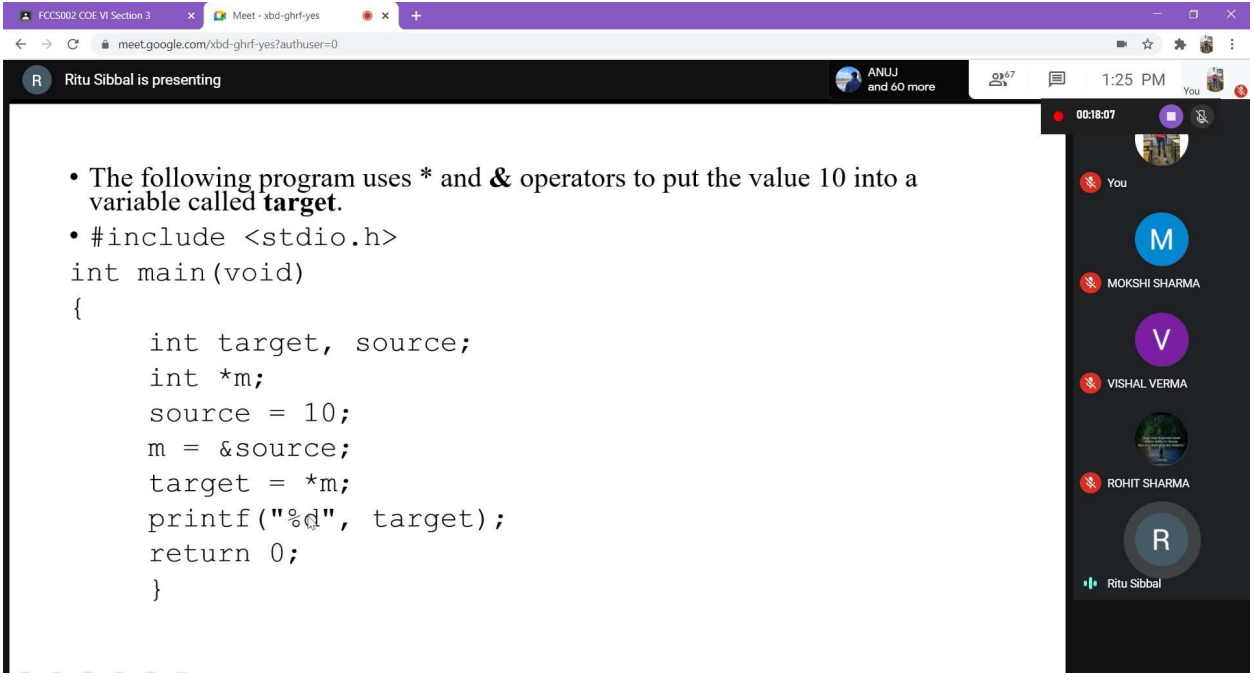

FCCS002 COE VI Section 3
Meet - xbd-ghrf-yes
meet.google.com/xbd-ghrf-yes?authuser=0
R
Ritu Sibbal is presenting
ABHISHEK KUM...
and 57 more
64
1:14 PM
You
Example
• Assume that the variable count is at memory location 2000.
• Also assume that count has a value of 100. Then, after the previous assignment, m will have the value 2000
You
A
AMAN SONI
ROHIT SHARMA
MEGHNA
R
Ritu Sibbal

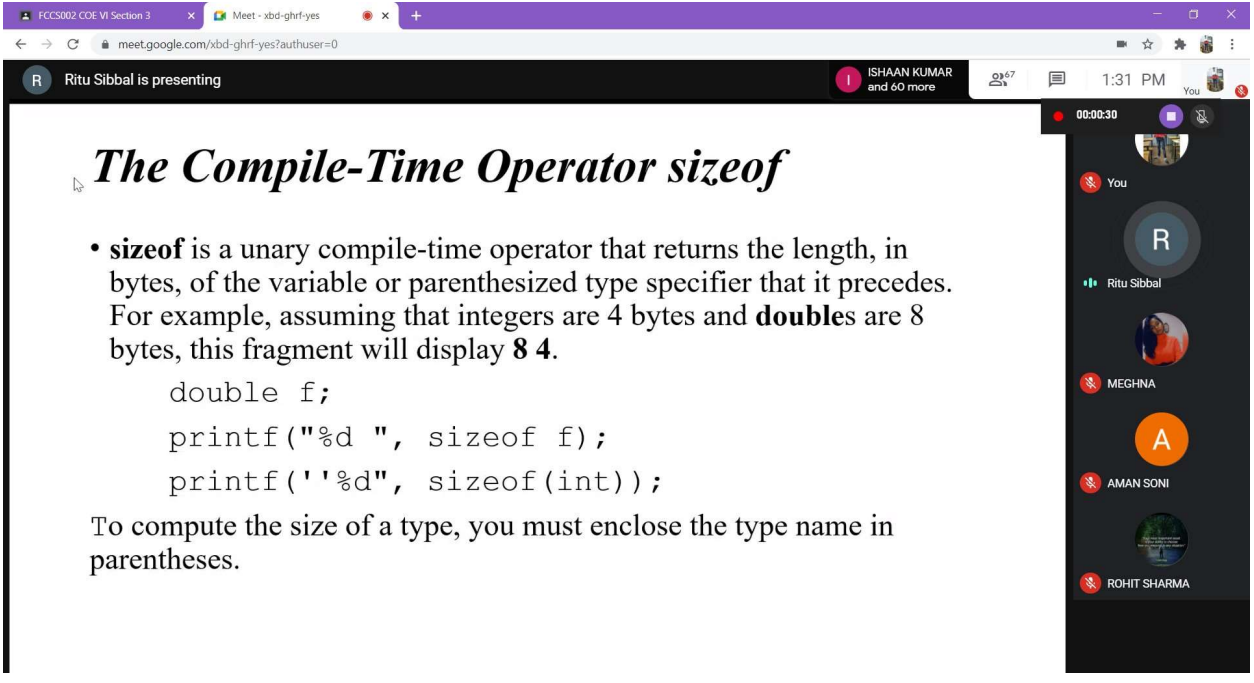

FCCS002-COE VI Section 3
Meet - xbd-ghrf-yes
meet.google.com/xbd-ghrf-yes?authuser=0
Ritu Sibbal is presenting
SHIRISHTI JAIN and 56 more
1:13 PM
You
• The first pointer operator is &, a unary operator that returns the memory address of its operand.
For example,
• m = &count;
• places into m the memory address of the variable count.
• This address is the computer's internal location of the variable. It has nothing to do with the value of count .
• You can think of & as meaning "the address of." Therefore, the preceding assignment statement means "m receives the address of count."
You
AMAN SONI
ASHISH KANT
AMOGH GARG
Ritu Sibbal

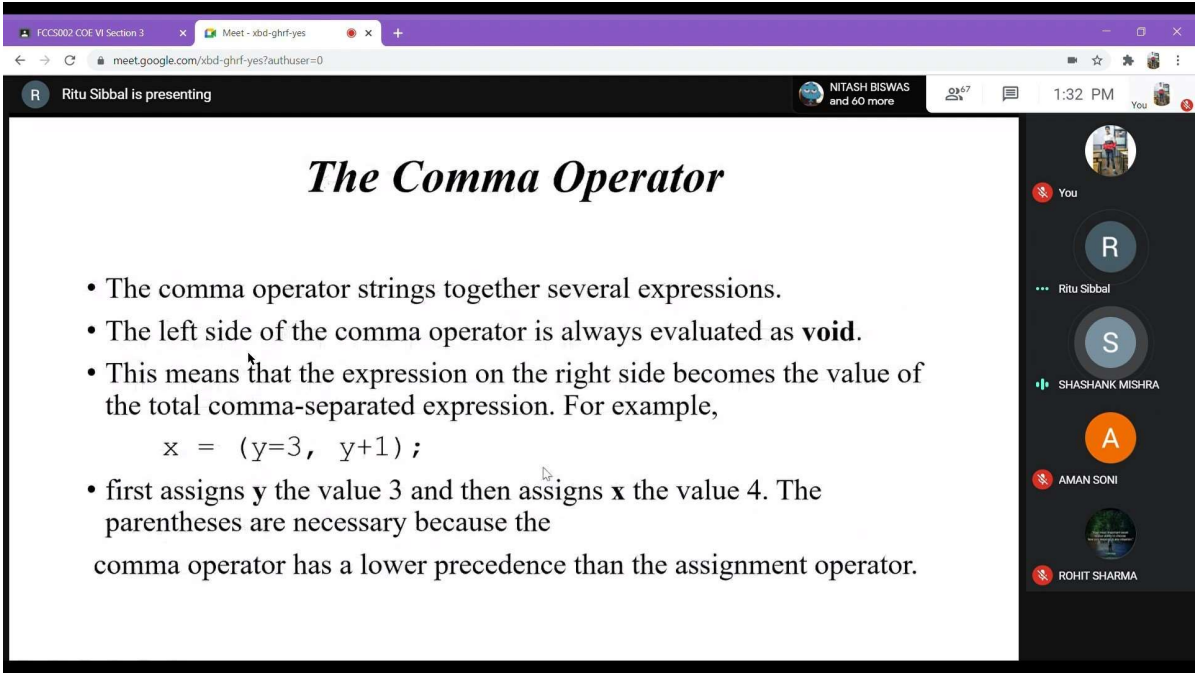

FCCS002 COE VI Section 3
Meet - xbd-ghrf-yes
meet.google.com/xbd-ghrf-yes?authuser=0
Ritu Sibbal is presenting
ANUJ and 57 more
1:15 PM
You
00:08:11
You
AMAN SONI
ROHIT SHARMA
MEGHNA
Ritu Sibbal
• For example, if **m** contains memory address of the variable **count**,
q = *m;
• places the value of **count** into **q**. Now **q** has the value 100 because 100 is stored at location 2000, the memory address that was stored in **m**.
• Think of * as meaning "at address."
• In this case, you could read the statement as "**q** receives the value at address **m**."
• Unfortunately, the multiplication symbol and the "at address" symbol are the same, and the symbol
• for the bitwise AND and the "address of" symbol are the same. These operators have no relationship
• to each other. Both **&** and * have a higher precedence than all other arithmetic operators except the
• unary minus, with which they share equal precede.

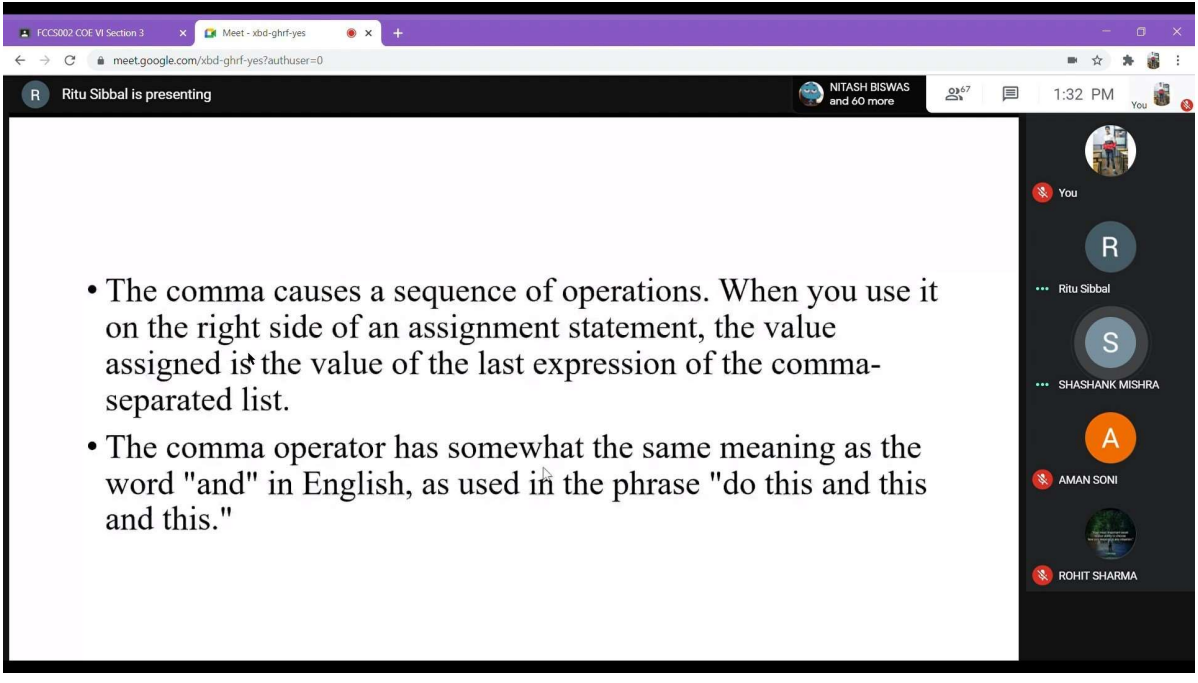

FCCS002 COE VI Section 3
Meet - xbd-ghrf-yes
meet.google.com/xbd-ghrf-yes?authuser=0
Ritu Sibbal is presenting
AMRISHA DAS and 59 more
1:20 PM
You
00:12:45
You
MEGHNA
AMAN SONI
RAHUL SHARMA
Ritu Sibbal
• Variables that will hold pointers must be declared as such, by putting * in front of the variable name.
• This indicates to the compiler that it will hold a pointer to that type of variable. For example, to declare ch as a pointer to a character, write
char *ch;
• It is important to understand that ch is not a character but a pointer to a character— there is a big difference. The type of data that a pointer points to, in this case char, is called the base type of the pointer.
• The pointer variable itself is a variable that holds the address to an object of the base type.
• Thus, a character pointer (or any type of pointer) is of sufficient size to hold an address as defined by the architecture of the host computer. It is the base type that determines what that address contains.


FCCS002 COE VI Section 3
Meet - xbd-ghrf-yes
meet.google.com/xbd-ghrf-yes?authuser=0
Ritu Sibbal is presenting
MOKSHI SHAR... and 60 more
67
1:22 PM
You
00:14:54
You
MEGHNA
A
VISHAL VERMA
R
RAHUL SHARMA
R
Ritu Sibbal
• You can mix both pointer and nonpointer variables in the same declaration statement. For example,
• int x, *y, count;
• declares x and count as integer types and y as a pointer to an integer type.


FCCS002 COE VI Section 3
Meet - xbd-ghrf-yes
meet.google.com/xbd-ghrf-yes?authuser=0
Ritu Sibbal is presenting
ANUJ and 60 more
1:25 PM
You
00:18:07
• The following program uses * and & operators to put the value 10 into a variable called target.
• #include <stdio.h>
int main(void)
{
int target, source;
int *m;
source = 10;
m = &source;
target = *m;
printf("%d", target);
return 0;
}
You
MOKSHI SHARMA
VISHAL VERMA
ROHIT SHARMA
Ritu Sibbal


FCCS002 COE VI Section 3
Meet - xbd-ghrf-yes
meet.google.com/xbd-ghrf-yes?authuser=0
Ritu Sibbal is presenting
ISHAAN KUMAR and 60 more
1:31 PM
You
00:00:30
You
Ritu Sibbal
MEGHNA
AMAN SONI
ROHIT SHARMA
The Compile-Time Operator sizeof
• sizeof is a unary compile-time operator that returns the length, in bytes, of the variable or parenthesized type specifier that it precedes. For example, assuming that integers are 4 bytes and doubles are 8 bytes, this fragment will display 8 4.
double f;
printf("%d ", sizeof f);
printf(''%d", sizeof(int));
To compute the size of a type, you must enclose the type name in parentheses.


FCCS002 COE VI Section 3
Meet - xbd-ghrf-yes
meet.google.com/xbd-ghrf-yes?authuser=0
Ritu Sibbal is presenting
NITASH BISWAS and 60 more
67
1:32 PM
You
The Comma Operator
• The comma operator strings together several expressions.
• The left side of the comma operator is always evaluated as void.
• This means that the expression on the right side becomes the value of the total comma-separated expression. For example,
x = (y=3, y+1);
• first assigns y the value 3 and then assigns x the value 4. The parentheses are necessary because the
comma operator has a lower precedence than the assignment operator.
You
R
Ritu Sibbal
S
SHASHANK MISHRA
A
AMAN SONI
ROHIT SHARMA


FCCS002 COE VI Section 3
Meet - xbd-ghrf-yes
meet.google.com/xbd-ghrf-yes?authuser=0
Ritu Sibbal is presenting
NITASH BISWAS and 60 more
67
1:32 PM
You
• The comma causes a sequence of operations. When you use it on the right side of an assignment statement, the value assigned is the value of the last expression of the comma-separated list.
• The comma operator has somewhat the same meaning as the word "and" in English, as used in the phrase "do this and this and this."
You
R
Ritu Sibbal
S
SHASHANK MISHRA
A
AMAN SONI
ROHIT SHARMA