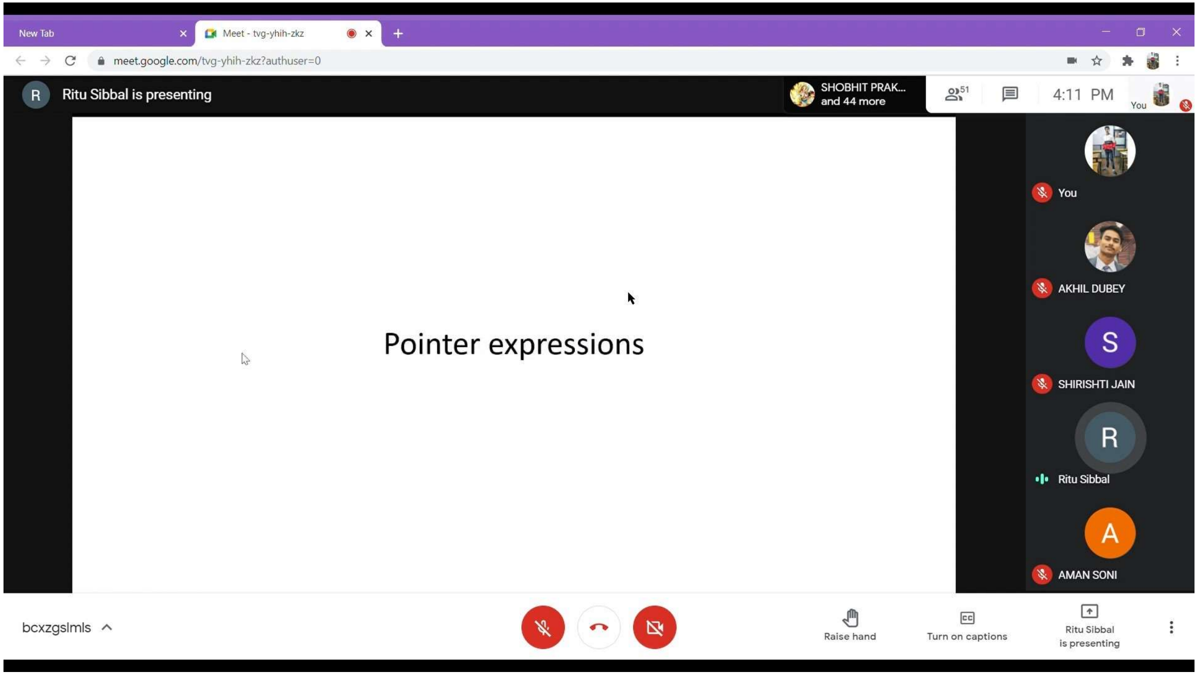
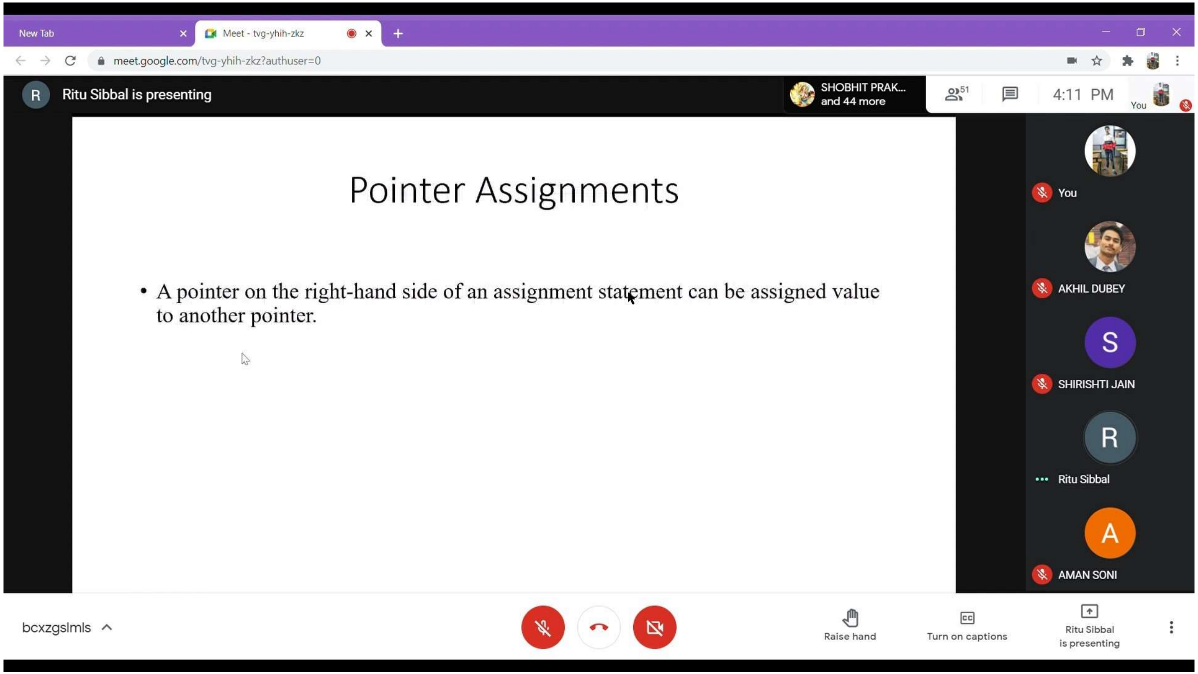
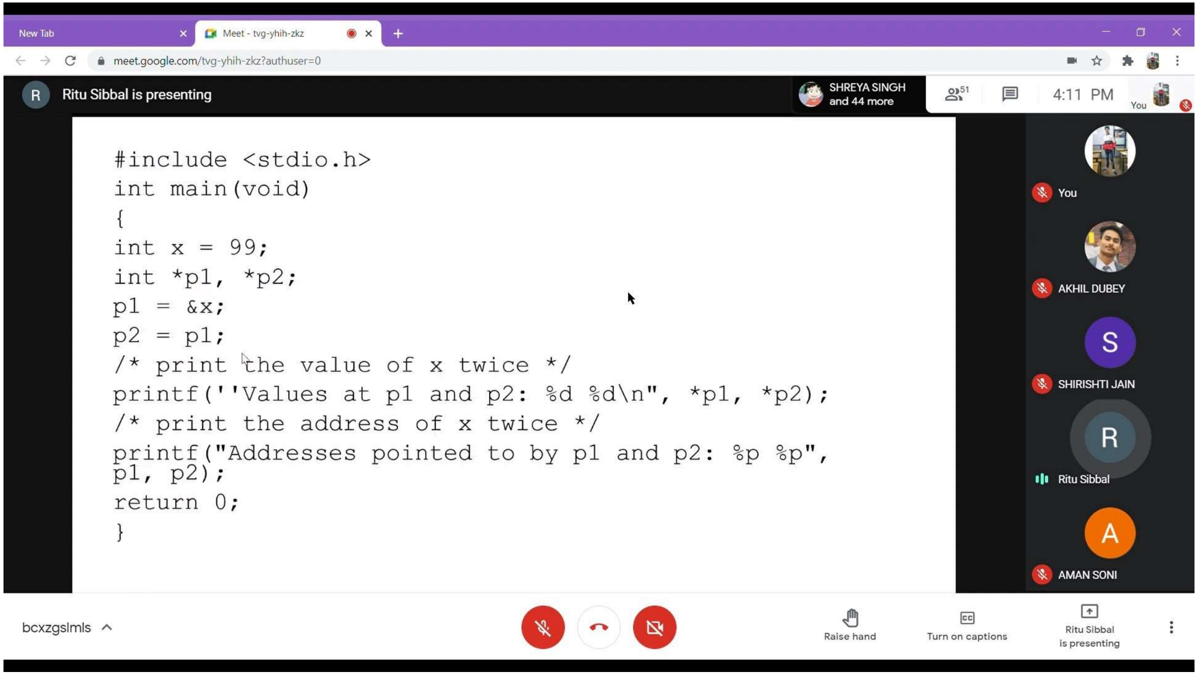
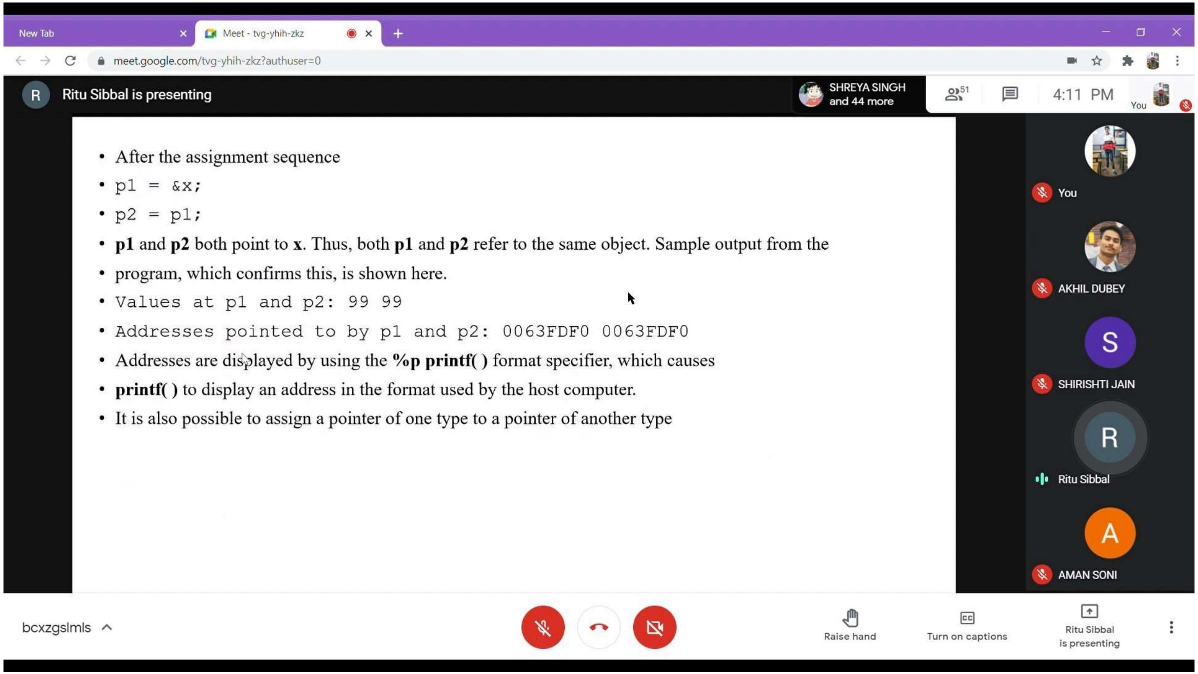
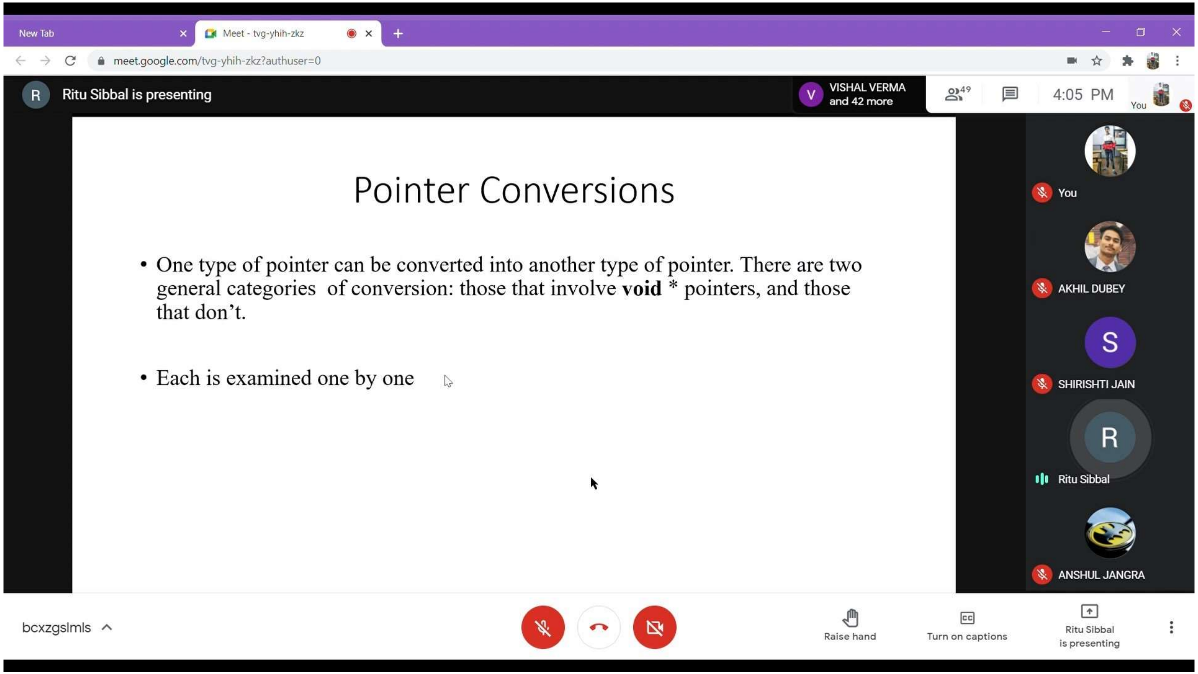

New Tab
Meet - tvg-yhih-zkz
meet.google.com/tvg-yhih-zkz?authuser=0
Ritu Sibbal is presenting
SHOBHIT PRAK...
and 44 more
4:11 PM
You
Pointer expressions
You
AKHIL DUBEY
SHIRISHTI JAIN
Ritu Sibbal
AMAN SONI
bcxzgslmls
Raise hand
Turn on captions
Ritu Sibbal
is presenting

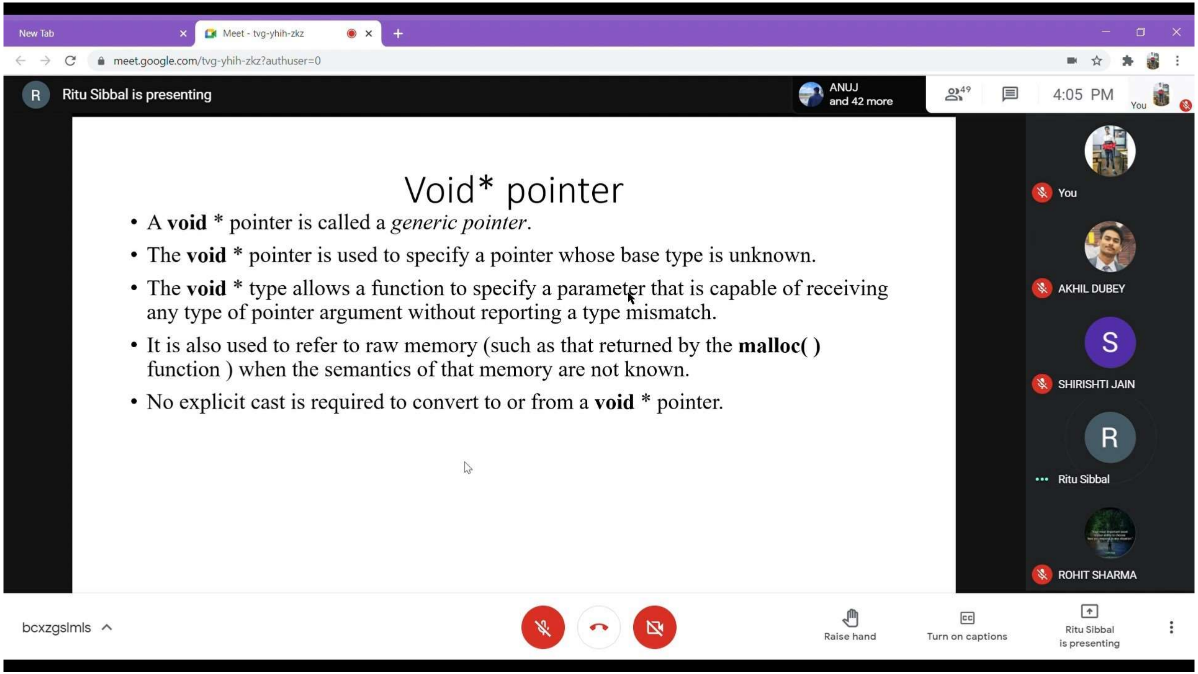

New Tab
Meet - tvg-yhih-zkz
meet.google.com/tvg-yhih-zkz?authuser=0
Ritu Sibbal is presenting
SHOBHIT PRAK...
and 44 more
51
4:11 PM
You
Pointer Assignments
• A pointer on the right-hand side of an assignment statement can be assigned value to another pointer.
You
AKHIL DUBEY
SHIRISHTI JAIN
Ritu Sibbal
AMAN SONI
bcxzgslmls
Raise hand
Turn on captions
Ritu Sibbal is presenting

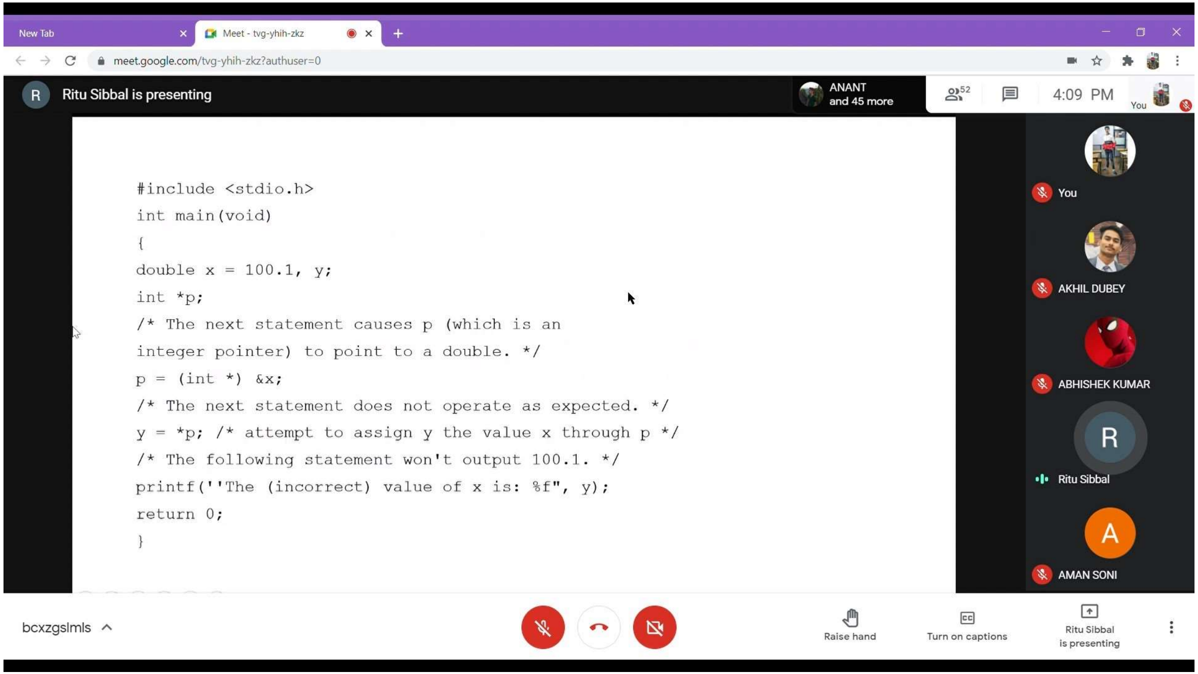

New Tab
Meet - tvg-yhih-zkz
meet.google.com/tvg-yhih-zkz?authuser=0
Ritu Sibbal is presenting
SHREYA SINGH and 44 more
4:11 PM
#include <stdio.h>
int main(void)
{
int x = 99;
int *p1, *p2;
p1 = &x;
p2 = p1;
/* print the value of x twice */
printf(''Values at p1 and p2: %d %d\n", *p1, *p2);
/* print the address of x twice */
printf("Addresses pointed to by p1 and p2: %p %p", p1, p2);
return 0;
}
You
AKHIL DUBEY
SHIRISHTI JAIN
Ritu Sibbal
AMAN SONI
bcxzgslmls
Raise hand
Turn on captions
Ritu Sibbal is presenting

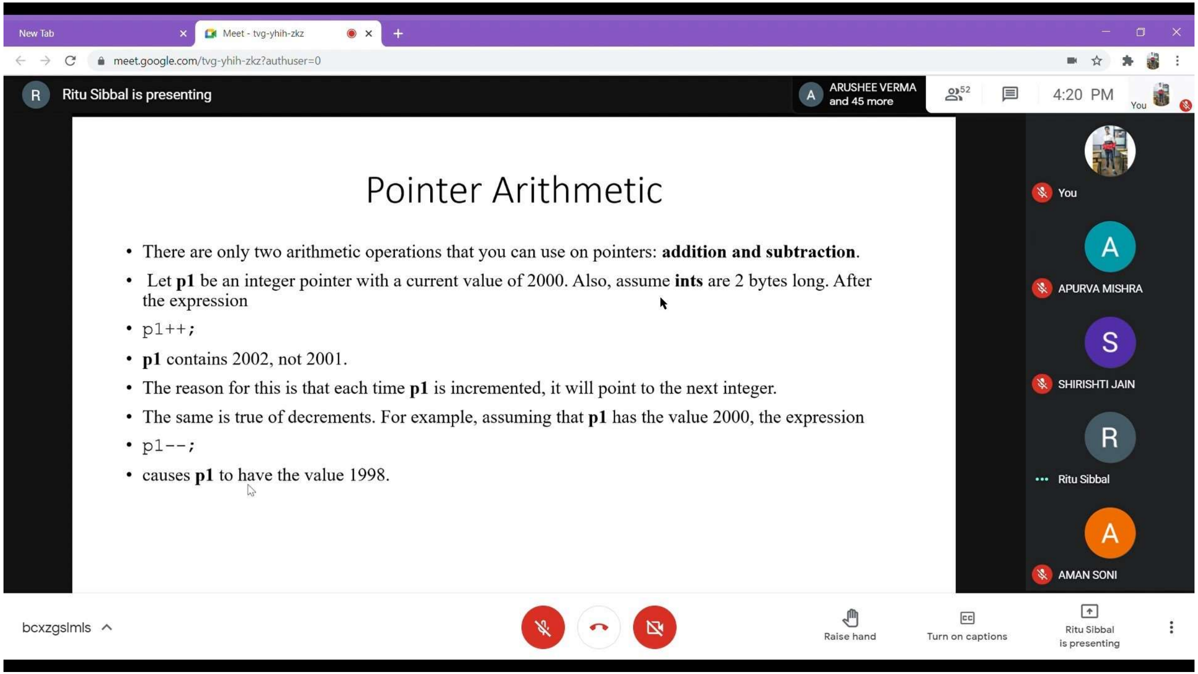

New Tab
Meet - tvg-yhih-zkz
meet.google.com/tvg-yhih-zkz?authuser=0
Ritu Sibbal is presenting
SHREYA SINGH and 44 more
4:11 PM
• After the assignment sequence
• p1 = &x;
• p2 = p1;
• **p1** and **p2** both point to **x**. Thus, both **p1** and **p2** refer to the same object. Sample output from the
• program, which confirms this, is shown here.
• Values at p1 and p2: 99 99
• Addresses pointed to by p1 and p2: 0063FDF0 0063FDF0
• Addresses are displayed by using the **%p printf( )** format specifier, which causes
• **printf( )** to display an address in the format used by the host computer.
• It is also possible to assign a pointer of one type to a pointer of another type
You
AKHIL DUBEY
SHIRISHTI JAIN
Ritu Sibbal
AMAN SONI
bcxzgslmls
Raise hand
Turn on captions
Ritu Sibbal is presenting

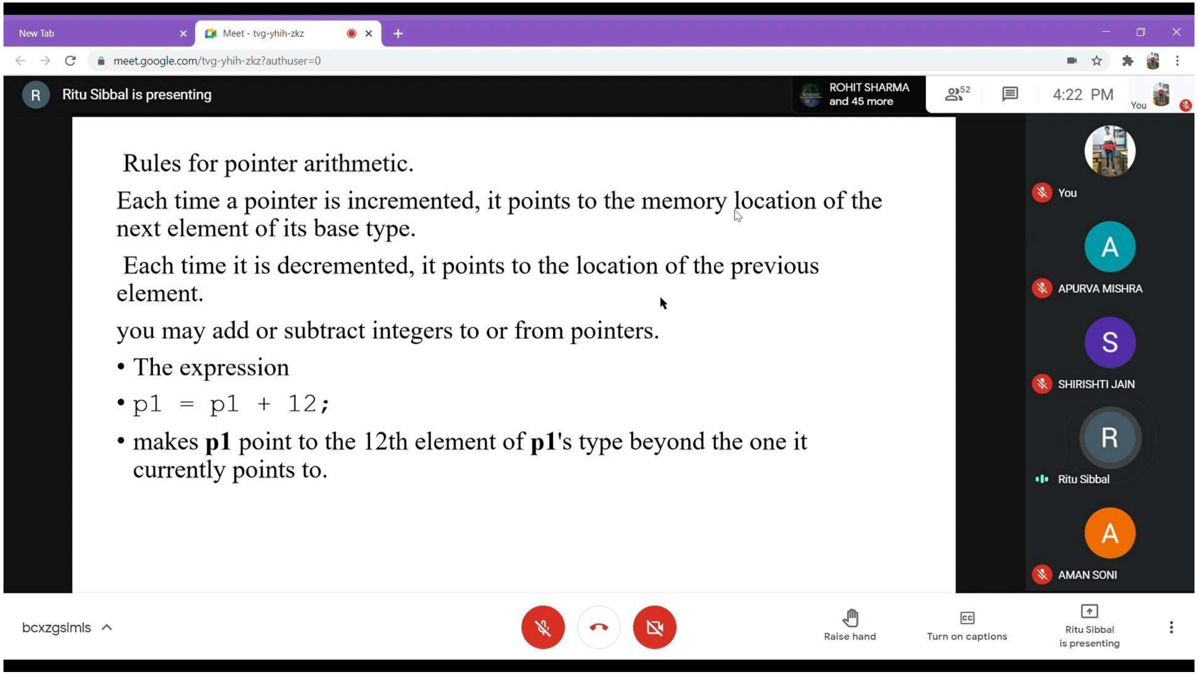


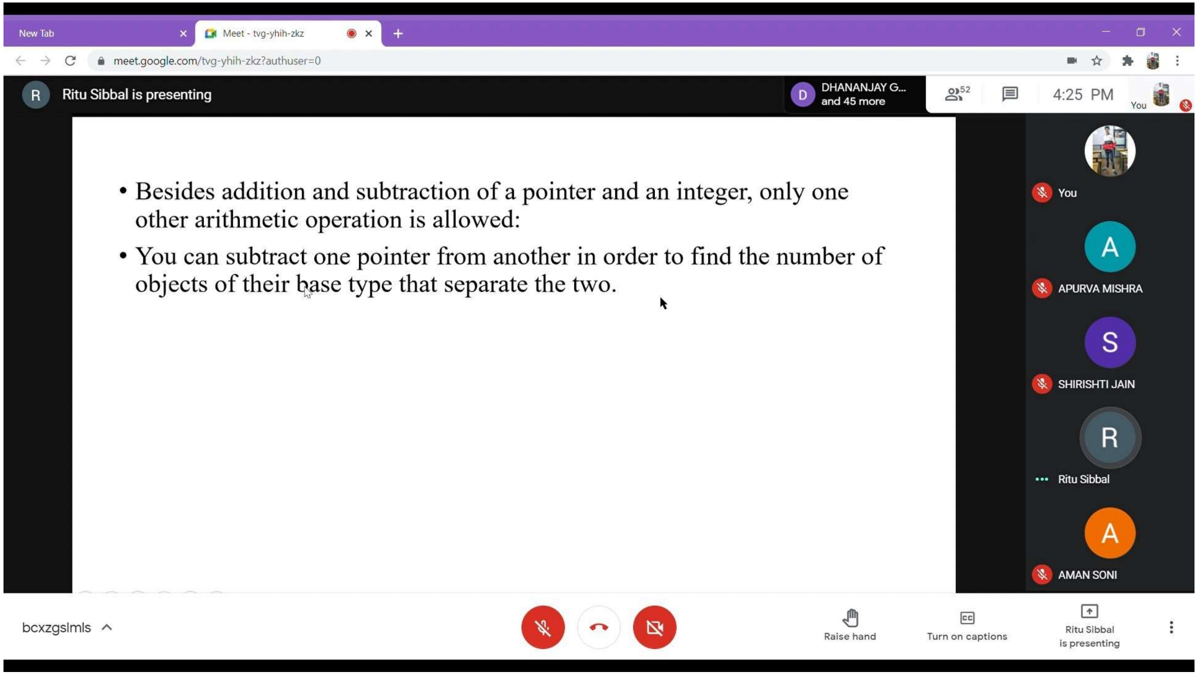
New Tab
Meet - tvg-yhih-zkz
meet.google.com/tvg-yhih-zkz?authuser=0
Ritu Sibbal is presenting
VISHAL VERMA and 42 more
4:05 PM
Pointer Conversions
• One type of pointer can be converted into another type of pointer. There are two general categories of conversion: those that involve **void** * pointers, and those that don't.
• Each is examined one by one
You
AKHIL DUBEY
SHIRISHTI JAIN
Ritu Sibbal
ANSHUL JANGRA
bcxzgslmls
Raise hand
Turn on captions
Ritu Sibbal is presenting

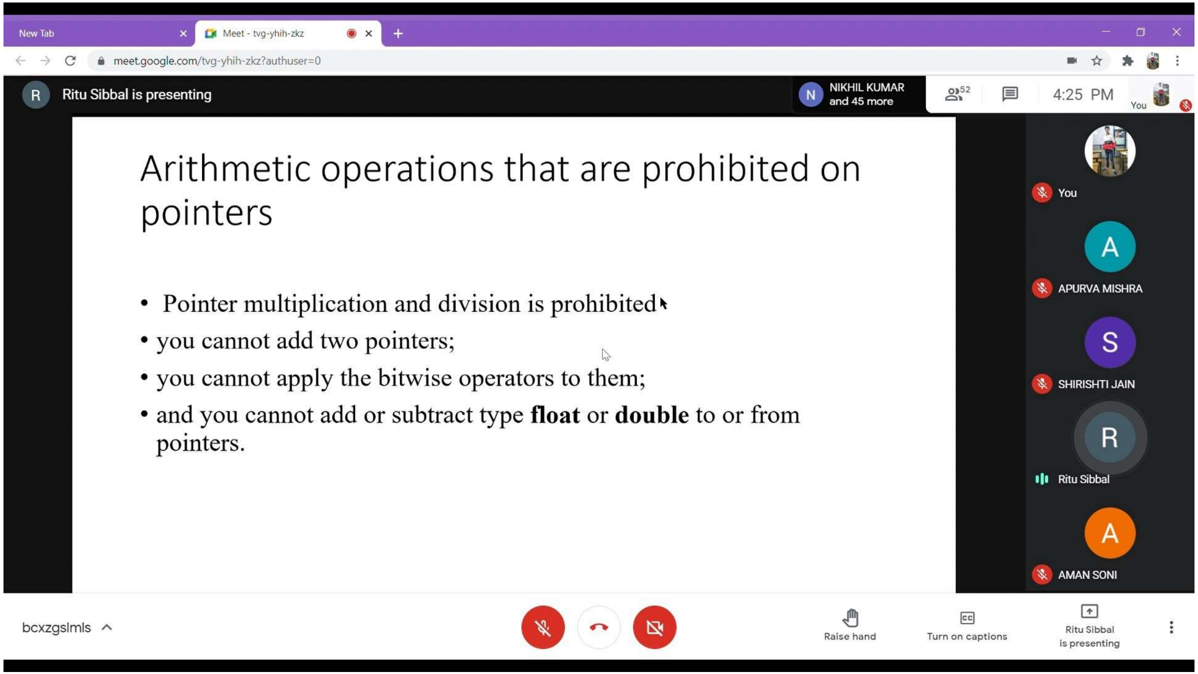

New Tab
Meet - tvg-yhih-zkz
meet.google.com/tvg-yhih-zkz?authuser=0
Ritu Sibbal is presenting
ANUJ
and 42 more
4:05 PM
You
Void* pointer
• A void * pointer is called a generic pointer.
• The void * pointer is used to specify a pointer whose base type is unknown.
• The void * type allows a function to specify a parameter that is capable of receiving any type of pointer argument without reporting a type mismatch.
• It is also used to refer to raw memory (such as that returned by the malloc( ) function) when the semantics of that memory are not known.
• No explicit cast is required to convert to or from a void * pointer.
You
AKHIL DUBEY
S
SHIRISHTI JAIN
R
Ritu Sibbal
ROHIT SHARMA
bcxzgslmls
Raise hand
Turn on captions
Ritu Sibbal
is presenting

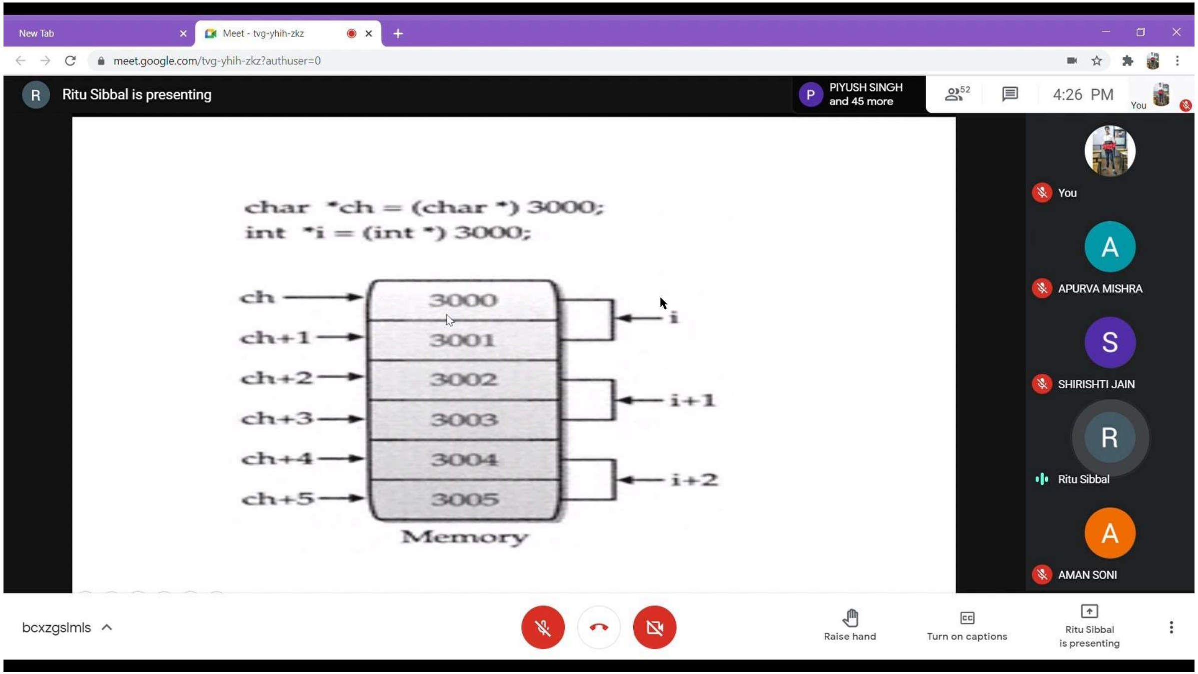

New Tab
Meet - tvg-yhih-zkz
meet.google.com/tvg-yhih-zkz?authuser=0
Ritu Sibbal is presenting
ANANT
and 45 more
4:09 PM
You
#include <stdio.h>
int main(void)
{
double x = 100.1, y;
int *p;
/* The next statement causes p (which is an
integer pointer) to point to a double. */
p = (int *) &x;
/* The next statement does not operate as expected. */
y = *p; /* attempt to assign y the value x through p */
/* The following statement won't output 100.1. */
printf(''The (incorrect) value of x is: %f", y);
return 0;
}
You
AKHIL DUBEY
ABHISHEK KUMAR
R
Ritu Sibbal
A
AMAN SONI
bcxzgslmls
Raise hand
Turn on captions
Ritu Sibbal
is presenting

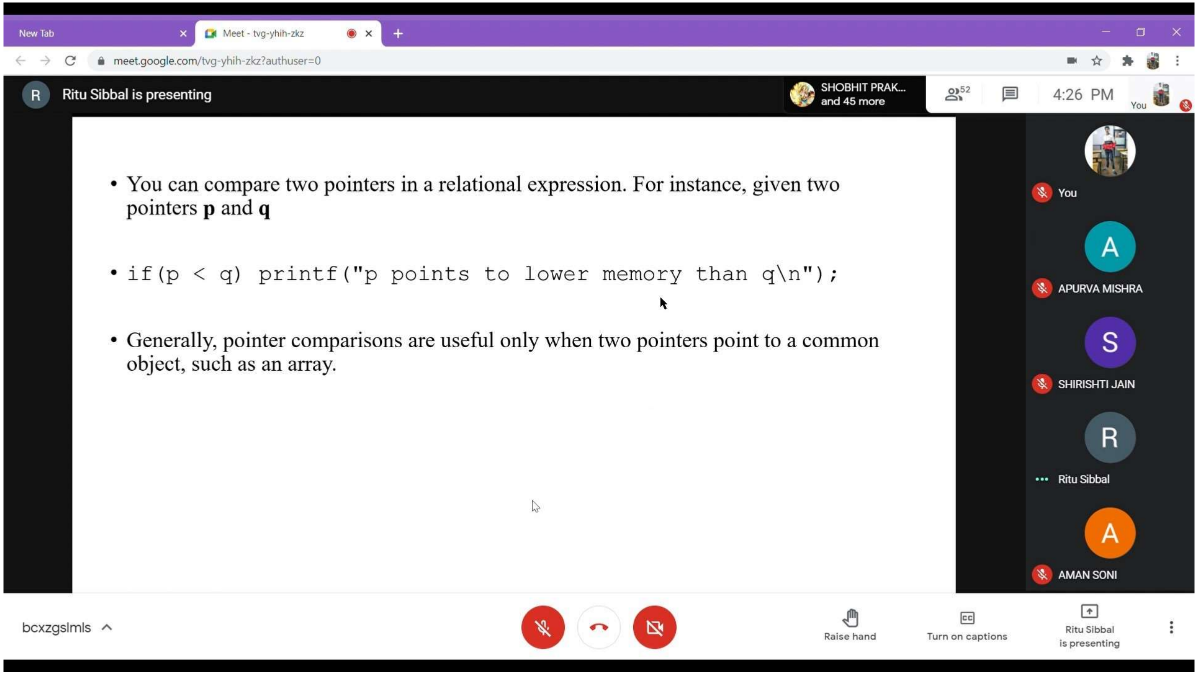

New Tab
Meet - tvg-yhih-zkz
meet.google.com/tvg-yhih-zkz?authuser=0
Ritu Sibbal is presenting
ARUSHEE VERMA
and 45 more
52
4:20 PM
You
Pointer Arithmetic
• There are only two arithmetic operations that you can use on pointers: **addition and subtraction**.
• Let **p1** be an integer pointer with a current value of 2000. Also, assume **ints** are 2 bytes long. After the expression
• p1++;
• **p1** contains 2002, not 2001.
• The reason for this is that each time **p1** is incremented, it will point to the next integer.
• The same is true of decrements. For example, assuming that **p1** has the value 2000, the expression
• p1--;
• causes **p1** to have the value 1998.
You
APURVA MISHRA
SHIRISHTI JAIN
Ritu Sibbal
AMAN SONI
bcxzgslmls
Raise hand
Turn on captions
Ritu Sibbal
is presenting

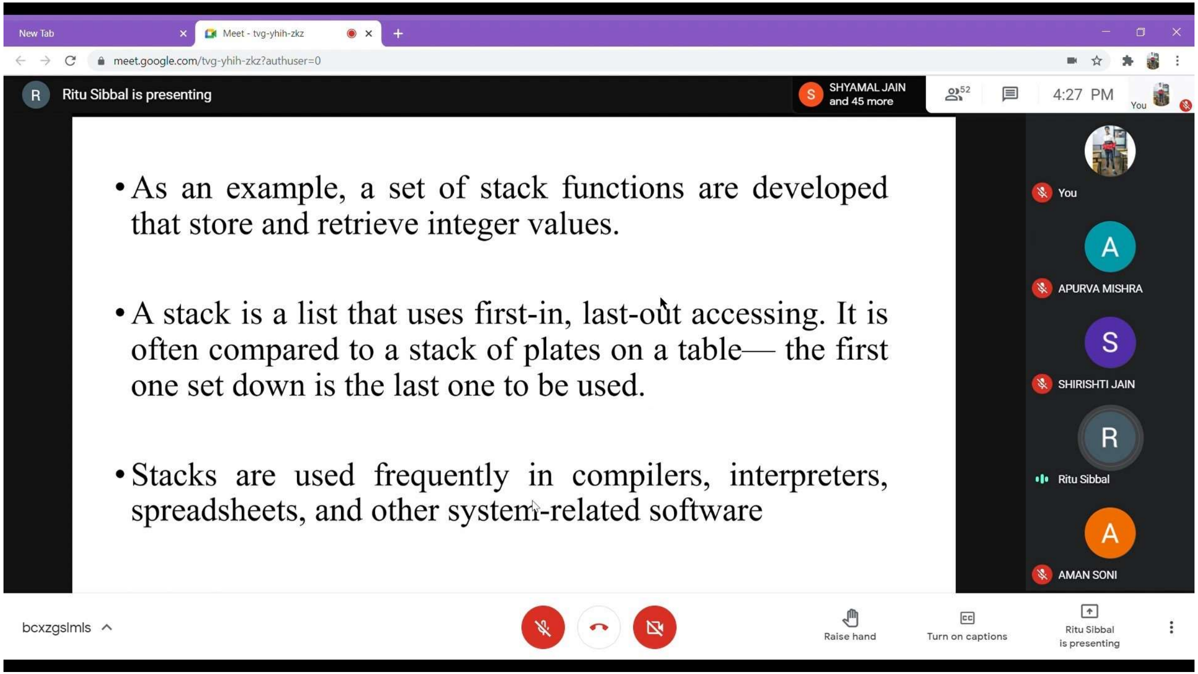

New Tab
Meet - tvg-yhih-zkz
meet.google.com/tvg-yhih-zkz?authuser=0
Ritu Sibbal is presenting
ROHIT SHARMA
and 45 more
52
4:22 PM
You
Rules for pointer arithmetic.
Each time a pointer is incremented, it points to the memory location of the next element of its base type.
Each time it is decremented, it points to the location of the previous element.
you may add or subtract integers to or from pointers.
• The expression
• p1 = p1 + 12;
• makes p1 point to the 12th element of p1's type beyond the one it currently points to.
You
APURVA MISHRA
SHIRISHTI JAIN
Ritu Sibbal
AMAN SONI
bcxzgslmls
Raise hand
Turn on captions
Ritu Sibbal
is presenting

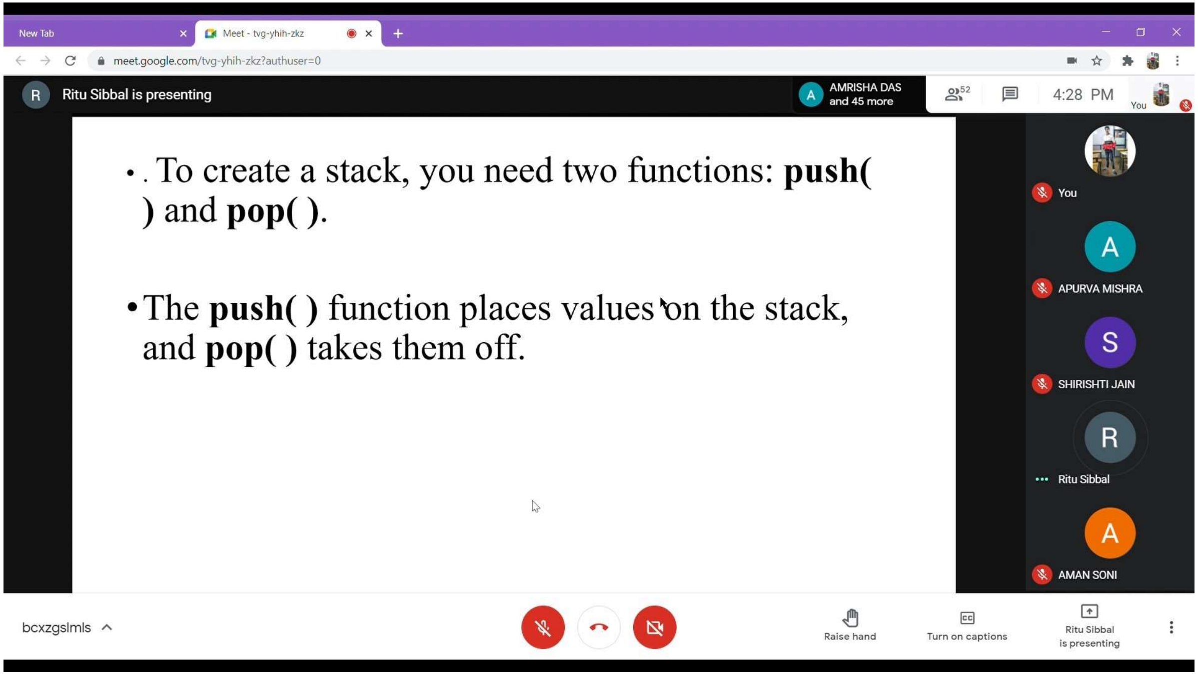

New Tab
Meet - tvg-yhih-zkz
meet.google.com/tvg-yhih-zkz?authuser=0
Ritu Sibbal is presenting
DHANANJAY G...
and 45 more
4:25 PM
• Besides addition and subtraction of a pointer and an integer, only one other arithmetic operation is allowed:
• You can subtract one pointer from another in order to find the number of objects of their base type that separate the two.
You
APURVA MISHRA
SHIRISHTI JAIN
Ritu Sibbal
AMAN SONI
bcxzgslmls
Raise hand
Turn on captions
Ritu Sibbal
is presenting


New Tab
Meet - tvg-yhih-zkz
meet.google.com/tvg-yhih-zkz?authuser=0
Ritu Sibbal is presenting
NIKHIL KUMAR
and 45 more
52
4:25 PM
You
Arithmetic operations that are prohibited on pointers
• Pointer multiplication and division is prohibited
• you cannot add two pointers;
• you cannot apply the bitwise operators to them;
• and you cannot add or subtract type **float** or **double** to or from pointers.
You
APURVA MISHRA
SHIRISHTI JAIN
Ritu Sibbal
AMAN SONI
bcxzgslmls
Raise hand
Turn on captions
Ritu Sibbal
is presenting

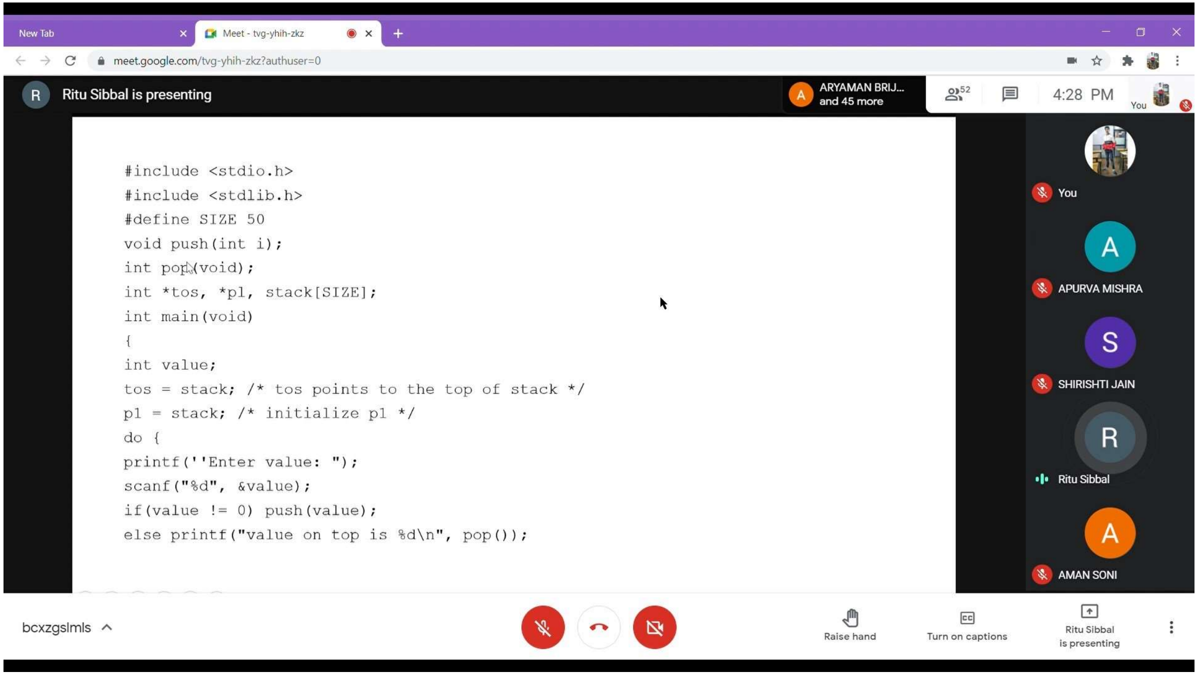

New Tab
Meet - tvg-yhih-zkz
meet.google.com/tvg-yhih-zkz?authuser=0
Ritu Sibbal is presenting
PIYUSH SINGH
and 45 more
4:26 PM
You
char *ch = (char *) 3000;
int *i = (int *) 3000;
ch
ch+1
ch+2
ch+3
ch+4
ch+5
3000
3001
3002
3003
3004
3005
i
i+1
i+2
Memory
You
APURVA MISHRA
SHIRISHTI JAIN
Ritu Sibbal
AMAN SONI
bcxzgslmls
Raise hand
Turn on captions
Ritu Sibbal
is presenting

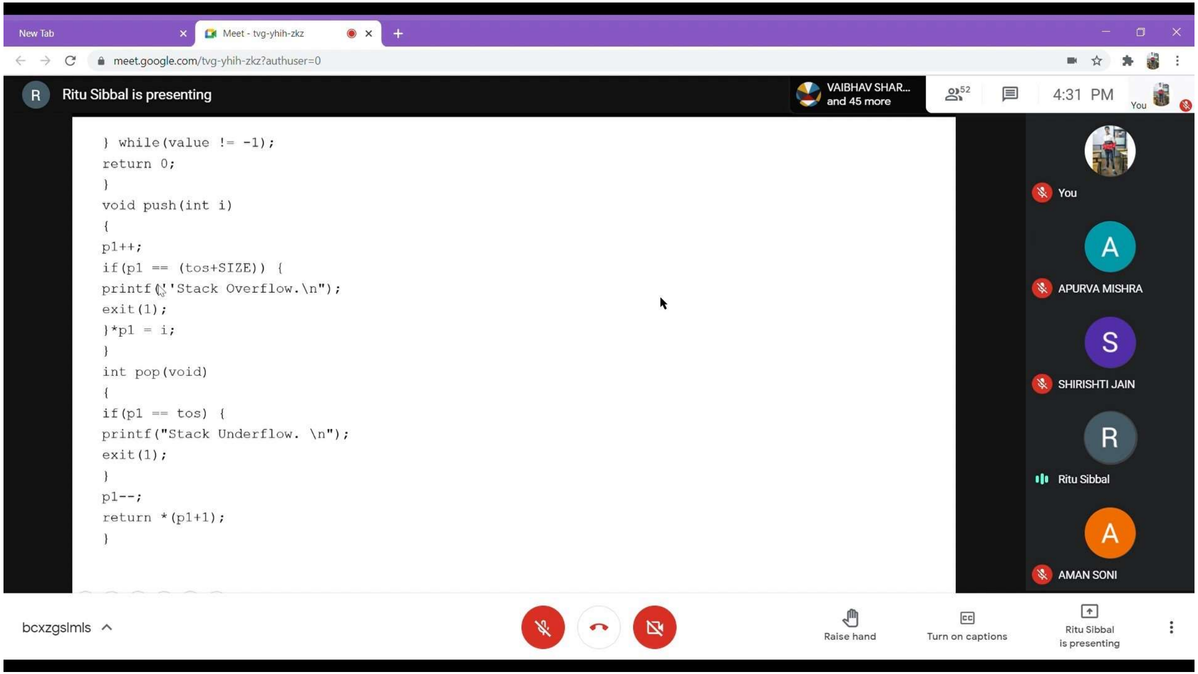

New Tab
Meet - tvg-yhih-zkz
meet.google.com/tvg-yhih-zkz?authuser=0
Ritu Sibbal is presenting
SHOBHIT PRAK...
and 45 more
52
4:26 PM
You
• You can compare two pointers in a relational expression. For instance, given two pointers **p** and **q**
• if(p < q) printf("p points to lower memory than q\n");
• Generally, pointer comparisons are useful only when two pointers point to a common object, such as an array.
You
APURVA MISHRA
SHIRISHTI JAIN
Ritu Sibbal
AMAN SONI
bcxzgslmls
Raise hand
Turn on captions
Ritu Sibbal
is presenting


New Tab
Meet - tvg-yhih-zkz
meet.google.com/tvg-yhih-zkz?authuser=0
Ritu Sibbal is presenting
SHYAMAL JAIN
and 45 more
4:27 PM
You
• As an example, a set of stack functions are developed that store and retrieve integer values.
• A stack is a list that uses first-in, last-out accessing. It is often compared to a stack of plates on a table— the first one set down is the last one to be used.
• Stacks are used frequently in compilers, interpreters, spreadsheets, and other system-related software
You
APURVA MISHRA
SHIRISHTI JAIN
Ritu Sibbal
AMAN SONI
bcxzgslmls
Raise hand
Turn on captions
Ritu Sibbal
is presenting



New Tab
Meet - tvg-yhih-zkz
meet.google.com/tvg-yhih-zkz?authuser=0
Ritu Sibbal is presenting
AMRISHA DAS and 45 more
4:28 PM
• . To create a stack, you need two functions: **push( )** and **pop( )**.
• The **push( )** function places values on the stack, and **pop( )** takes them off.
You
APURVA MISHRA
SHIRISHTI JAIN
Ritu Sibbal
AMAN SONI
bcxzgslmls
Raise hand
Turn on captions
Ritu Sibbal is presenting


New Tab
Meet - tvg-yhih-zkz
meet.google.com/tvg-yhih-zkz?authuser=0
Ritu Sibbal is presenting
ARYAMAN BRIJ...
and 45 more
52
4:28 PM
You
#include <stdio.h>
#include <stdlib.h>
#define SIZE 50
void push(int i);
int pop(void);
int *tos, *p1, stack[SIZE];
int main(void)
{
int value;
tos = stack; /* tos points to the top of stack */
p1 = stack; /* initialize p1 */
do {
printf(''Enter value: ");
scanf("%d", &value);
if(value != 0) push(value);
else printf("value on top is %d\n", pop());
You
APURVA MISHRA
SHIRISHTI JAIN
Ritu Sibbal
AMAN SONI
bcxzgslmls
Raise hand
Turn on captions
Ritu Sibbal
is presenting


New Tab
Meet - tvg-yhih-zkz
meet.google.com/tvg-yhih-zkz?authuser=0
Ritu Sibbal is presenting
VAIBHAV SHAR...
and 45 more
52
4:31 PM
You
} while(value != -1);
return 0;
}
void push(int i)
{
p1++;
if(p1 == (tos+SIZE)) {
printf("Stack Overflow.\n");
exit(1);
}*p1 = i;
}
int pop(void)
{
if(p1 == tos) {
printf("Stack Underflow. \n");
exit(1);
}
p1--;
return *(p1+1);
}
You
APURVA MISHRA
SHIRISHTI JAIN
Ritu Sibbal
AMAN SONI
bcxzgslmls
Raise hand
Turn on captions
Ritu Sibbal
is presenting