FCCS002 COE VI Section 3 | Meet - fyt-txme-esa

meet.google.com/fyt-txme-esa?authuser=0

Ritu Sibbal is presenting

SHUBHANGI KA... and 49 more

56

4:05 PM

You

Compilers vs. Interpreters

- There are two general methods by which a program can be executed.
- It can be *compiled*, or it can be *interpreted*.
- some languages are designed more for one form of execution than the other.
- For example, Java was designed to be interpreted, and C was designed to
- be compiled.

DHRUV KUMAR

ARYAN CHAUHAN

Ritu Sibbal

MEGHNA

FCCS002 COE VI Section 3 | Meet - fyt-txme-esa

meet.google.com/fyt-txme-esa?authuser=0

Ritu Sibbal is presenting

ADARSH KUMAR and 51 more

58

4:06 PM

You

- In its simplest form, an interpreter reads the source code of the program one line at a time, performing the specific instructions contained in that line.

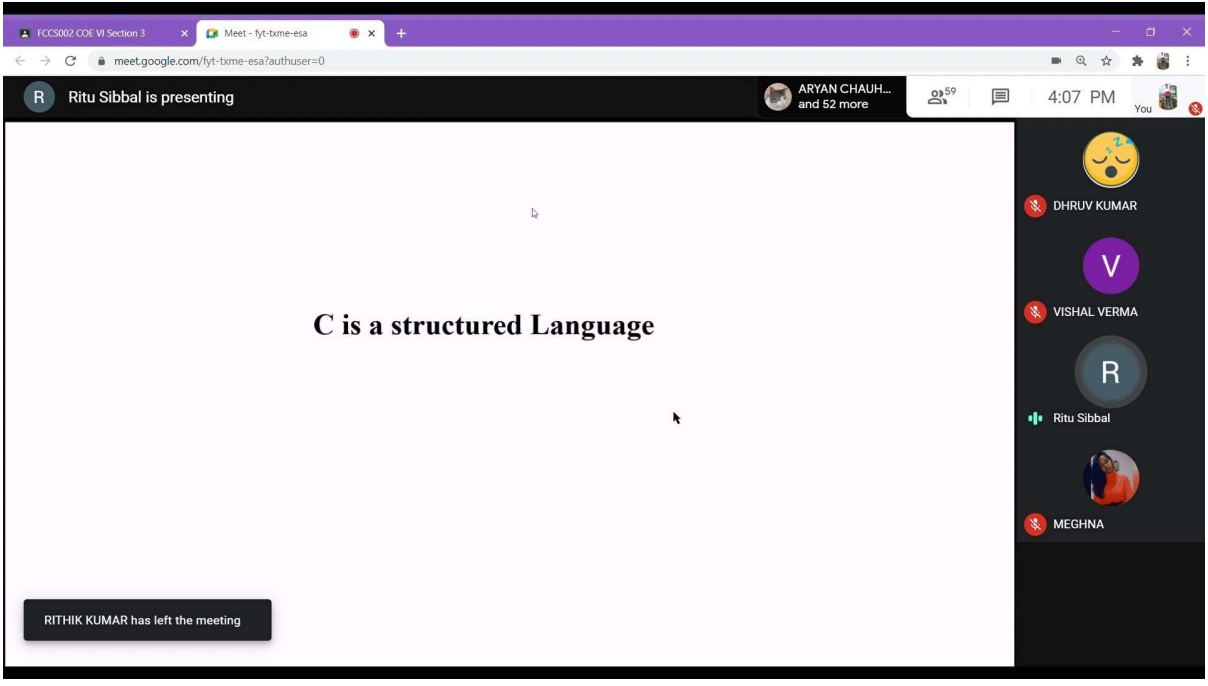
- A compiler reads the entire program and converts it into *object code*, which is a translation of the program's source code into a form that the computer can execute directly.
- Object code is also referred to as *binary code* or *machine code*.
- Once the program is compiled, a line of source code is no longer meaningful in the execution of your program.

DHRUV KUMAR

ARYAN CHAUHAN

Ritu Sibbal

MEGHNA


FCCS002 COE VI Section 3
Meet - fyt-txme-esa
meet.google.com/fyt-txme-esa?authuser=0
Ritu Sibbal is presenting
ARYAN CHAUH...
and 52 more
59
4:07 PM
You
C is a structured Language
DHRUV KUMAR
VISHAL VERMA
Ritu Sibbal
MEGHNA
RITHIK KUMAR has left the meeting

FCCS002 COE VI Section 3 | Meet - fyt-txme-esa

meet.google.com/fyt-txme-esa?authuser=0

Ritu Sibbal is presenting

KATARI PRAJEET YA... and 53 more

60

4:08 PM

You

- The distinguishing feature of a structured language is *compartmentalization* of code and data.

- This is the ability of a language to section off and hide from the rest of the program all information and instructions necessary to perform a specific task.

- One way to achieve compartmentalization is by using subroutines that employ local (temporary) variables.

- By using local variables, one can write subroutines so that the events that occur within them cause no side effects in other parts of the program.

DHRUV KUMAR

VISHAL VERMA

Ritu Sibbal

MEGHNA

FCCS002 COE VI Section 3 | Meet - fyt-txme-esa

meet.google.com/fyt-txme-esa?authuser=0

Ritu Sibbal is presenting

SHUBHAM AGGAR... and 56 more

63

4:10 PM

You

- This capability makes it very easy for C programs to share sections of code.
- Compartmentalized functions, need to know only what a function does, not how it does it.
- Excessive use of global variables (variables known throughout the entire program) may allow bugs to creep into a program by allowing unwanted side effects. It should be reduced

NIKHIL KUMAR has left the meeting

DHRUV KUMAR

VISHAL VERMA

Ritu Sibbal

NIKHIL KUMAR

FCCS002 COE VI Section 3 | Meet - fyt-txme-esa

meet.google.com/fyt-txme-esa?authuser=0

Ritu Sibbal is presenting

ANANT and 57 more

64

4:11 PM

You

- **C's main structural component is the function— C's stand-alone subroutine.**
- In C, functions are the building blocks in which all program activity occurs.
- They allow you to define and code individually the separate tasks in a program, thus allowing your programs to be modular.
- After you have created a function, you can rely on it to work properly in various situations without creating side effects in other parts of the program.
- **Being able to create stand-alone functions is extremely important in larger projects where one programmer's code must not accidentally affect another's.**

DHRUV KUMAR

VISHAL VERMA

Ritu Sibbal

MEGHNA

FCCS002 COE VI Section 3 | Meet - fyt-txme-esa

meet.google.com/fyt-txme-esa?authuser=0

Ritu Sibbal is presenting

MANAN SURI and 58 more

65

4:13 PM

You

- **Another way to structure and compartmentalize code in C is through the use of blocks of code.**
- A *code block* is a logically connected group of program statements that is treated as a unit.
- In C, you create a code block by placing a sequence of statements between opening and closing curly braces.
- In this example,

```
if (x < 10) {
printf(''Too low, try again.\n");
scanf("%d", &x);
}
```

- the two statements after the **if** and between the curly braces are both executed if **x** is less than 10.
- These two statements together with the braces represent a code block. They are a logical unit: One of the statements cannot execute without the other executing also

DHRUV KUMAR

VISHAL VERMA

Ritu Sibbal

MEGHNA

FCCS002 COE VI Section 3 | Meet - fyt-txme-esa

meet.google.com/fyt-txme-esa?authuser=0

Ritu Sibbal is presenting

SHUBHAM AGGAR... and 58 more

65

4:14 PM

You

- . Code blocks allow many algorithms to be implemented with clarity, elegance, and efficiency.
- They help the programmer better conceptualize the true nature of the algorithm being implemented.

DHRUV KUMAR

VISHAL VERMA

Ritu Sibbal

ROHIT SHARMA

FCCS002 COE VI Section 3 | Meet - fyt-txme-esa

meet.google.com/fyt-txme-esa?authuser=0

Ritu Sibbal is presenting

AASTIK CHAUDH... and 59 more

66

4:15 PM

You

```
• Global declarations
• int main(parameter list)
        {
                statement sequence
        }
return-type f1(parameter list)
        {
        statement sequence
        }
return-type f2(parameter list)
        {
        statement sequence
        }...
return-type fN(parameter list)
        {
        statement sequence
        }
```

DHRUV KUMAR

VISHAL VERMA

Ritu Sibbal

ROHIT SHARMA

FCCS002 COE VI Section 3 | Meet - fyt-txme-esa

meet.google.com/fyt-txme-esa?authuser=0

Ritu Sibbal is presenting

MEGHNA and 61 more

68

4:18 PM

You

- All C programs consist of one or more functions.
- As a general rule, the only function that must be present is called **main( )**, which is the first function called when program execution begins.
- **main( )** contains what is, in essence, an outline of what the program does.
- The outline is composed of function calls.
- Although **main( )** is not a keyword, treat it as if it were. For example, don't try to use **main** as the name of a variable because you will probably confuse the compiler.

DHRUV KUMAR

A

AMRISHA DAS

R

Ritu Sibbal

S

SHUBHAM SHARMA

FCCS002 COE VI Section 3 | Meet - fyt-txme-esa

meet.google.com/fyt-txme-esa?authuser=0

Ritu Sibbal is presenting

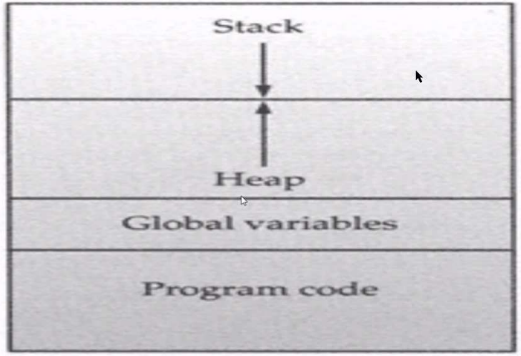
SHIRISHTI JAIN and 62 more

69

4:19 PM

You

Conceptualized memory map of a C program



DHRUV KUMAR

A

AMRISHA DAS

R

Ritu Sibbal

S

SHUBHAM SHARMA

FCCS002 COE VI Section 3 | Meet - fyt-txme-esa

meet.google.com/fyt-txme-esa?authuser=0

Ritu Sibbal is presenting

NISHCHAY HASI... and 63 more

70

4:23 PM

You

- A compiled C program creates and uses four logically distinct regions of memory.
- The first region is the memory that actually holds the program's executable code.
- The next region is memory where global variables are stored.
- The remaining two regions are the stack and the heap. The *stack* is used for a great many things while your program executes.
  - It holds the return addresses of function calls, arguments to functions, and local variables.
  - It will also save the current state of the CPU.

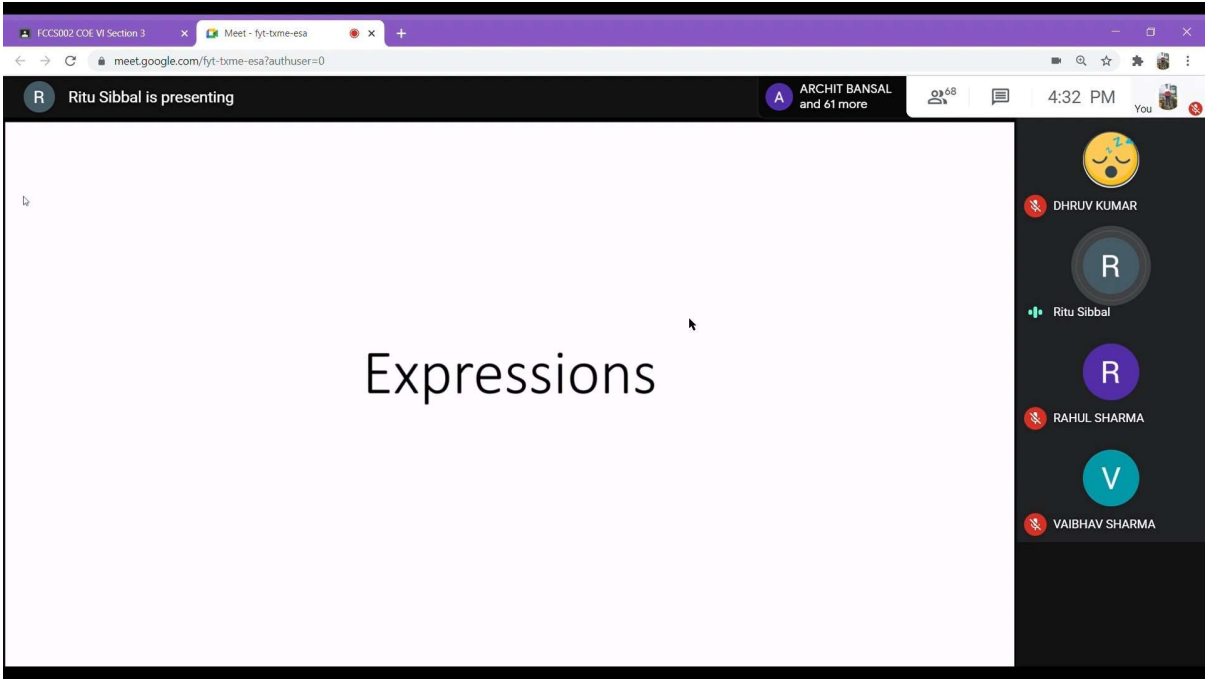
  The *heap* is a region of free memory that your program can use via C's dynamic memory allocation functions.
- The exact physical layout of each of the four regions of memory differs among CPU types and C implementations,
- The diagram shows conceptually how your C programs appear in memory.

DHRUV KUMAR

AMRISHA DAS

Ritu Sibbal

SHUBHAM SHARMA


FCCS002 COE VI Section 3
Meet - fyt-txme-esa
meet.google.com/fyt-txme-esa?authuser=0
Ritu Sibbal is presenting
ARCHIT BANSAL and 61 more
68
4:32 PM
You
Expressions
DHRUV KUMAR
Ritu Sibbal
RAHUL SHARMA
VAIBHAV SHARMA

FCCS002 COE VI Section 3 | Meet - fyt-txme-esa

meet.google.com/fyt-txme-esa?authuser=0

Ritu Sibbal is presenting

DAKSH CHETIW... and 61 more

68

4:32 PM

You

- Expressions are the most fundamental elements of the 'C' language
- Expressions are formed from the atomic elements: data and operators.
- Data may be represented by variables, constants, or values returned by functions.
- C supports several different types of data. It also provides a wide variety of operators

DHRUV KUMAR

Ritu Sibbal

RAHUL SHARMA

VAIBHAV SHARMA

FCCS002 COE VI Section 3 | Meet - fyt-txme-esa

meet.google.com/fyt-txme-esa?authuser=0

Ritu Sibbal is presenting

KESHAV SINGH... and 61 more

68

4:33 PM

You

Basic Data Types

- A *data type* defines a set of values that a variable can store along with a set of operations that can be performed on that variable. .
- **There are five foundational data types**
- character
- integer
- floating-point
- double floating-point
- and valueless
- These are declared using **char**, **int**, **float**, **double**, and **void**, respectively.

DHRUV KUMAR

Ritu Sibbal

RAHUL SHARMA

VAIBHAV SHARMA

FCCS002 COE VI Section 3 | Meet - fyt-txme-esa

meet.google.com/fyt-txme-esa?authuser=0

Ritu Sibbal is presenting

NIKHIL KUMAR and 61 more

68

4:33 PM

You

To the five basic data types there are three more data types:

- ***_Bool***,
- ***_Complex***, *and*
- ***_Imaginary***.

DHRUV KUMAR

Ritu Sibbal

RAHUL SHARMA

VAIBHAV SHARMA

FCCS002 COE VI Section 3 | Meet - fyt-txme-esa

meet.google.com/fyt-txme-esa?authuser=0

Ritu Sibbal is presenting

NITASH BISWAS and 61 more

68

4:34 PM

You

- These types form the basis for several other types.
- The size and range of these data types may vary among processor types and compilers.
- However, in all cases an object of type **char** is 1 byte.
- The size of an **int** is usually the same as the word length of the execution environment of the program. For most 16-bit environments, such as DOS or Windows 3.1, an **int** is 16 bits.
- For most 32-bit environments, such as Windows 95/98/NT/2000, an **int** is 32 bits.

DHRUV KUMAR

Ritu Sibbal

RAHUL SHARMA

VAIBHAV SHARMA

FCCS002 COE VI Section 3 | Meet - fyt-txme-esa

meet.google.com/fyt-txme-esa?authuser=0

Ritu Sibbal is presenting

VISHAL MALIK and 61 more

68

4:35 PM

You

- The range of **float** and **double** will depend upon the method used to represent the floating-point numbers.
- Standard C specifies that the minimum range for a floating-point value is 1E–37 to 1E+37.

- Variables of type **char** are generally used to hold values defined by the ASCII character set.

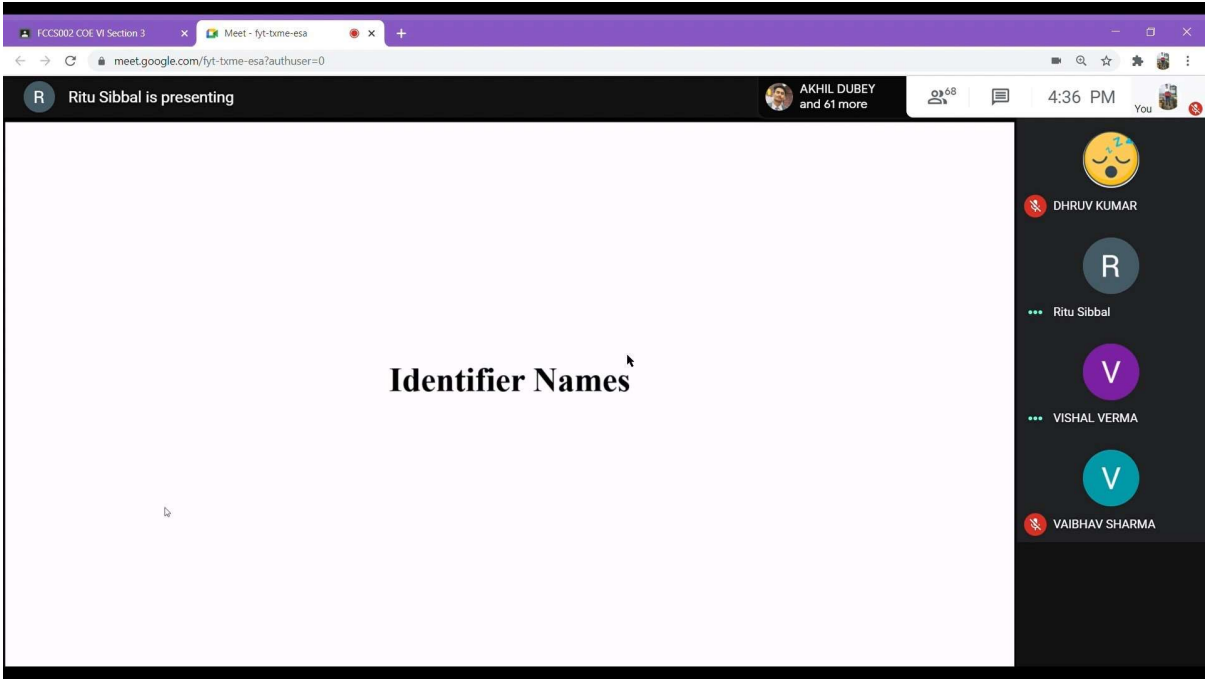
- The type **void** either explicitly declares a function as returning no value or creates generic pointers

DHRUV KUMAR

Ritu Sibbal

RAHUL SHARMA

VAIBHAV SHARMA


FCCS002 COE VI Section 3
Meet - fyt-txme-esa
meet.google.com/fyt-txme-esa?authuser=0
Ritu Sibbal is presenting
AKHIL DUBEY
and 61 more
68
4:36 PM
You
Identifier Names
DHRUV KUMAR
Ritu Sibbal
VISHAL VERMA
VAIBHAV SHARMA

FCCS002 COE VI Section 3 | Meet - fyt-txme-esa

meet.google.com/fyt-txme-esa?authuser=0

Ritu Sibbal is presenting

ANSHUL JANGRA and 61 more

68

4:36 PM

You

- In C, the names of variables, functions, labels, and various other user-defined items are called *identifiers*.
- The length of these identifiers can vary from one to several characters. The first character must be a letter or an underscore, and subsequent characters must be either letters, digits, or underscores.

- Here are some correct and incorrect identifier names:

Correct	Incorrect
count	1count
Test23	hi!there
high_balance	high . . . Balance

- In an identifier, upper- and lowercase are treated as distinct. Hence, **count** , **Count**, and **COUNT** are three separate identifiers.

DHRUV KUMAR

Ritu Sibbal

VISHAL VERMA

VAIBHAV SHARMA

FCCS002 COE VI Section 3 | Meet - fyt-txme-esa

meet.google.com/fyt-txme-esa?authuser=0

Ritu Sibbal is presenting

MANAN SURI and 61 more

68

4:38 PM

You

Variable

- A*variable* is a named location in memory that is used to hold a value that can be modified by the program. All variables must be declared before they can be used. The general form of a declaration is

  `*type variable_list*;

- Here, *type* must be a valid data type , and *variable_list* may consist of one or more identifier names separated by commas. Here are some declarations:

```
int i, j, l;
Char  si;
double balance, profit, loss;
```

DHRUV KUMAR

Ritu Sibbal

ISHAAN KUMAR

RITIK YADAV

FCCS002 COE VI Section 3 | Meet - fyt-txme-esa

meet.google.com/fyt-txme-esa?authuser=0

Ritu Sibbal is presenting

AMAN SONI and 62 more

4:41 PM

Where Variables are Declared

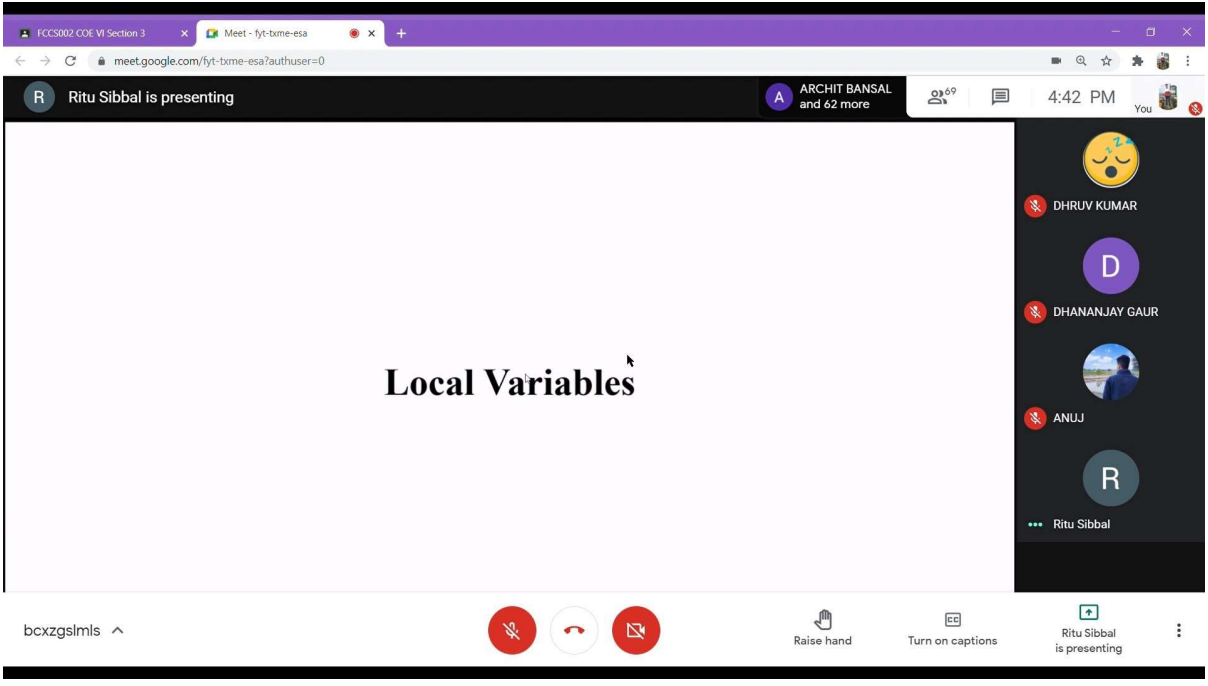
- Variables can be declared in three places: inside functions, in the definition of function parameters, and outside of all functions.
- **These positions correspond to local variables, formal parameters, and global variables, respectively.**

DHRUV KUMAR

Ritu Sibbal

MEGHNA

ANUJ


FCCS002 COE VI Section 3
Meet - fyt-txme-esa
meet.google.com/fyt-txme-esa?authuser=0
Ritu Sibbal is presenting
ARCHIT BANSAL and 62 more
69
4:42 PM
You
Local Variables
DHRUV KUMAR
DHANANJAY GAUR
ANUJ
Ritu Sibbal
bcxzgslmls
Raise hand
Turn on captions
Ritu Sibbal is presenting

FCCS002 COE VI Section 3 | Meet - fyt-txme-esa

meet.google.com/fyt-txme-esa?authuser=0

Ritu Sibbal is presenting

MAJITHIYA RISH... and 62 more

69

4:44 PM

You

Local Variables

- Variables that are declared inside a function are called *local variables*. Or *automatic variables* .
- Local variables can be used only by statements that are inside the block in which the variables are declared.
- Local variables are not known outside their own code block.
- Local variables exist only while the block of code in which they are declared is executing.
- That is, a local variable is created upon entry into its block and destroyed upon exit.
- A variable declared within one code block has no bearing on or relationship to

DHRUV KUMAR

P

PRATHAM JAIN

ANUJ

R

Ritu Sibbal

bcxzgslmls

Raise hand

Turn on captions

Ritu Sibbal is presenting

FCCS002 COE VI Section 3 | Meet - fyt-txme-esa

meet.google.com/fyt-txme-esa?authuser=0

Ritu Sibbal is presenting

MEGHNA and 62 more

69

4:44 PM

You

- The most common code block in which local variables are declared is the function. For example, consider the following two functions

```
void func1(void)
{
        int x;
        x = 10;
}
void func2(void)
{
        int x;
        x = -199;
```

DHRUV KUMAR

PRATHAM JAIN

ANUJ

Ritu Sibbal

bcxzgslmls

Raise hand

Turn on captions

Ritu Sibbal is presenting

FCCS002 COE VI Section 3 | Meet - fyt-txme-esa

meet.google.com/fyt-txme-esa?authuser=0

Ritu Sibbal is presenting

PARANJAY SIN... and 62 more

69

4:44 PM

You

- The integer variable **x** is declared twice, once in **func1( )** and once in **func2( )**.
- The **x** in **func1( )** has no bearing on or relationship to the **x** in **func2( )**.
- As explained, this is because each **x** is known only to the code within the block in which it is declared

DHRUV KUMAR

PRATHAM JAIN

ANUJ

Ritu Sibbal

bcxzgslmls

Raise hand

Turn on captions

Ritu Sibbal is presenting

FCCS002 COE VI Section 3 | Meet - fyt-txme-esa

meet.google.com/fyt-txme-esa?authuser=0

Ritu Sibbal is presenting

RAHUL and 62 more

4:44 PM

```
void f(void)
{
        int t;
        scanf("%d%*c", &t);
        if(t==1) {
                char s[80]; /* this is created only upon entry into this block */
                printf(''Enter name:");
                gets(s);
                /* do something . . . */
                }
/* s not known here */
```

DHRUV KUMAR

PRATHAM JAIN

ANUJ

Ritu Sibbal

bcxzgslmls

Raise hand

Turn on captions

Ritu Sibbal is presenting

FCCS002 COE VI Section 3 | Meet - fyt-txme-esa

meet.google.com/fyt-txme-esa?authuser=0

Ritu Sibbal is presenting

SHOBHIT PRAK... and 62 more

69

4:45 PM

You

- Here, the local variable **s** is created upon entry into the **if** code block and destroyed upon exit.
- **s** is known only within the **if** block and cannot be referenced elsewhere— even in other parts of the function that contains it.
- Declaring variables within the block of code that uses them helps prevent unwanted side effects.
- Since the variable does not exist outside the block in which it is declared, it cannot be accidentally altered by other code.

DHRUV KUMAR

PRATHAM JAIN

ANUJ

Ritu Sibbal

bcxzgslmls

Raise hand

Turn on captions

Ritu Sibbal is presenting

FCCS002 COE VI Section 3 | Meet - fyt-txme-esa

meet.google.com/fyt-txme-esa?authuser=0

Ritu Sibbal is presenting

ADARSH KUMAR and 62 more

69

4:46 PM

You

- When a variable declared within an inner block has the same name as a variable declared by an enclosing block, the variable in the inner block *hides* the variable in the outer block. Consider the following:

```
#include <stdio.h>
int main(void)
{
        int x;
        x = 10;
        if(x == 10) {
                int x; /* this x hides the outer x */
                x = 99;
                printf("Inner x: %d\n", x);
}
printf("Outer x: %d\n", x);
return 0;
}
```

DHRUV KUMAR

PRATHAM JAIN

RITIK YADAV

Ritu Sibbal

bcxzgslmls

Raise hand

Turn on captions

Ritu Sibbal is presenting

FCCS002 COE VI Section 3 | Meet - fyt-txme-esa

meet.google.com/fyt-txme-esa?authuser=0

Ritu Sibbal is presenting

AMAN SONI and 62 more

4:47 PM

- In this example, the **x** that is declared within the **if** block hides the outer **x**.
- Thus, the inner **x** and the outer **x** are two separate and distinct objects. Once that block ends, the outer **x** once again becomes visible
- The Program displays the output:

```
• Inner x: 99
• Outer x: 10
```

DHRUV KUMAR

PRATHAM JAIN

RITIK YADAV

Ritu Sibbal

bcxzgslmls

Raise hand

Turn on captions

Ritu Sibbal is presenting

- Because local variables are created and destroyed with each entry and exit from the block in which they are declared, their content is lost once the block is left.
- Therefore When a function is called, its local variables are created, and upon its return they are destroyed.
- Local variables cannot retain their values between calls. (However, you can direct the compiler to retain their values by using the **static** modifier.)
- Unless otherwise specified, local variables are stored on the stack.
- The fact that the stack is a dynamic and changing region of memory explains why local variables cannot, in general, hold their values between function calls.

FCCS002-COE VI Section 3 | Meet - fyt-txme-esa

meet.google.com/fyt-txme-esa?authuser=0

Ritu Sibbal is presenting

KESHAV SINGH... and 61 more

68

4:54 PM

You

Lecture2 'C' Expressions [Autosaved].pptx - PowerPoint

Ritu Sibal

File | Home | Insert | Design | Transitions | Animations | Slide Show | Review | View | Help | Acrobat | Tell me what you want to do | Share

From Beginning | From Current Slide | Present Online | Custom Slide Show | Start Slide Show

Set Up Slide Show | Hide Slide | Rehearse Timings | Record Slide Show | Keep Slides Updated | Play Narrations | Use Timings | Show Media Controls | Set Up

Monitor: Automatic | Use Presenter View | Monitors

- You can initialize a local variable to some known value. This value will be assigned to the variable each time the block of code in which it is declared is entered. For example, the following program prints the number 10 ten times:
- #include <stdio.h>

```
void f(void);
int main(void)
{
int i;
for(i=0; i<10; i++) f();
return 0;
}
void f(void)
{
int j = 10;
printf("%d ", j);
j++; /* this line has no lasting effect */
}
```

Slide 20 of 25 | English (United States) | Notes | Comments | 66%

DHRUV KUMAR

ROHIT SHARMA

RITIK YADAV

R

Ritu Sibbal