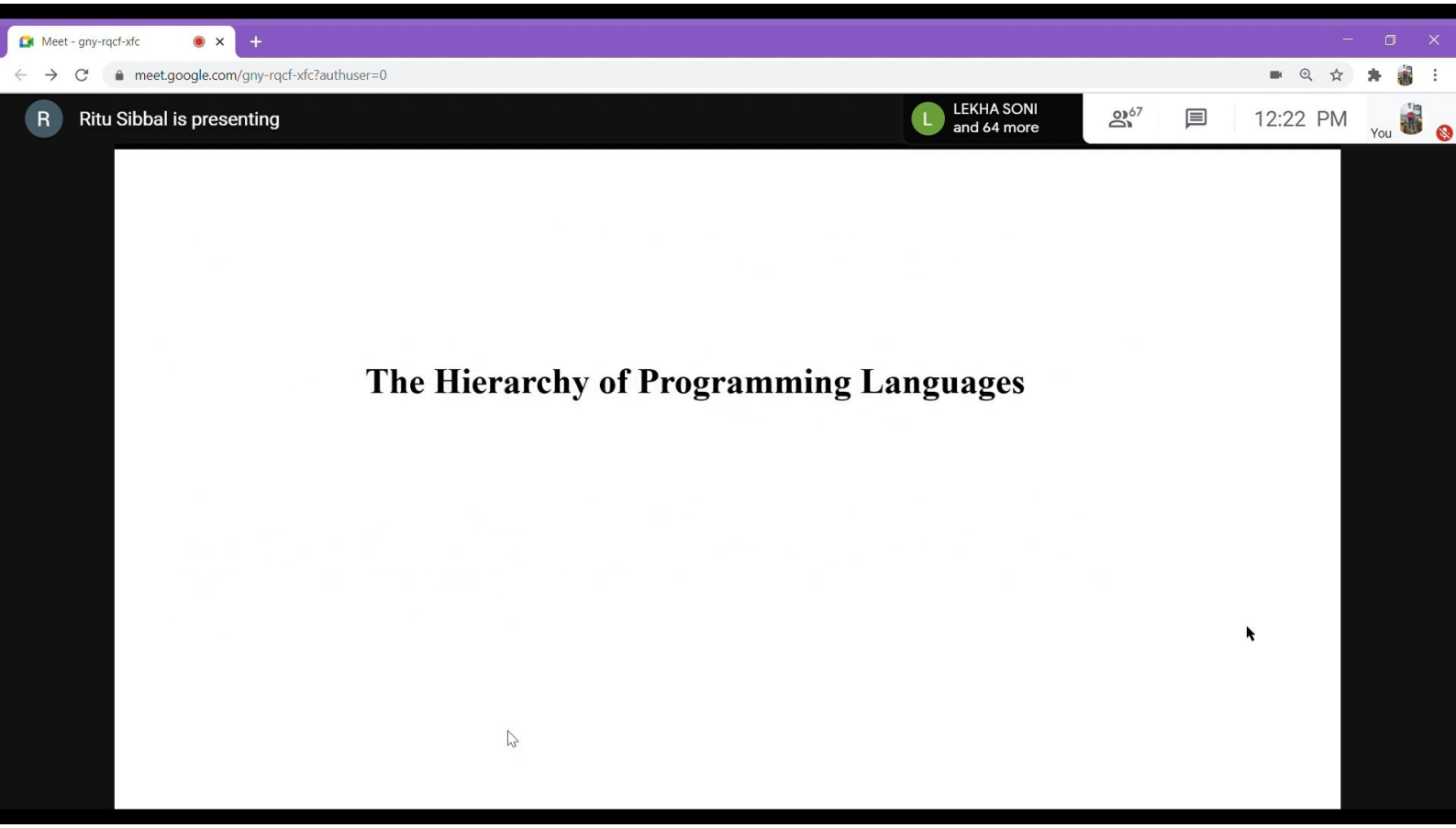
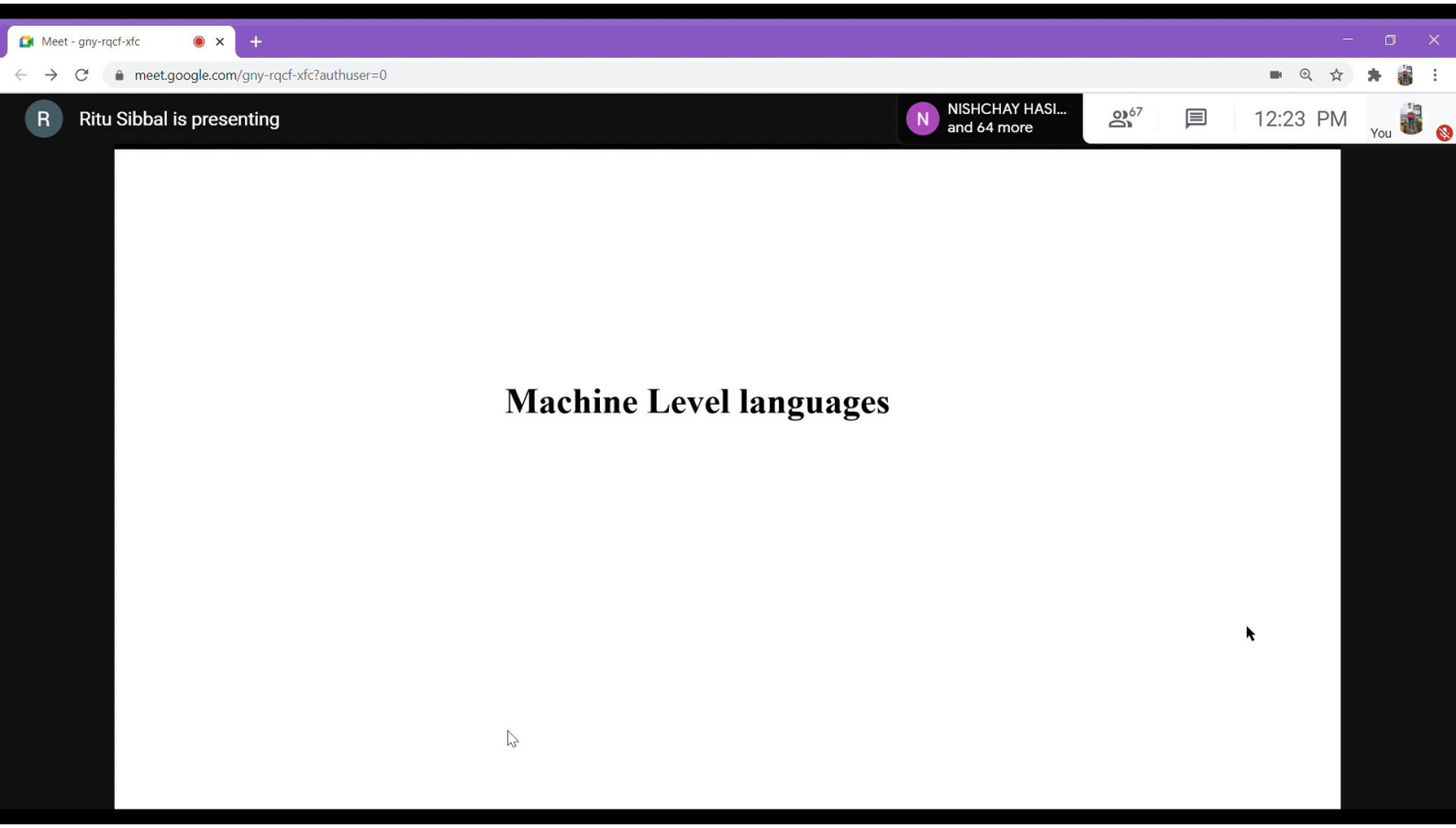
A brief history of C

- C was invented and first implemented by Dennis Ritchie on a DEC PDP-11 that used the Unix operating system.
- C is the result of a development process that started with an older language called BCPL.
- BCPL was developed by Martin Richards, and it influenced a language called B, which was invented by Ken Thompson.
- B led to the development of C in the 1970s.

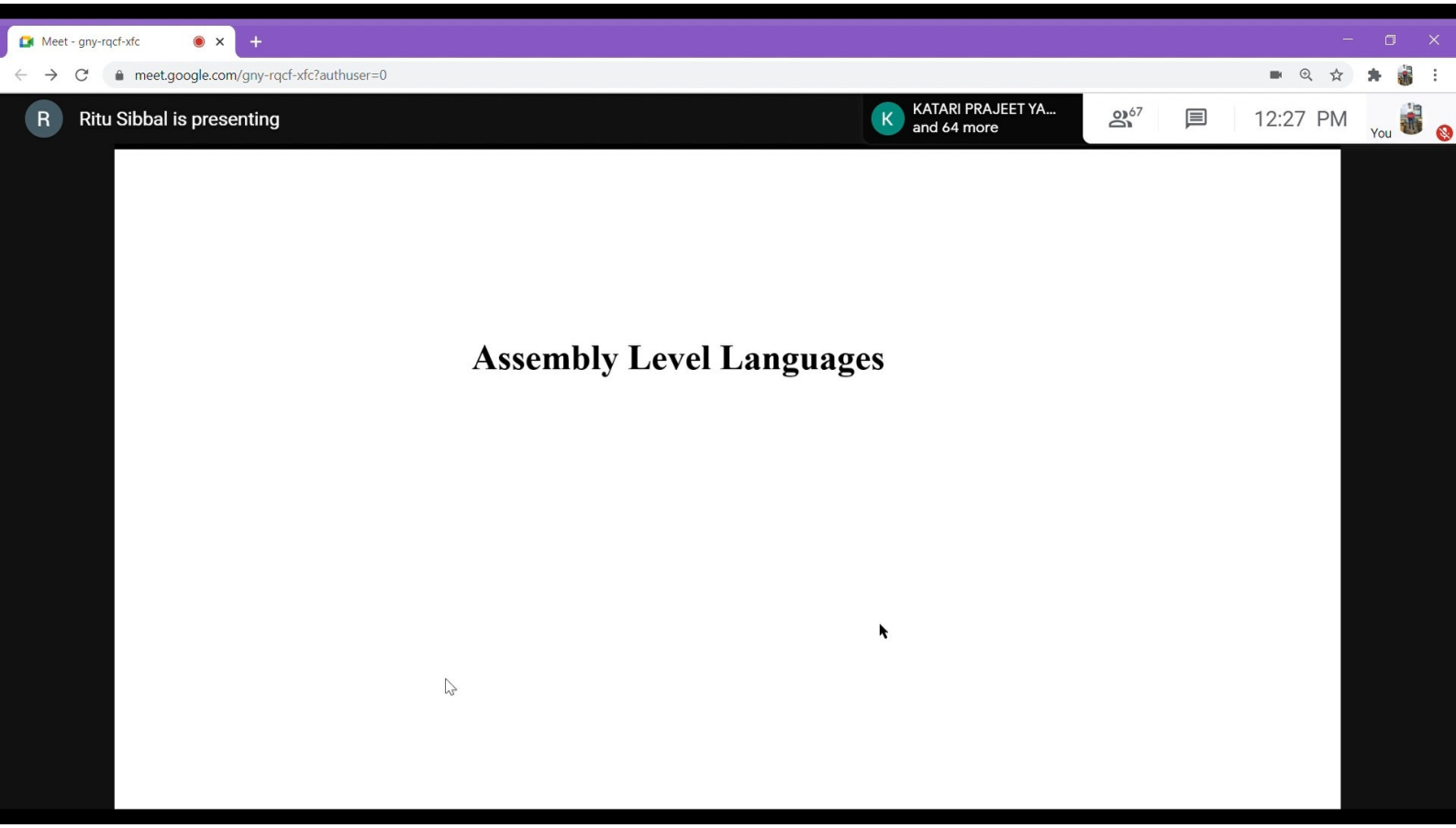
- Much of the world's code runs on C.
- It is the language upon which C++ was built, and its syntax formed the basis for Java.
-
- C is as vital today as when it was first invented. Although C's progeny (C++ and Java) are certainly important, C has a staying power that no other computer language can claim.
- ***C will always be remembered as forming the foundation for C++, it will also be known as one of the world's great programming languages on its own.***

- C is often called a *middle-level* computer language.
- It combines the best elements of high-level languages with the control and flexibility of assembly language.
- As a middle-level language, C allows the manipulation of bits, bytes, and addresses— the basic elements with which the computer functions


Meet - gny-rqcf-xfc
meet.google.com/gny-rqcf-xfc?authuser=0
R
Ritu Sibbal is presenting
L
LEKHA SONI
and 64 more
67
12:22 PM
You
The Hierarchy of Programming Languages


Meet - gny-rqcf-xfc
meet.google.com/gny-rqcf-xfc?authuser=0
Ritu Sibbal is presenting
NISHCHAY HASI...
and 64 more
67
12:23 PM
You
Machine Level languages

- It is the most basic language
- ***Machine language is a*** collection of very detailed, cryptic instructions that control the computer's internal circuitry.
- It is the natural dialect of the computer.
- Very few computer programs are actually written in machine language because
  - **First because machine language is very cumbersome to work with and**
  - **second, because every different type of computer has its own unique instruction set.**
- Thus, a machine-language program written for one type of computer cannot be run on another type of computer without significant alterations.


Meet - gny-rqcf-xfc
meet.google.com/gny-rqcf-xfc?authuser=0
R
Ritu Sibbal is presenting
K
KATARI PRAJEET YA...
and 64 more
67
12:27 PM
You
Assembly Level Languages

- Assembly language uses a symbolic representation of the actual binary code that the computer executes directly.
- Each assembly-language operation maps into a single task for the computer to perform.
- Although assembly language gives programmers the potential to accomplish tasks with maximum flexibility and efficiency, it is notoriously difficult to work with when developing and debugging a program.
- Assembly language is unstructured,
- the final program tends to be spaghetti code— a tangled mess of jumps, calls, and indexes.
- This lack of structure makes assembly-language programs difficult to read, enhance, and maintain.
- Assembly-language routines are not portable between machines with different CPUs.

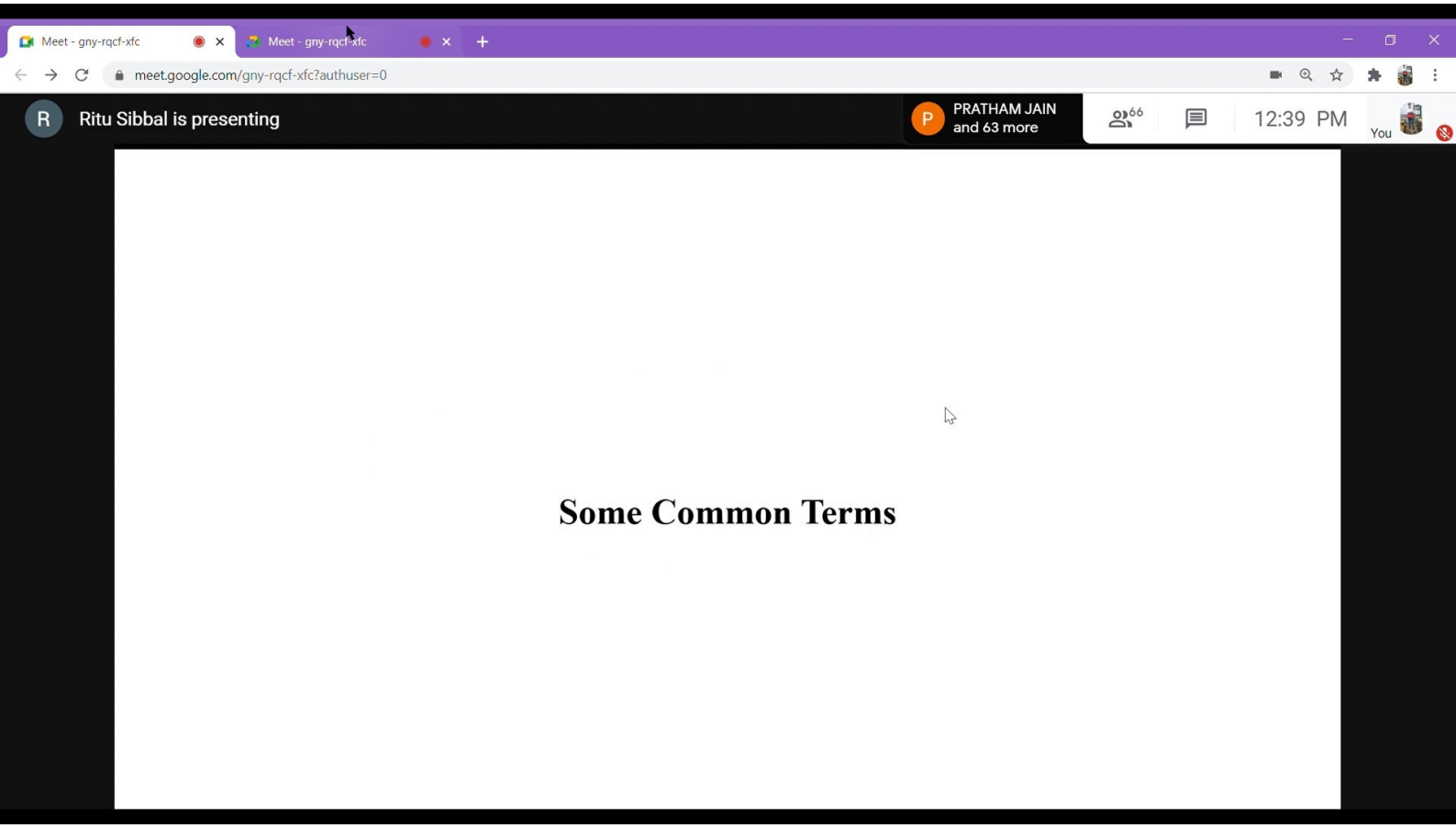
High Level programming Languages

- Usually, a computer program will be written in some ***high-level*** language, whose instruction set is more compatible with human languages and human thought processes.
- Most of these are ***general-purpose*** languages such as C.

Advantages of Coding in High level languages

- A single instruction in a high-level language will be equivalent to several instructions in machine language.
- This greatly simplifies the task of writing complete, correct programs
- The rules for programming in a particular high-level language are much the same for all computers, so that a program written for one computer can generally be run on many different computers with little or no alteration.

- A high-level language offers three significant advantages over machine language:
- ***simplicity,***
- ***uniformity*** and
- ***portability*** (i.e., machine independence).


Meet - gny-rqcf-xfc
Meet - gny-rqcf-xfc
meet.google.com/gny-rqcf-xfc?authuser=0
R
Ritu Sibbal is presenting
P
PRATHAM JAIN
and 63 more
66
12:39 PM
You
Some Common Terms

- ***Source code*** The text of a program that a user can read, commonly thought of as the program. The source code is input into the C compiler.

Object code Translation of the source code of a program into machine code, which the computer can read and execute directly. Object code is the input to the linker.

- ***Linker*** A program that links separately compiled modules into one program. It also combines the functions in the Standard C library with the code that you wrote. The output of the linker is an executable program.

Library The file containing the standard functions that your program can use. These functions include all I/O operations as well as other useful routines.

Compile time The time during which your program is being compiled.

Run time The time during which your program is executing.