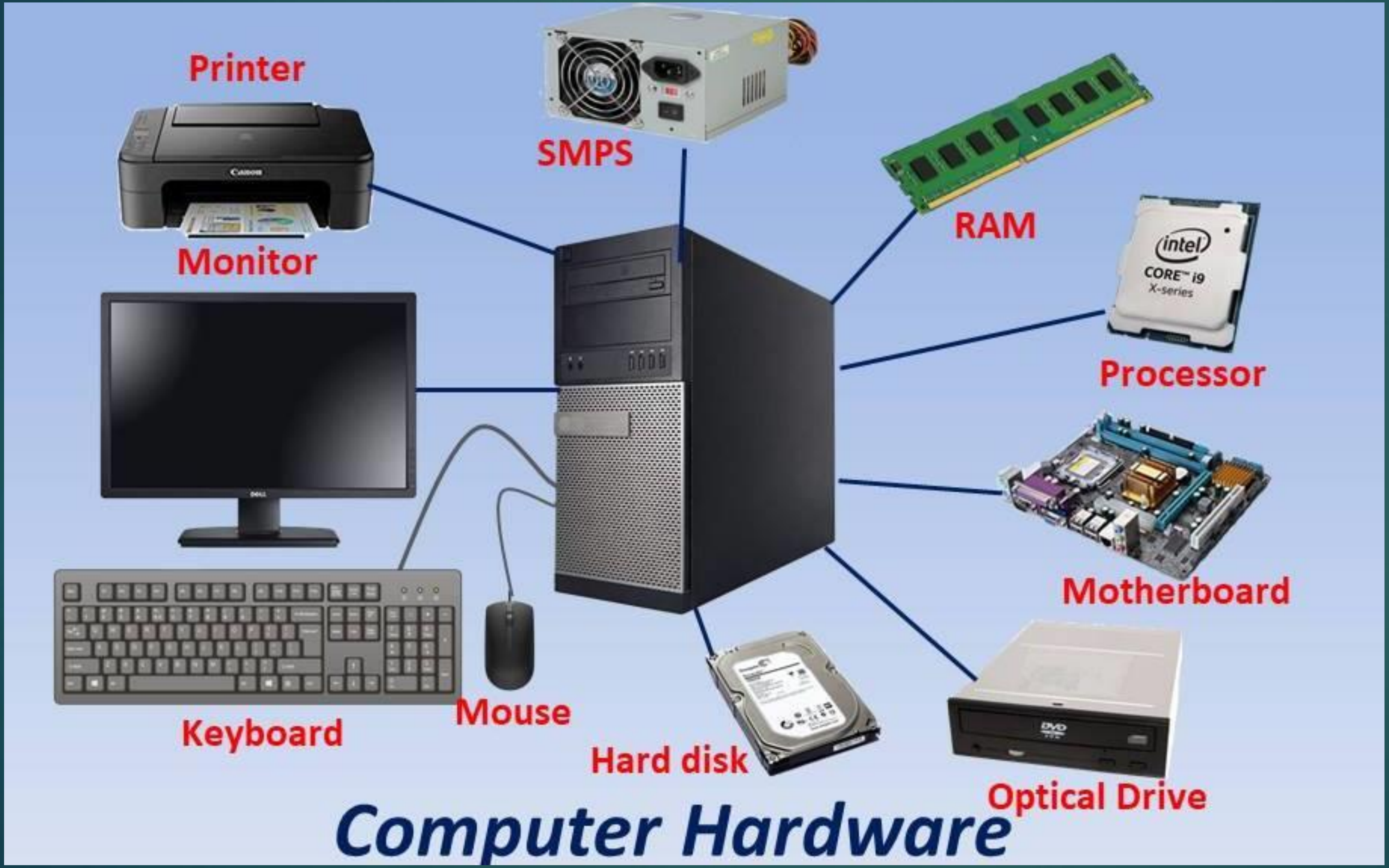
Lecture 1

Introduction for Programing

BY

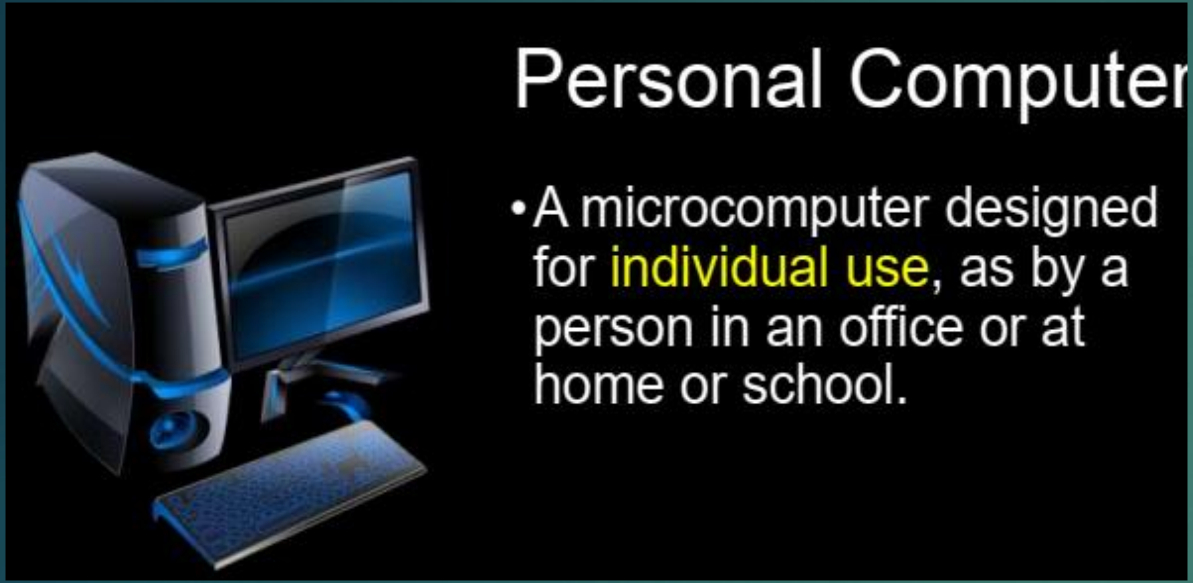
MOHAMMED ABDELFATTAH ALI

What Is a Computer?



What Is a Computer?

Computer Types

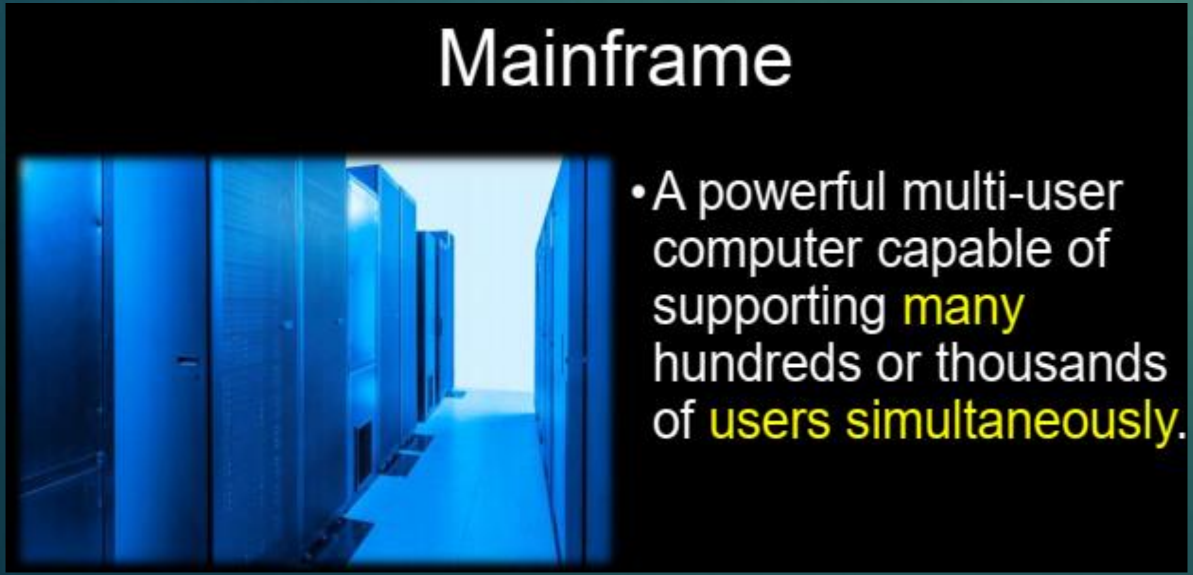


Personal Computer

- A microcomputer designed for individual use, as by a person in an office or at home or school.

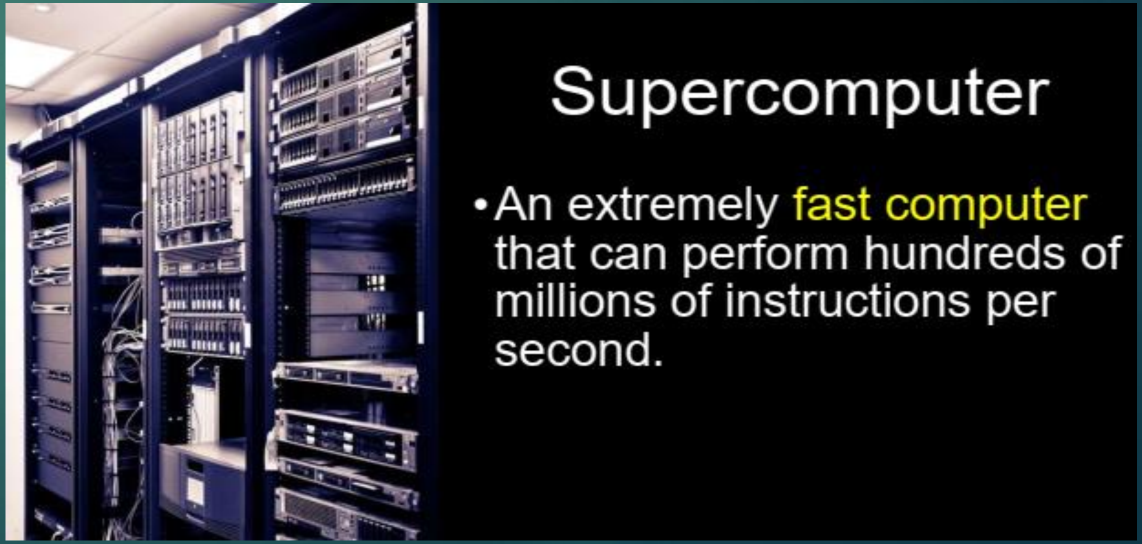
Smartphone

- A small, pocket sizes, single-user computer based on a microprocessor.



Mainframe

- A powerful multi-user computer capable of supporting many hundreds or thousands of users simultaneously.

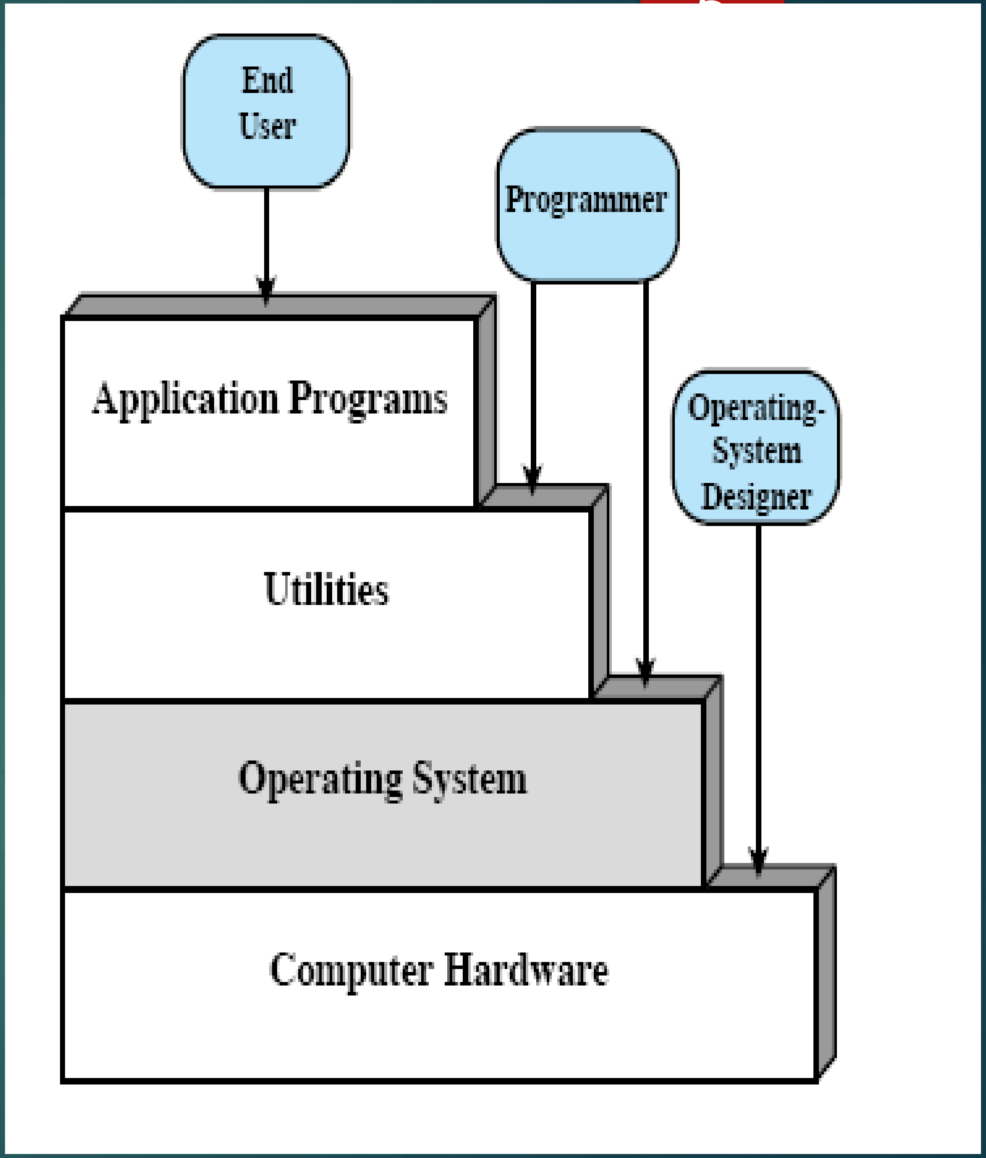


Supercomputer

- An extremely fast computer that can perform hundreds of millions of instructions per second.

Operating System

- Interface between a user and the computer hardware
- Provides an environment in which a user can execute programs
- Goals
  - Make the computer system convenient to use
  - Use the computer hardware (resources) in an efficient manner



What is Computer Science?

- Computer science is the ***study of computation***
  - Investigating problems that can be solved computationally
  - Programming languages used to describe computations
  - Machines that carry out computations
  - Theoretical limits of computation (what is or is not computable)

Algorithms

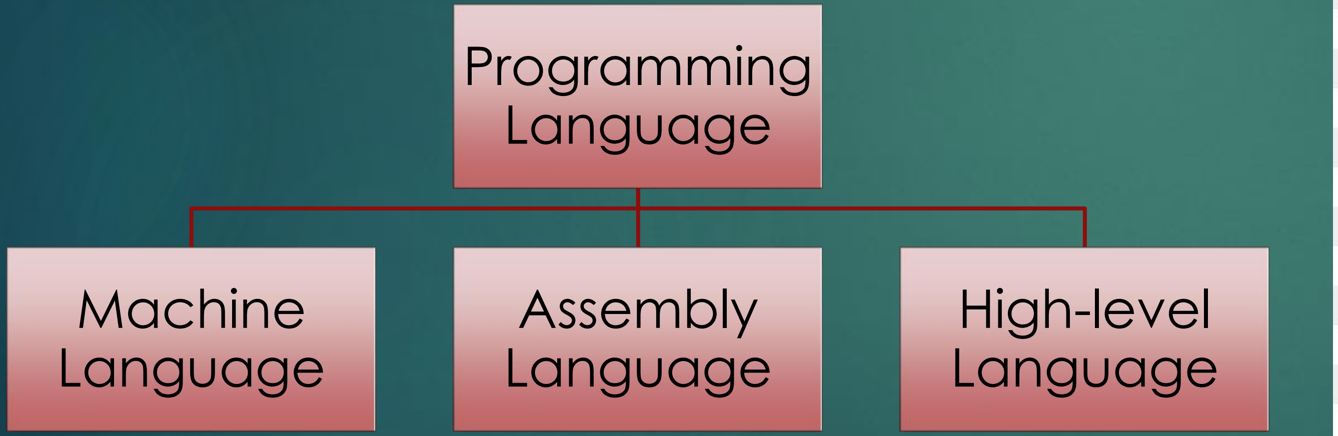
- The sequence of steps carried out during a computation are defined by an *algorithm*
    - an algorithm can be thought of as a “prescription”
    - “follow these steps and you will solve your problem”

- An algorithm includes a complete description of
    - the set of *inputs*, or starting conditions
        - a full specification of the problem to be solved
    - the set of *outputs*
        - descriptions of valid solutions to the problem
    - a sequence of *operations* that will eventually produce the output
        - steps must be simple and precise

languages

- A set of rules ,symbols and characters that allows the user to communicate with the computer.

- The three major families of languages are:

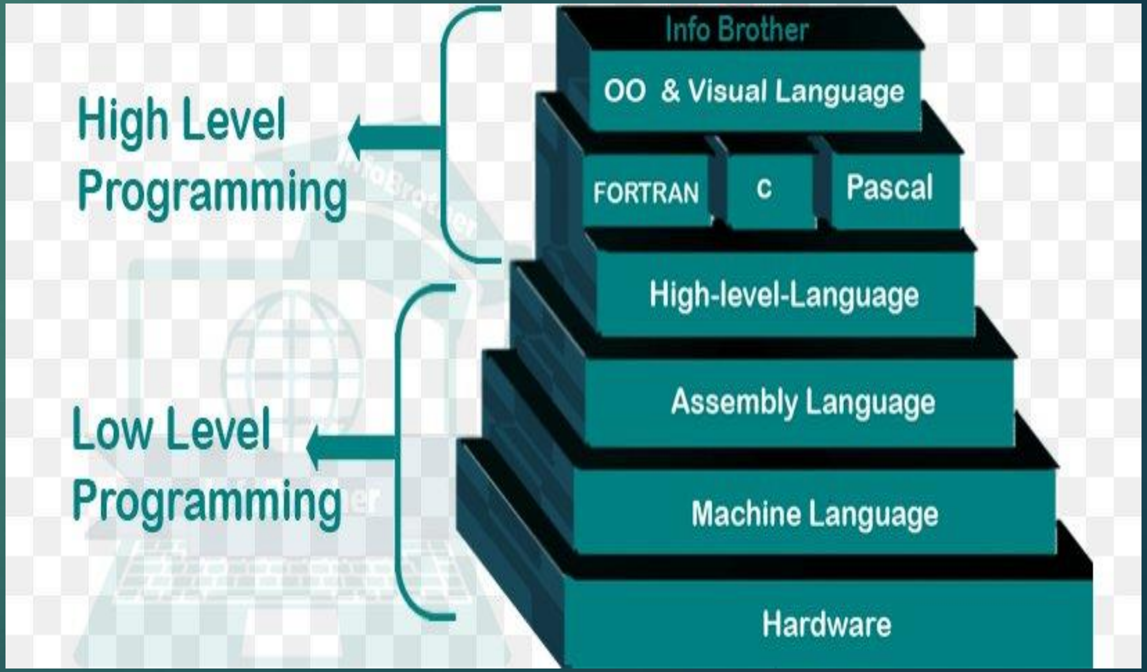


Programming Language

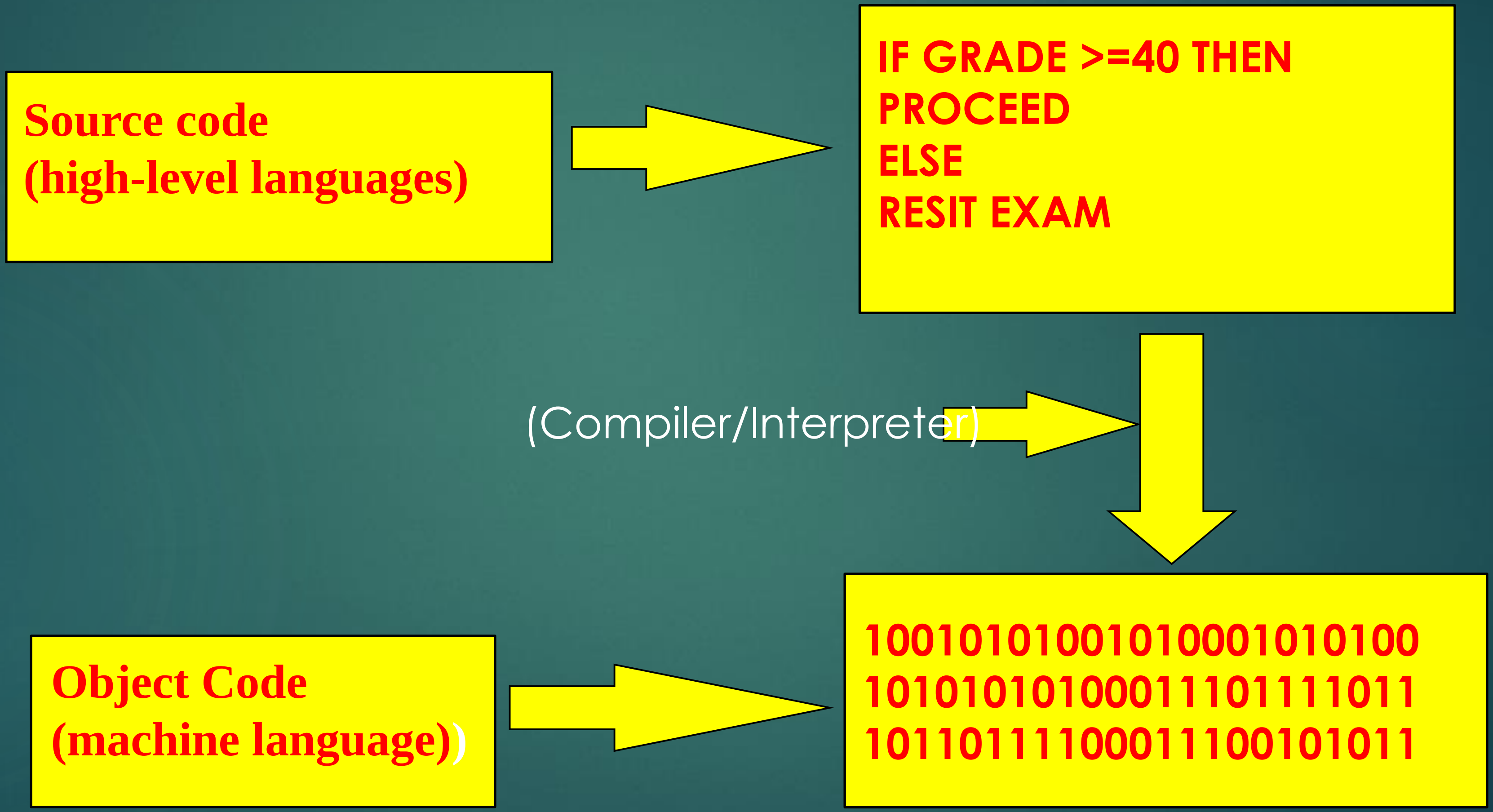
Machine Language

Assembly Language

High-level Language



Translation process



Source code
(high-level languages)

IF GRADE >=40 THEN
PROCEED
ELSE
RESIT EXAM

(Compiler/Interpreter)

10010101001010001010100
10101010100011101111011
10110111100011100101011

Object Code
(machine language))

HIGH LEVEL LANGUAGE (3RD GENERATION LANGUAGE – 1960S)

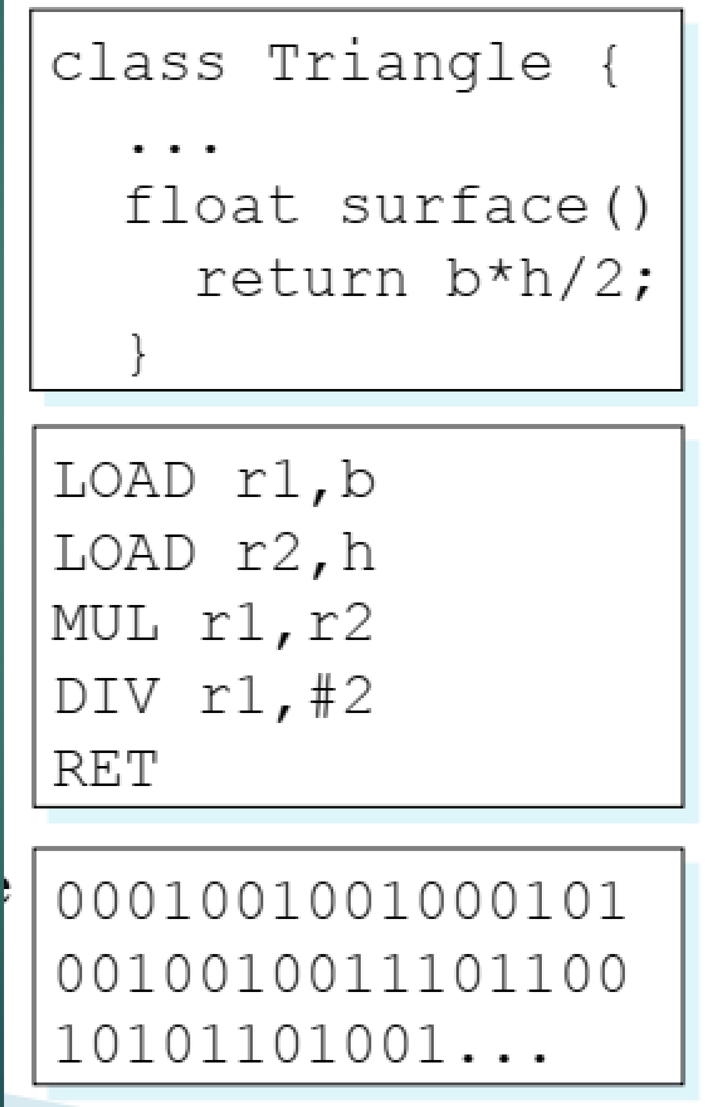
FORTRAN	Formula Translation Language
COBOL	Common Business Oriented Language
ALGOL	Algorithmic Language
RPG	Report Program Generator
APL	A Programming Language
BASIC	Beginners All Purpose Symbolic Instruction Code
PL/I	Programming Language I
PASCAL	Named after Blaise Pascal, a French Philosopher
Ada	Named after Lady Lovelace Ada
C	General Purpose Programming Language
C++	Object Oriented Programming Language
JAVA	Object Oriented Programming Language

Example of Programing Languages Levels

High-level program

Low-level program

Executable Machine code

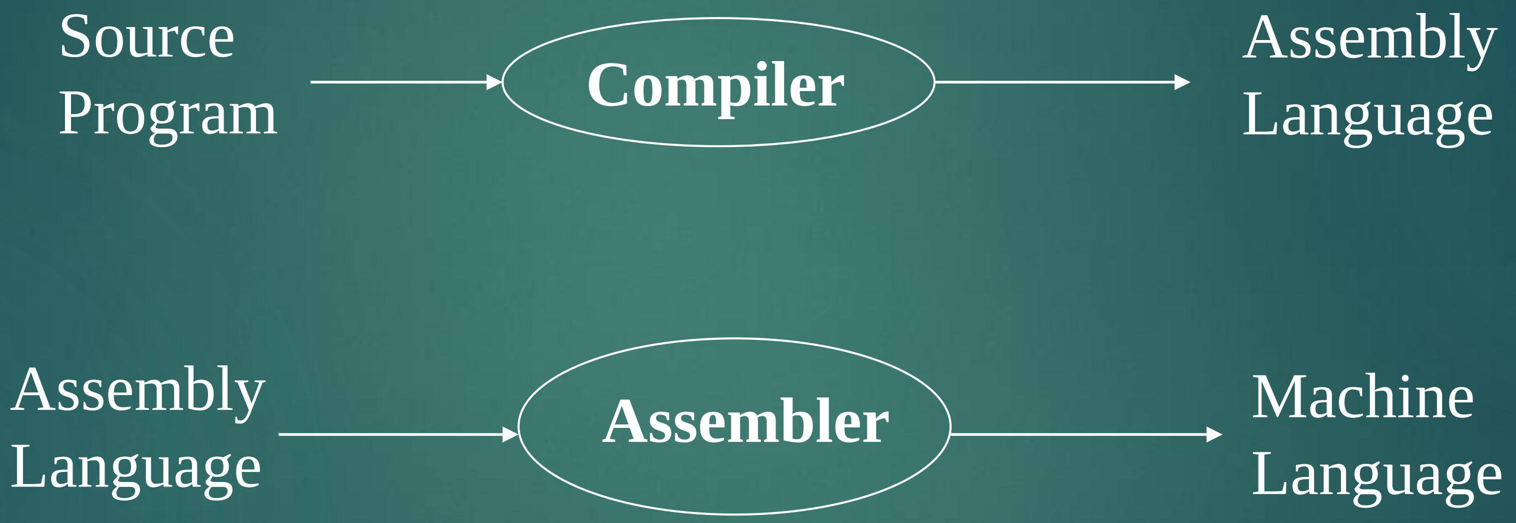


```
class Triangle {
  ...
  float surface()
    return b*h/2;
  }
```

```
LOAD r1,b
LOAD r2,h
MUL r1,r2
DIV r1,#2
RET
```

```
0001001001000101
0010010011101100
10101101001...
```

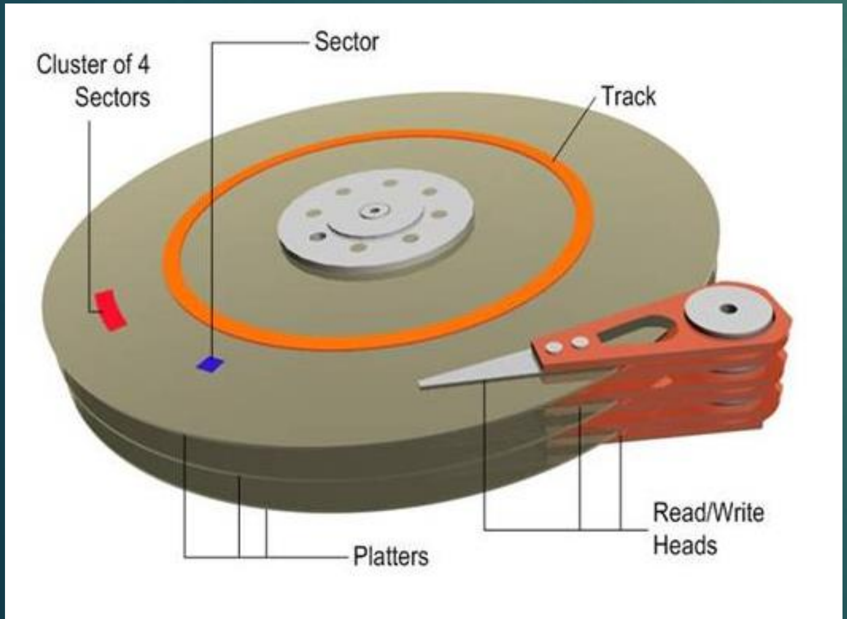
Compilation into Assembly L

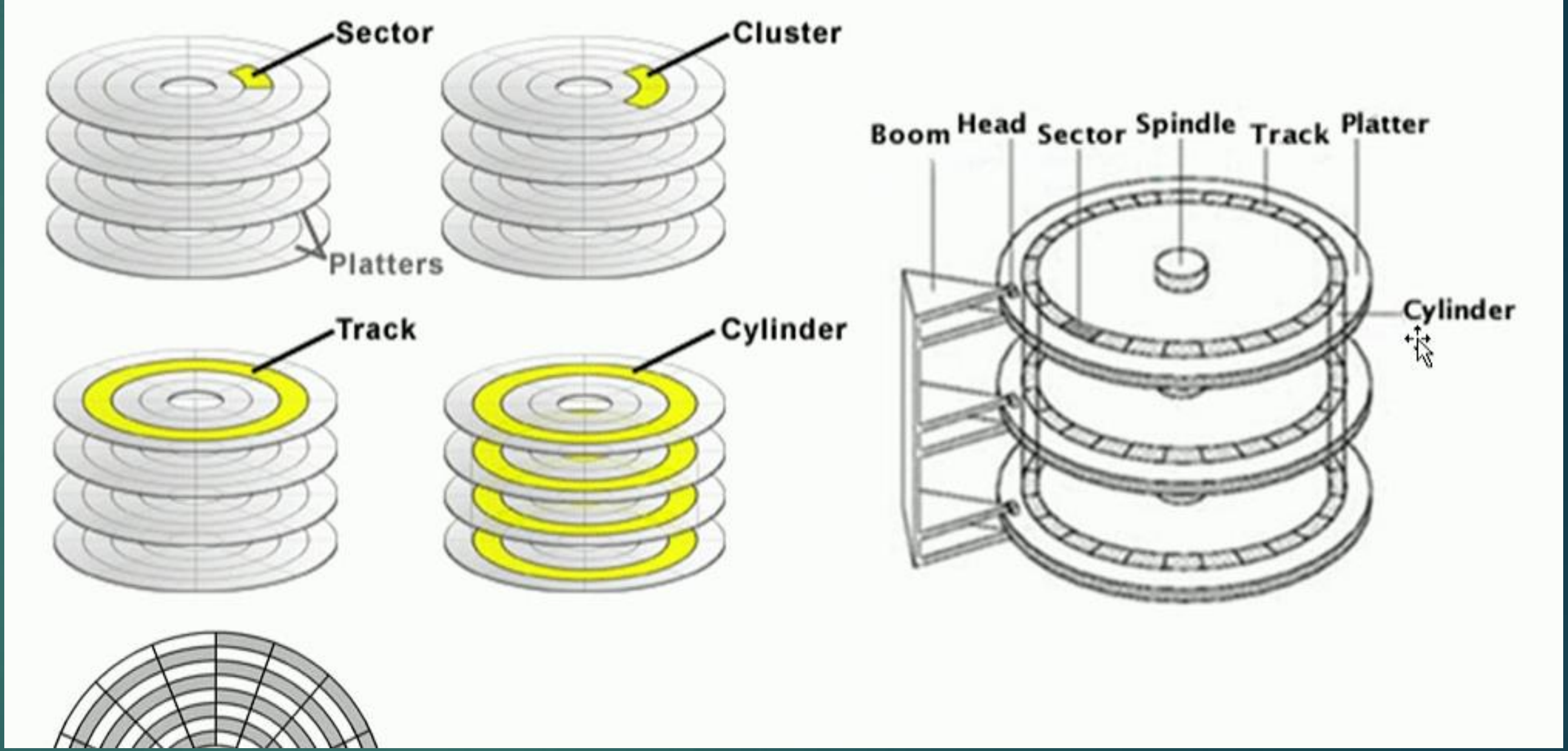


Compilation vs. Interpretation

- Compilation:
  - Syntax errors caught before running the program
  - Better performance
  - Decisions made once, at compile time
- Interpretation:
  - Better diagnostics (error messages)
  - More flexibility
  - Supports **late binding** (delaying decisions about program implementation until runtime)
    - Can better cope with PLs where type and size of variables depend on input
  - Supports creation/modification of program code on the fly (e.g. Lisp, Prolog)

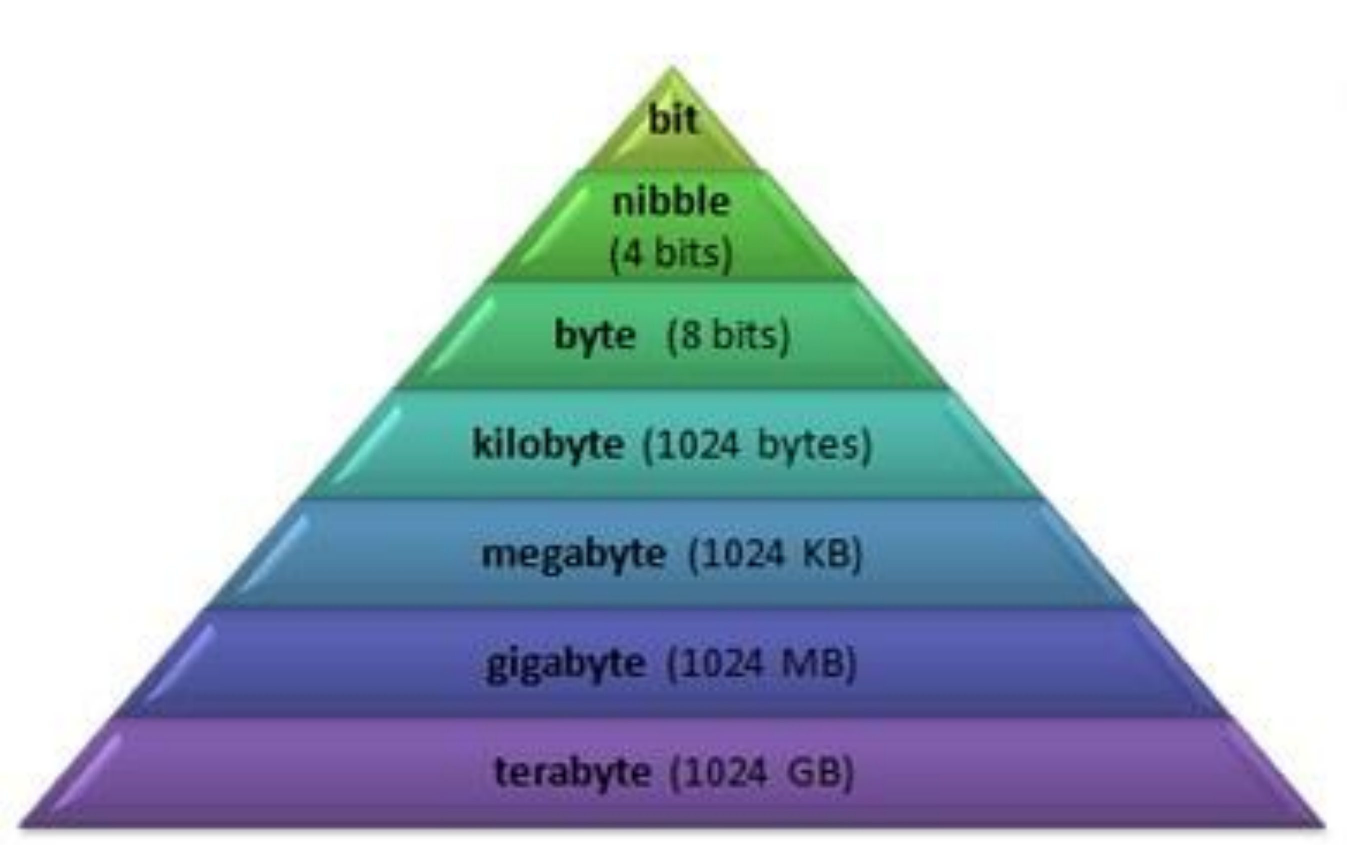
Storage (Hard Disk)



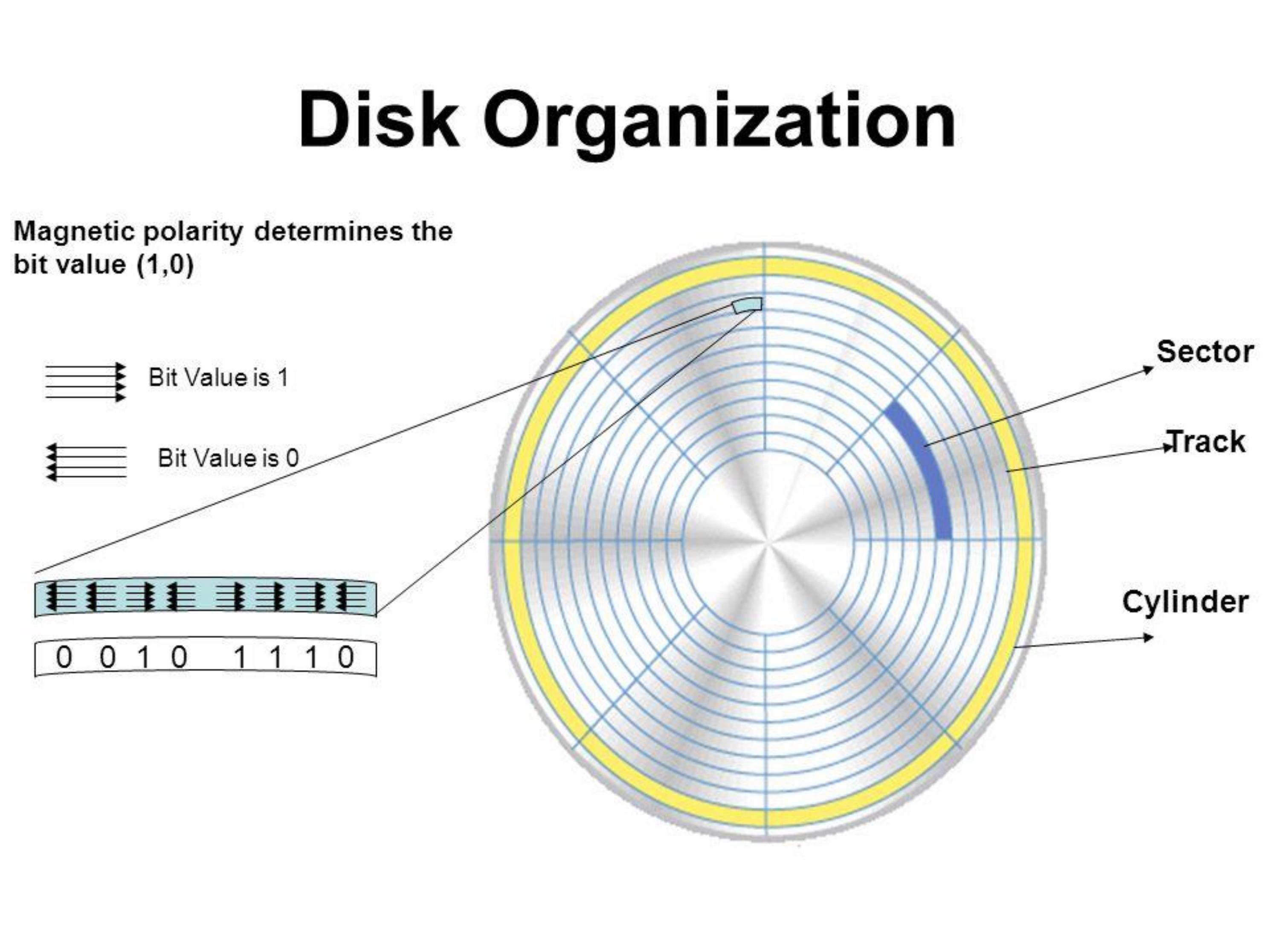


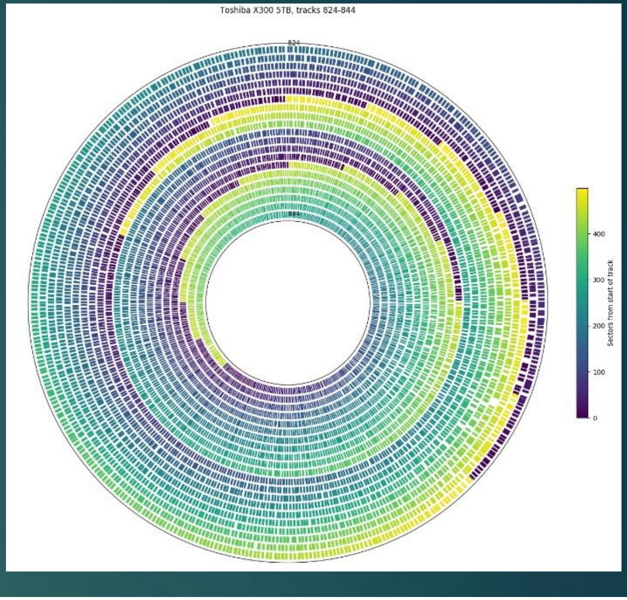
Storage Capacities

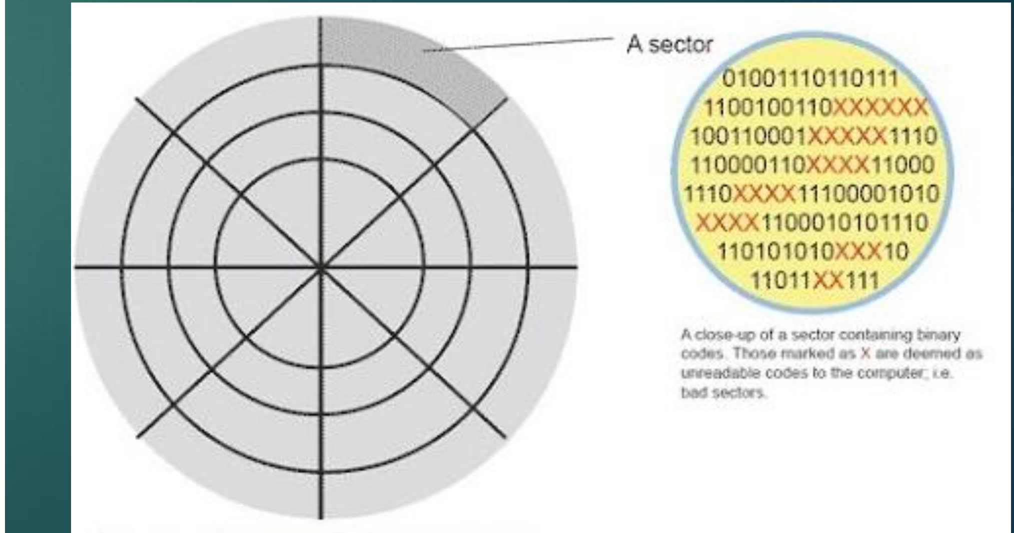
- bit – smallest capacity
- nibble = 4 bits
- byte = 2 nibbles = 8 bits
  - storage for one character
- 1 kilobyte (KB) = 1024 bytes
- 1 megabyte (MB) = 1024 KB
- 1 gigabyte (GB) = 1024 MB



Storage Capacities





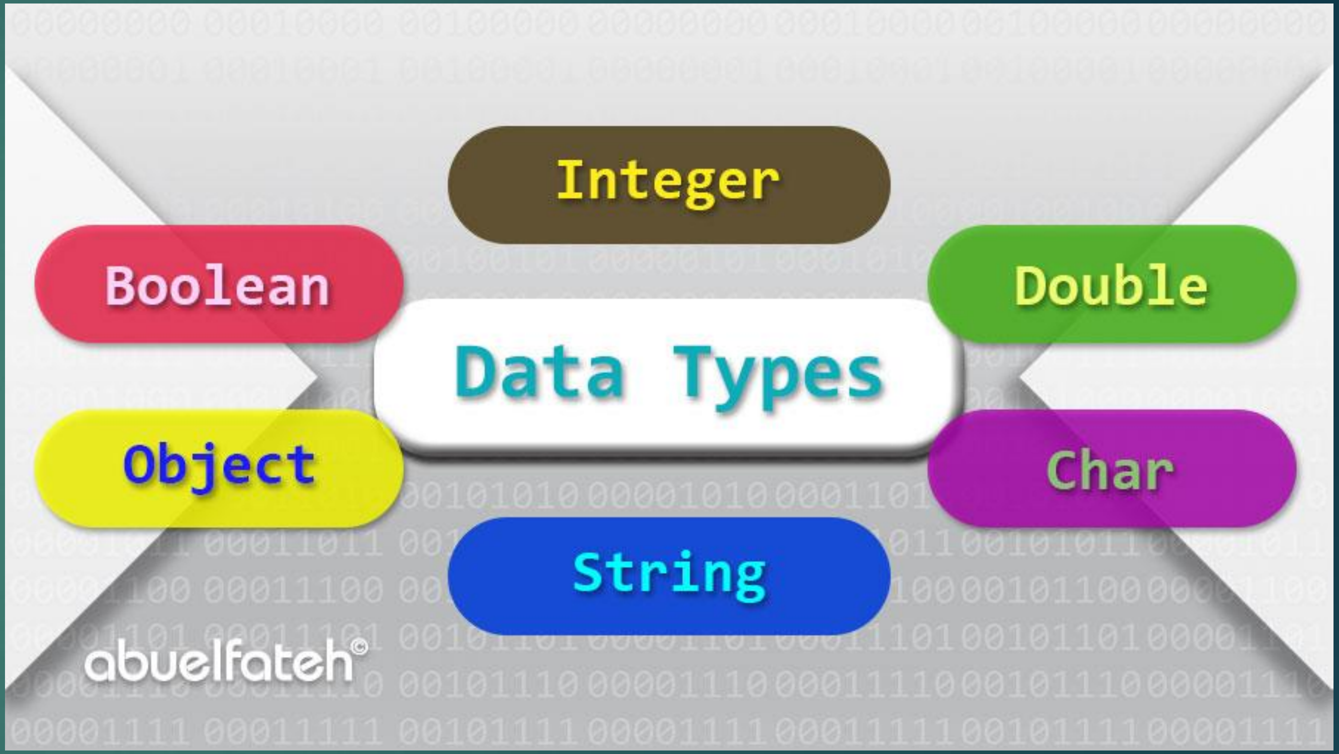


A close-up of a sector containing binary codes. Those marked as X are deemed as unreadable codes to the computer; i.e. bad sectors.

Illustration of a Hard Disk Platter divided in sectors

Types of Data (Value, Built-In)

- Numeric Data
- Character data (alphanumeric)
- Boolean data (TRUE/FALSE)
- Other Data Types (Enum, … etc.)



Data Manipulation

=	assignment
+	addition
-	subtraction
*	multiplication
/	division
%	modulus
++	increment by one
--	decrement by one

Assignment Operations

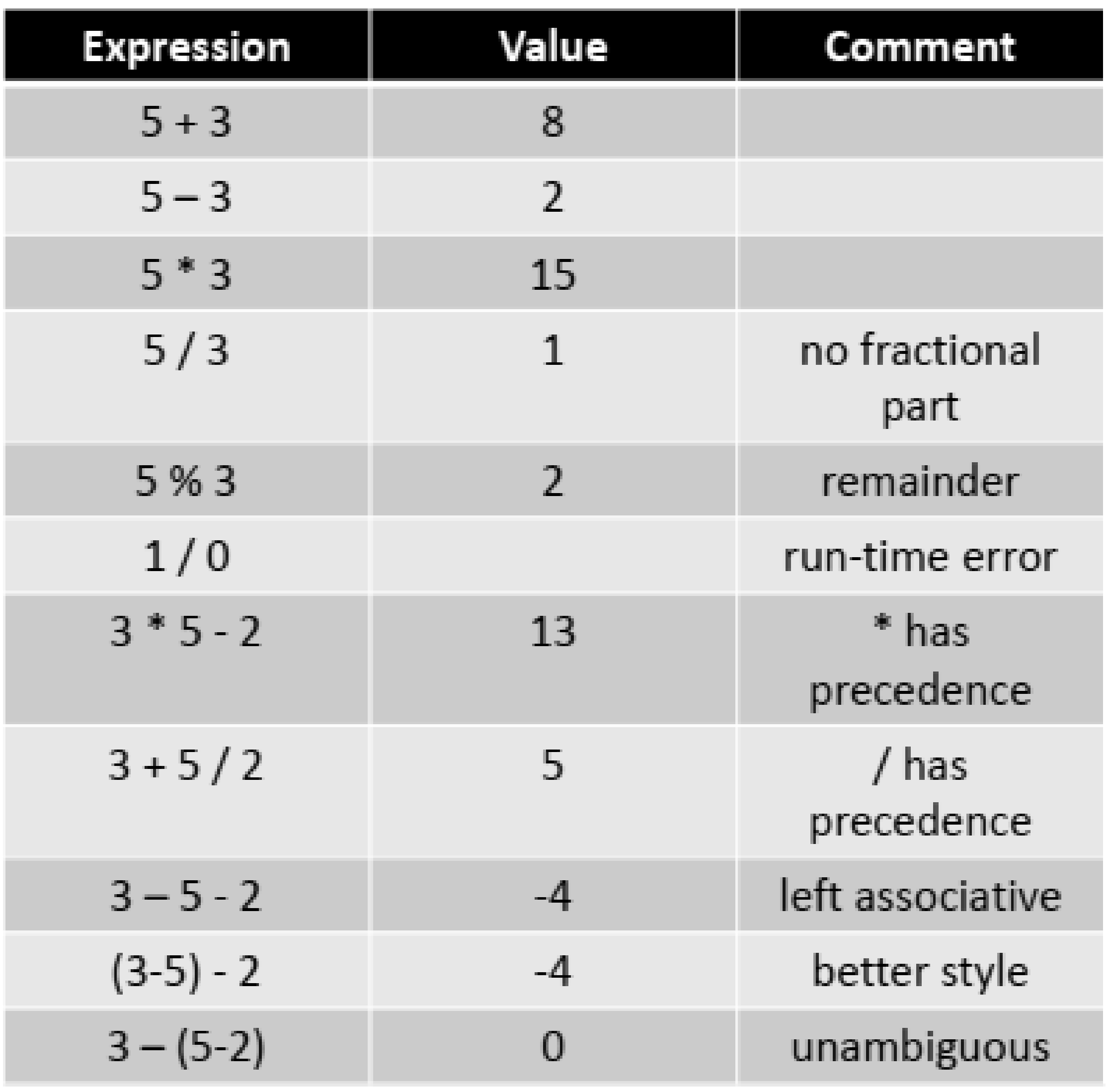
```
Set x = 1

Set x = x + 1

Set counter = counter + 1
```

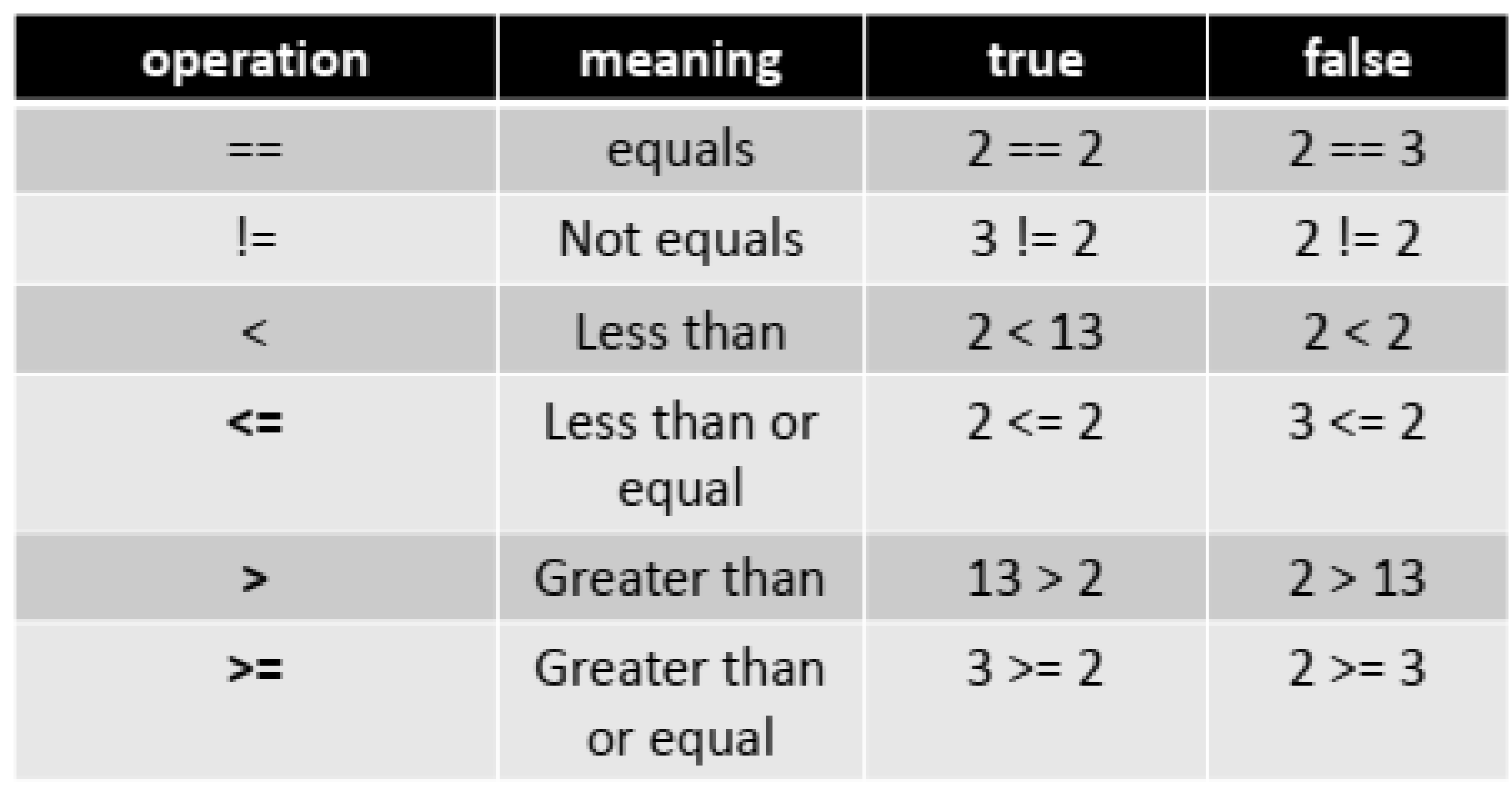
- Assignment statements change the value in a variable Take the value of `counter`, add 1, and store the result back in the same variable.

Integer Data Type



Expression	Value	Comment
5 + 3	8	
5 – 3	2	
5 * 3	15	
5 / 3	1	no fractional part
5 % 3	2	remainder
1 / 0		run-time error
3 * 5 - 2	13	* has precedence
3 + 5 / 2	5	/ has precedence
3 – 5 - 2	-4	left associative
(3-5) - 2	-4	better style
3 – (5-2)	0	unambiguous

Boolean Comparison



operation	meaning	true	false
==	equals	2 == 2	2 == 3
!=	Not equals	3 != 2	2 != 2
<	Less than	2 < 13	2 < 2
<=	Less than or equal	2 <= 2	3 <= 2
>	Greater than	13 > 2	2 > 13
>=	Greater than or equal	3 >= 2	2 >= 3

Thanks