

Java Full Stack Development

Python

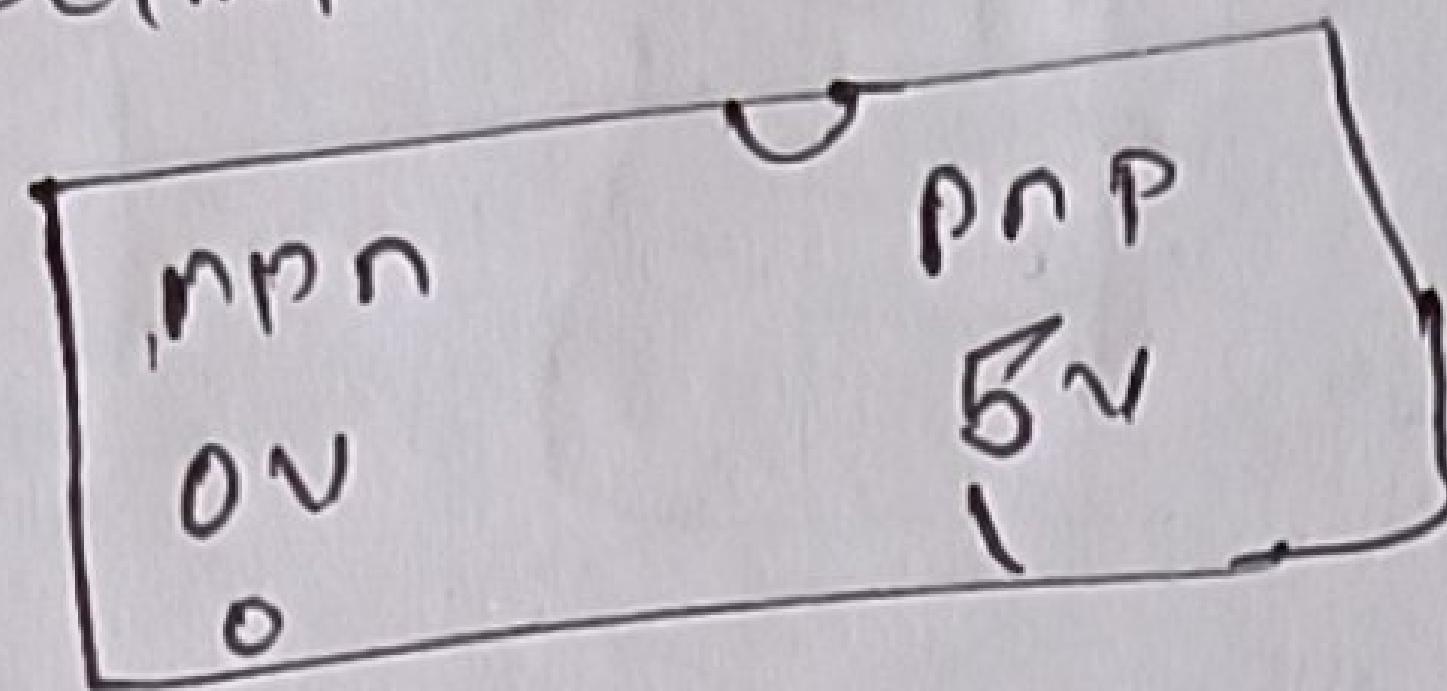
Angular

Fundamentals of Programming

⇒ CPU / microprocessor controls all the function of a device either that is mobile or Computer.

⇒ CPU is a ^{Technology} semiconductor device which is made out of transistors.

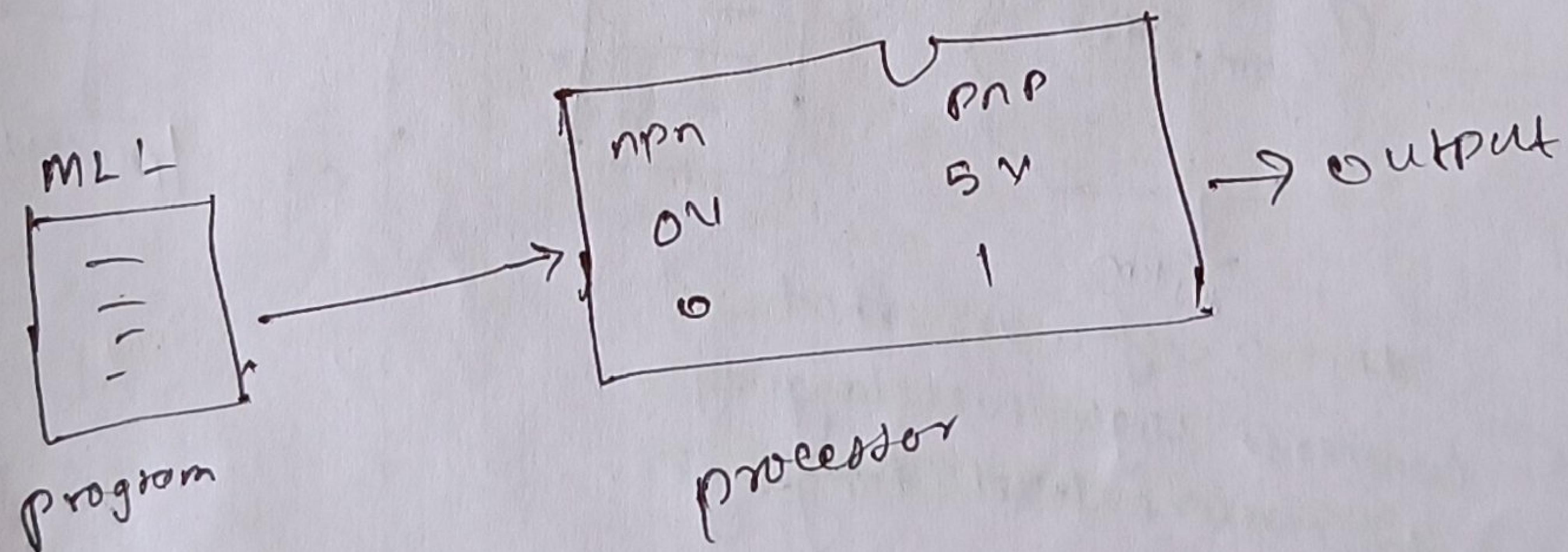
CPUIMP (Semiconductor Technology) → ^{is very fast}
in nature



⇒ Transistor is a device than only understand two things low voltage and high voltage.

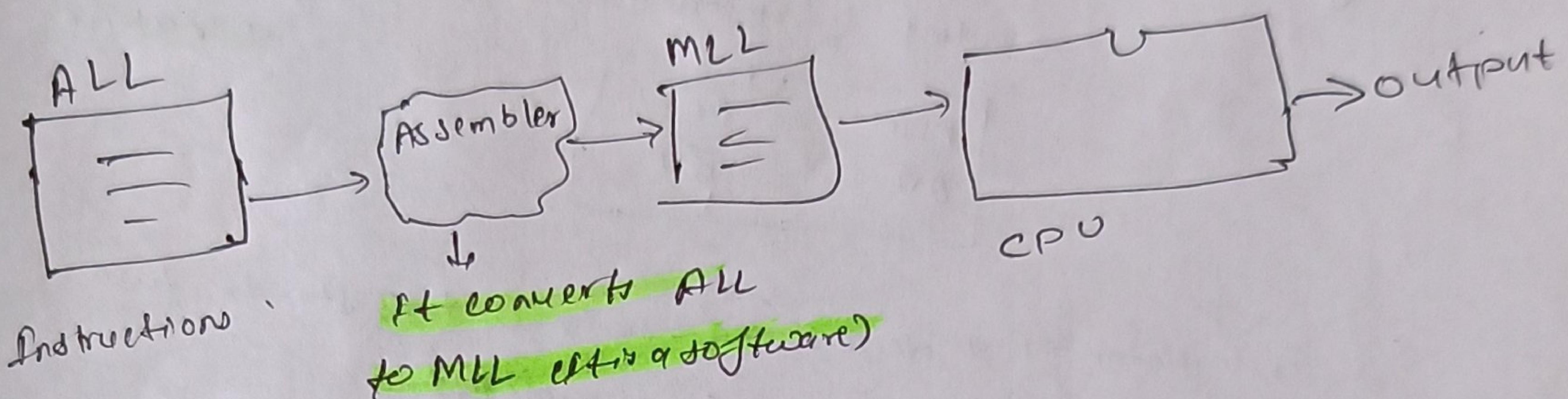
Note: machine Computer understands in the form of 0's and 1's. that's why it is called machine Level language.

For many instruction in one place it is called program



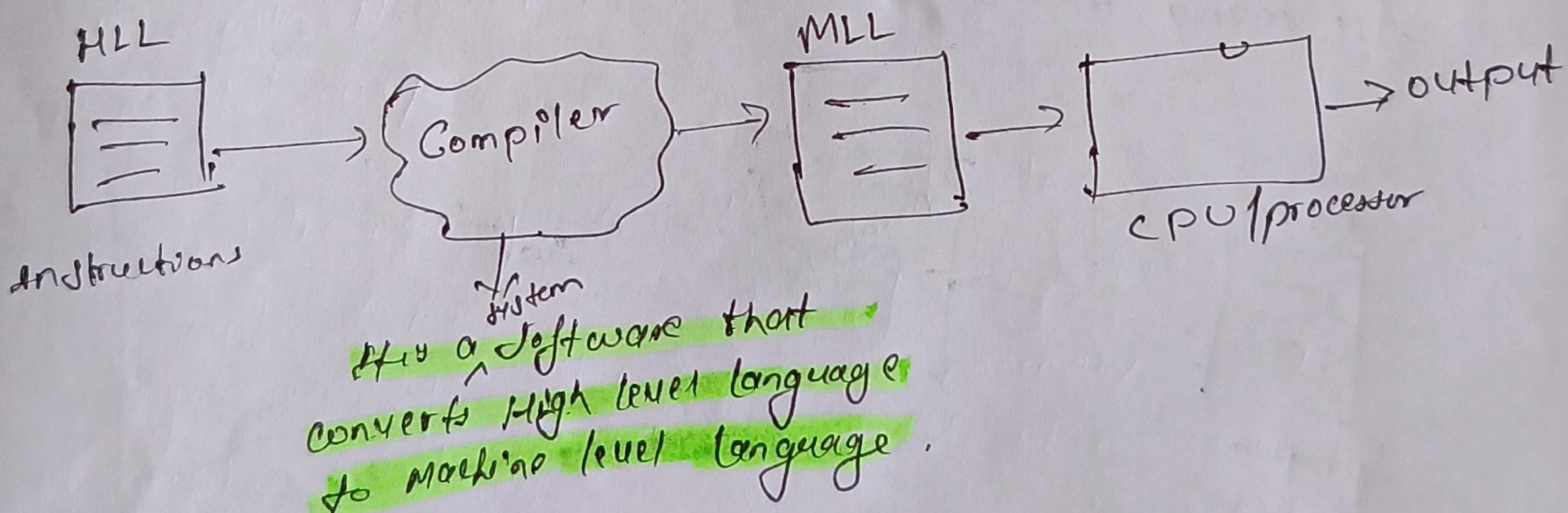
Note : Assembly language (ALL)

→ The style of writing pnumonics instructions in the form of neumonics i.e ADD for addition, SUB for subtraction etc. is called Assembly level language.

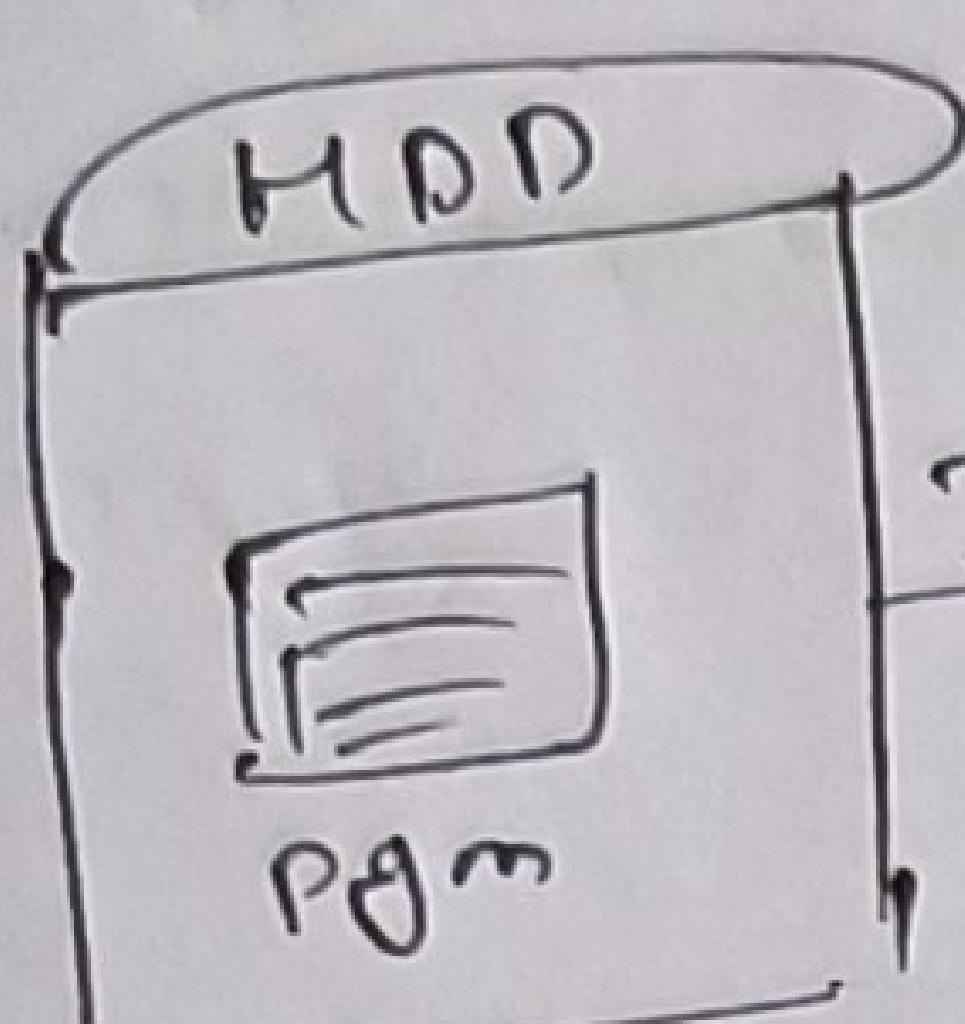
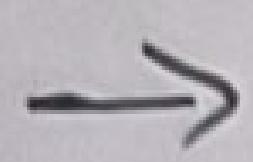


High level language (HLL)

→ Writing instruction by using symbols and english commands is called high level language.

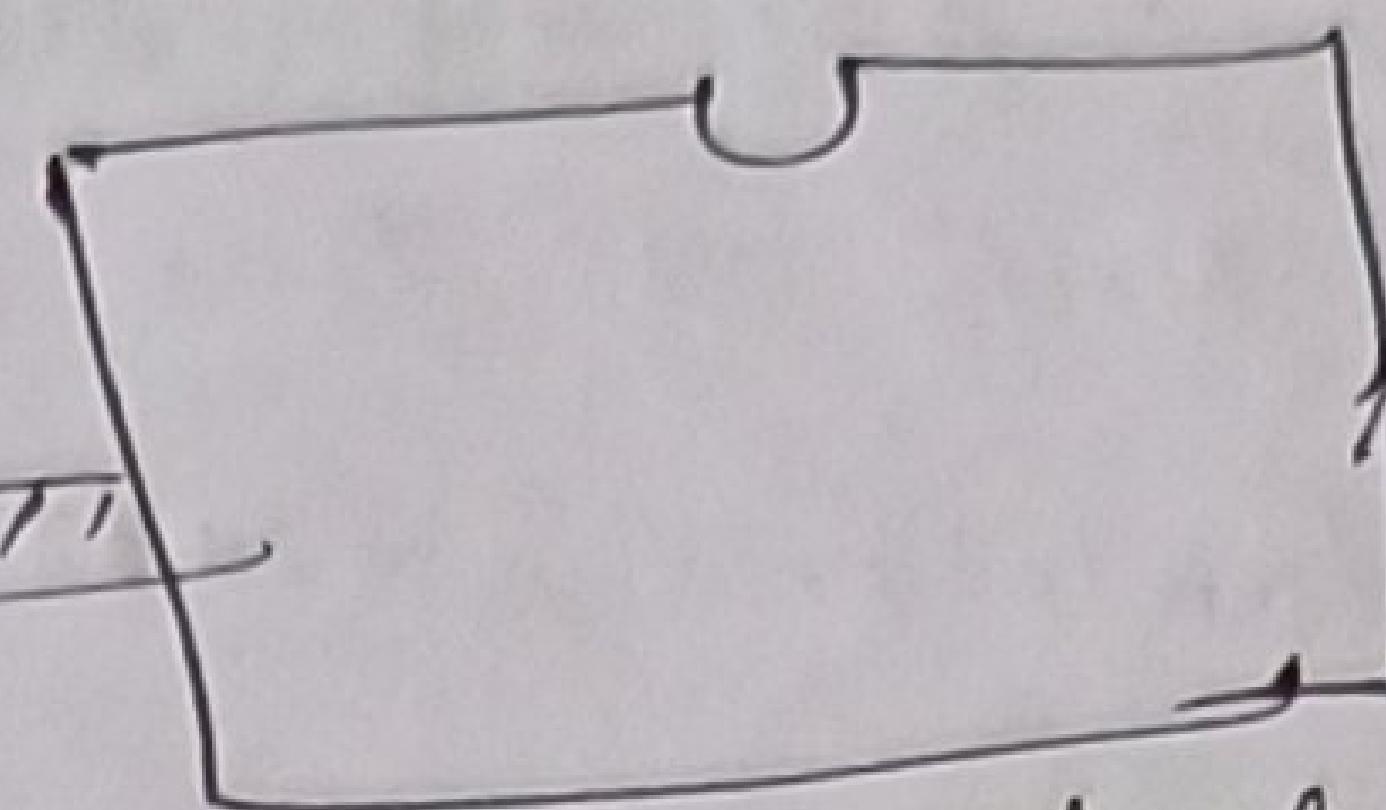


In what file?



(magnetic technology)

↓
it is slow in nature



(Semiconductor
Technology)

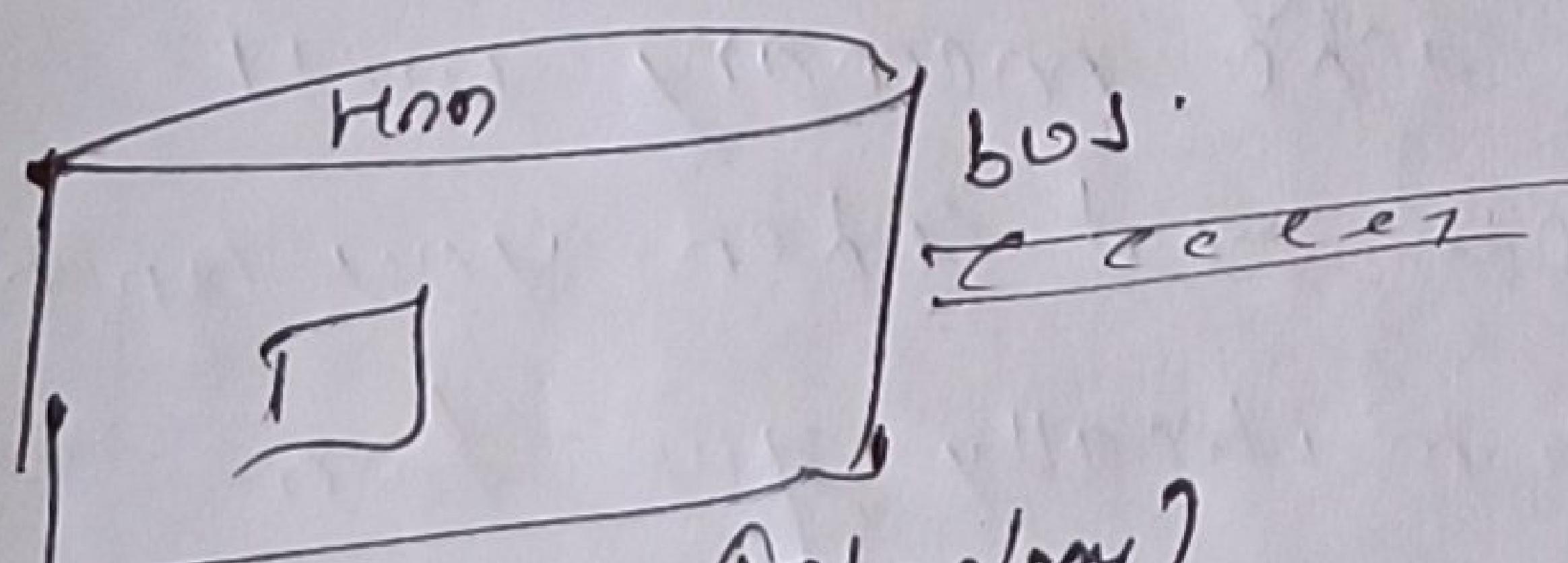
↑
it is very fast
in nature.

* BUS → it is used to transfer instruction from one point to another point.

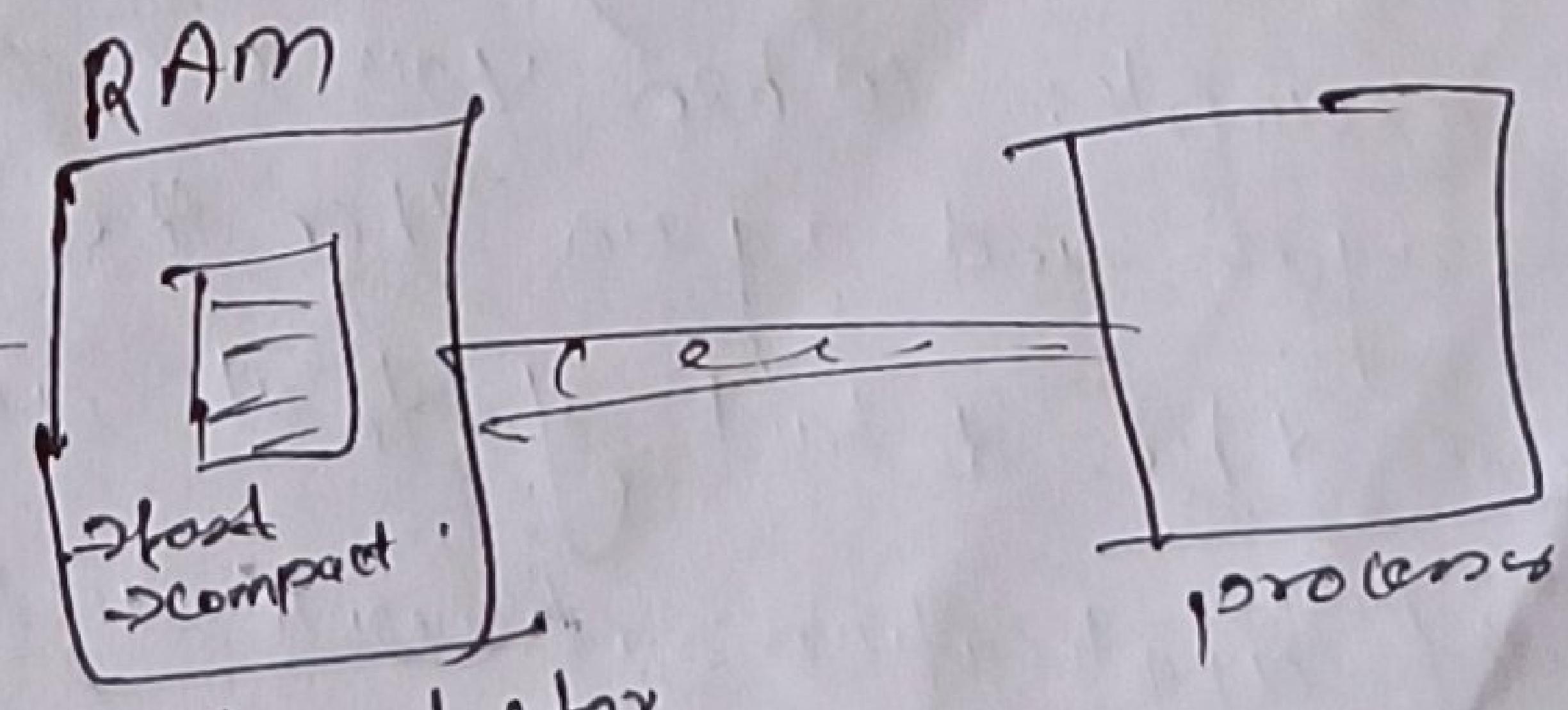
P

→ Our program is in the HDD (Hard Drive Device). So, in order to run it, Bus take out program instruction to the processor and it immediately gives the result since it is fast in nature but Hard disk sends slowly one by one. so, there is a speed mismatch between Hard disk and processor. So, there is decrease in efficiency.

+ In order to overcome speed mismatch. one more semiconductor technology device is introduce that is RAM.
(Random Access memory)



(magnetic technology)
(secondary memory)



Semiconductor
Technology
(primary memory)

→ Due to the RAM the speed mismatch problem is solved i.e. both RAM and processor are semiconductor technology device.

• RAM is less fast and compact but it is volatile in nature i.e. that needs continuous power supply if even for milliseconds the power is gone the data will be deleted.

Note:

Saving → the process of storing data from RAM to hard disk permanently.

Loading

→ It's the process of getting program to the RAM for execution is called Loading.

Note:

Byte: It is the place where data is stored in RAM

File: It is the place where data is stored in HDD

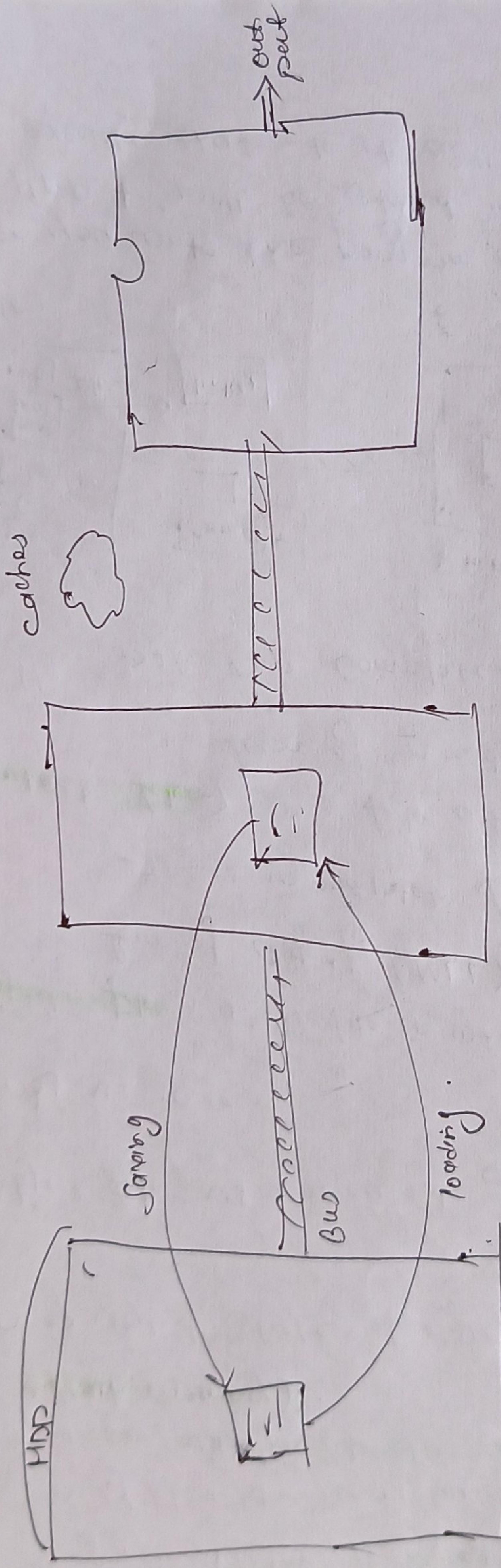
Register: It is the place where data is stored in processor.

Cache Memory

→ It's the memory which is nearest to the processor.

for example: when you are opening any application again and again then the cache memory will store the detail of that app and when you open that application it will automatically give from cache memory to processor.

→ It is volatile memory.



HDD

RAM

- Processor -
- Semiconductor technology
- Stores in form of Register
- faster
- RAM
- Semiconductor Technology
- volatile
- store in form of Bytes
- faster than HDD
- primary memory
-
- HDD
- magnetic technology
- store in form of file
- non volatile
- Secondary memory

SSD → It is the updated version of HDD but faster than HDD because it is also made up of semiconductor technology.

→ It is a semiconductor technology then how it is non-volatile due to the concept of flashing (flash memory).

SSD → Solid State Drive

Object file vs executable file.

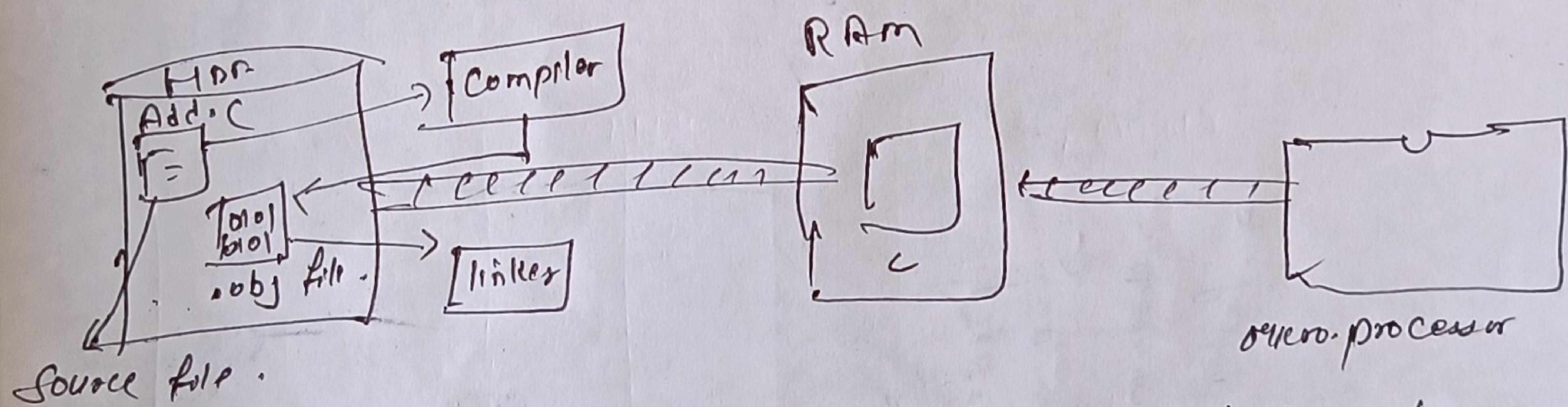
→ .obj vs .exe

Object file → It is a file in which is stored in the form of O and I (MLL)

→ It is incomplete file.

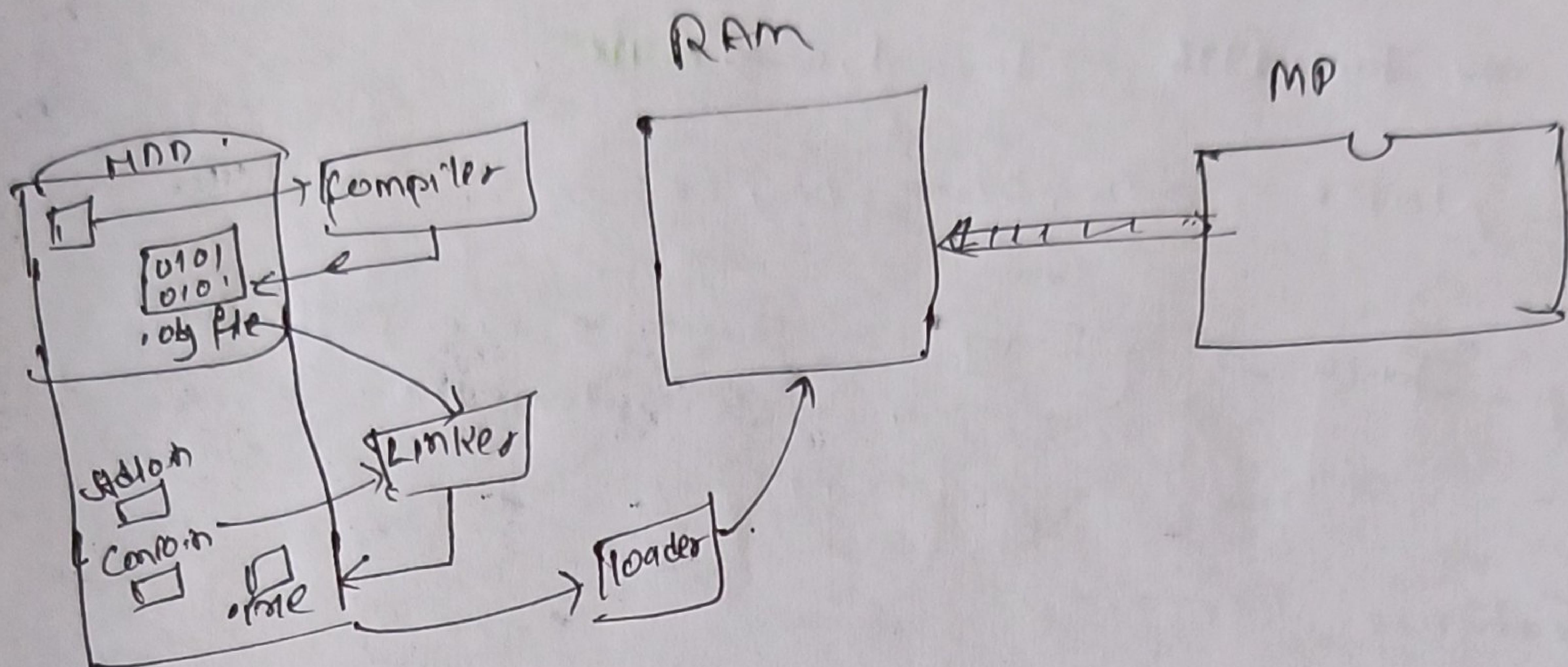
Executable file → It is a file which is stored in the form of O and I (MLL)

→ It is a complete file.



→ When you write the program in RAM and store it in the HDD it will be stored in the form of source code and to execute that code it will be in the form of compiled

and converted in the .obj file



→ The object file is incomplete so, the Linker is there which will make the program complete with the help of library files and convert them to complete executable file i.e. .exe file. and there is a concept of loader which will load the .exe file to RAM for execution.

7th Oct

Java :-

↳ 1991

↳ Sun Microsystems

↳ James Gosling

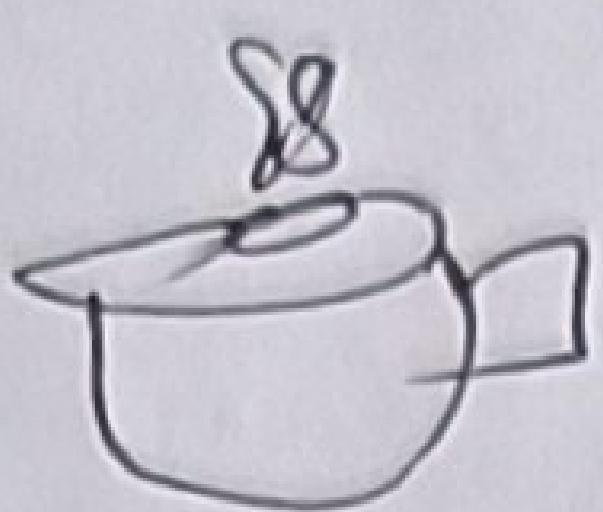
⇒ Easy to understand, object oriented

⇒ portable / platform independent of WORD

⇒ 1995 → Alpha & Beta version of freely downloadable open source

⇒ Jan 1996 → Java 1.0 → **oak**.

Name + oak, green, ix, etc — etc.

Java →  → Java is one coffee land in Indonesia.

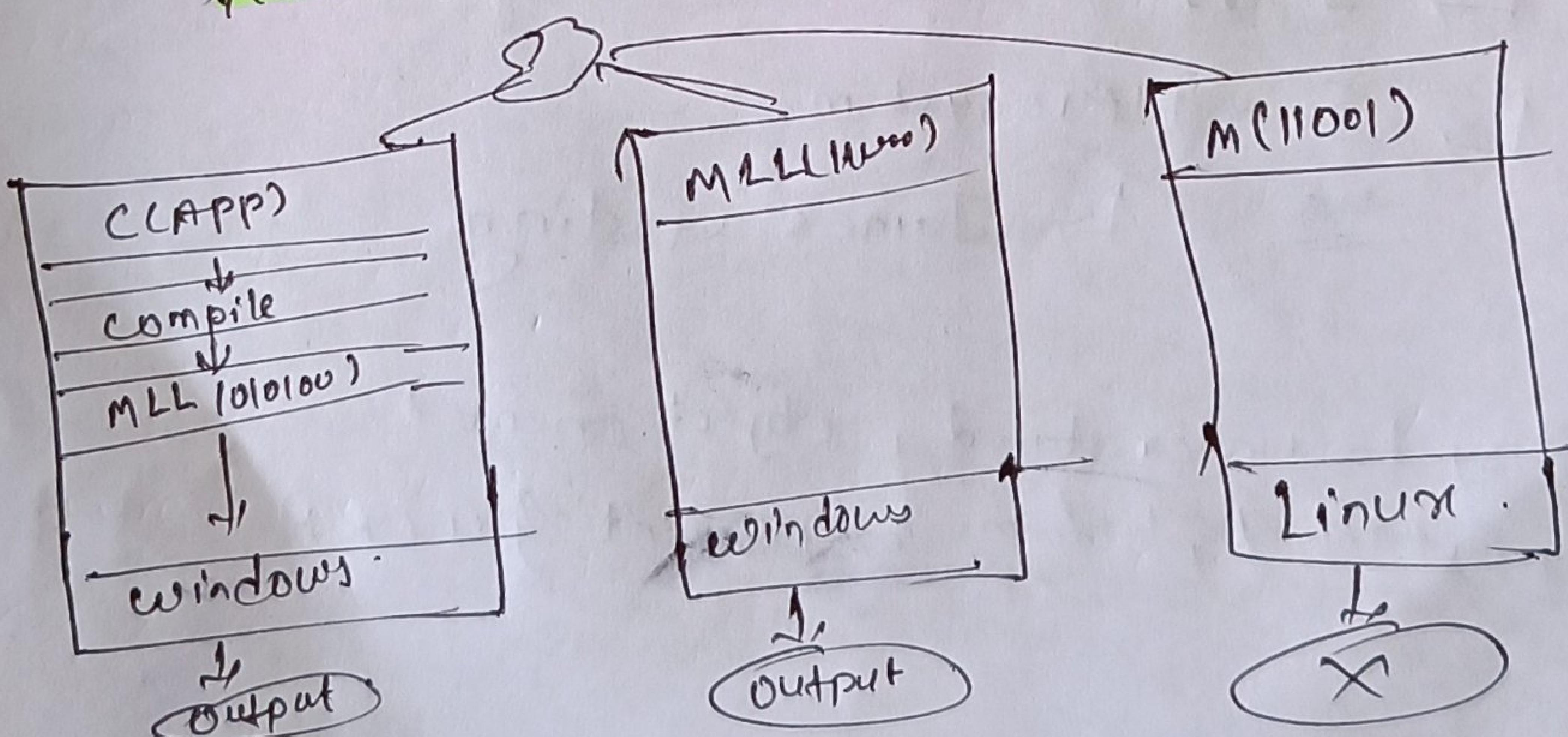
Features:

- object oriented
- platform independent or portable or word.
- Internet programming language.

platform dependency / not portable.

Note! "platform" ⇒ Microprocessor + OS
Intel + windows → platform.

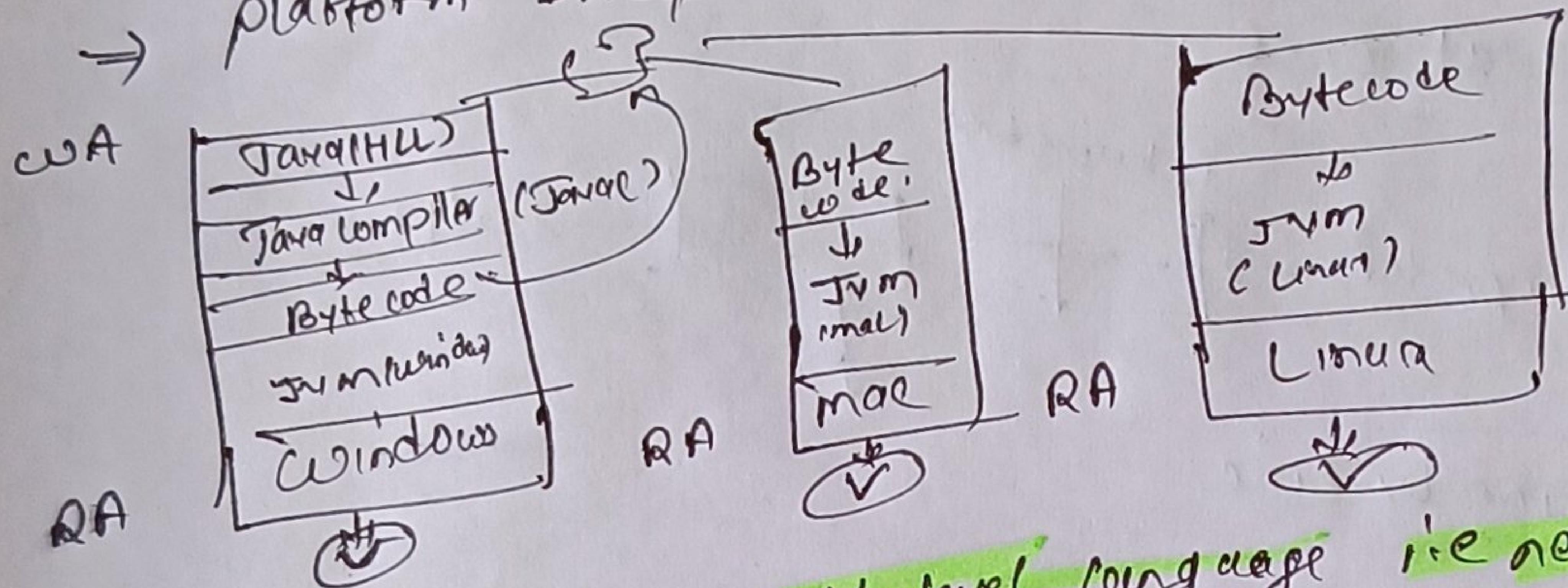
S/W → platform → OS → In software engineering term we will tell as **platform**.



Code	compilation	Execution	Result
①	windows	windows	✓
②	windows	windows	✓
③	windows	Linux	✗

⇒ Note: Any language is called platform dependent if the platform of execution and platform of compilation should be same. for e.g: C, C++ etc.

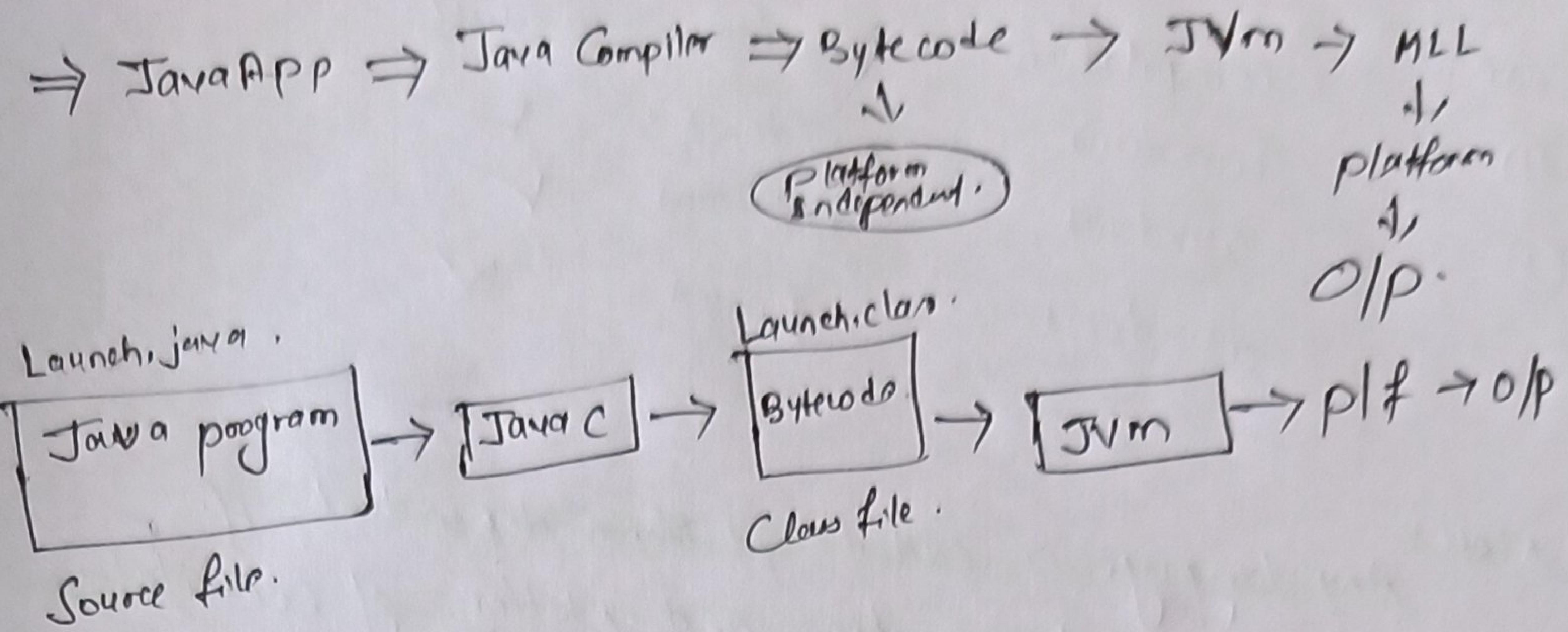
→ Platform Independent:



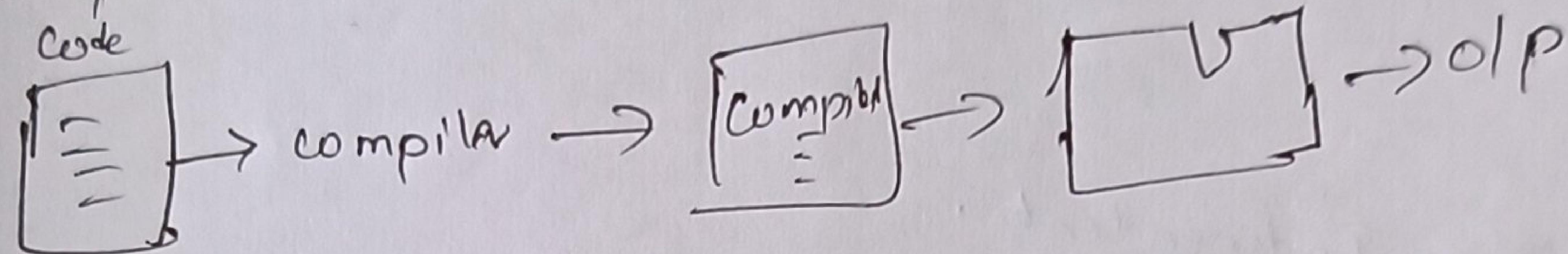
* Bytecode is a mid-level language i.e neither HLL nor machine level language.

→ when we write Java code and compile with Java compiler it will convert that code into Byte code which is platform independent. to run Byte code we will be downloading JDK for different OS and all OS have JRE convert JVM which is platform dependent.

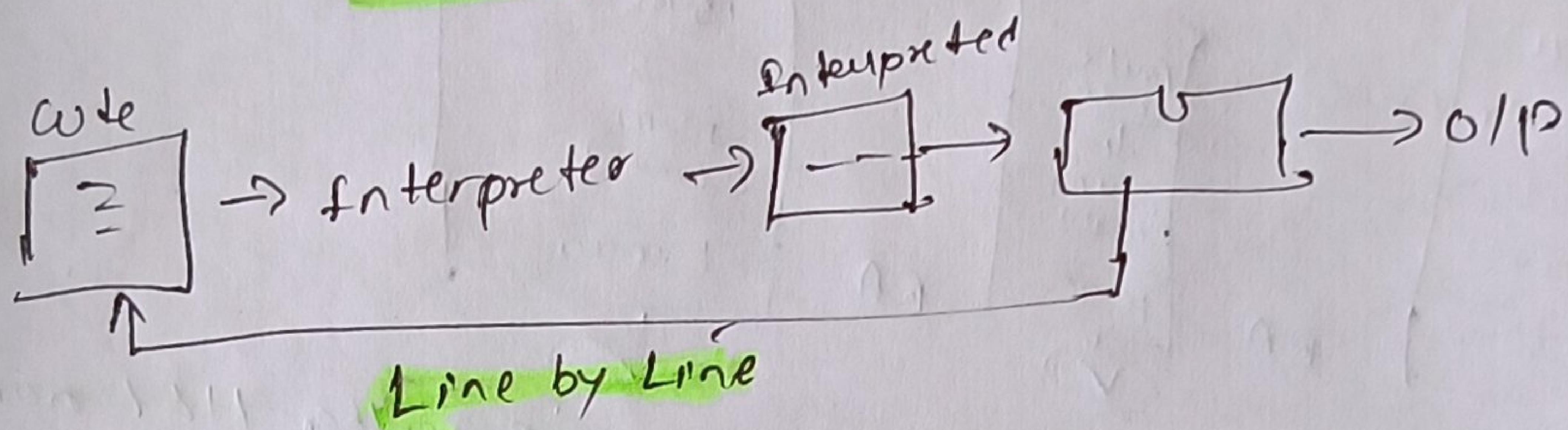
Notes: JVM converts the bytecode into machine level code internally with the help of interpreter -



Compiler VS Interpreter



all at once



Why Java is called Compiled and Interpreted language

- Because it uses both compiler and interpreter for the code.
- * Compiler at early stage to convert to byte code
- * Interpreter at execution stage to convert of MLL.

Version of Java .

→ Java version .

J2SE 1.2 → 1998

- collection framework
- swing
- JIT compiler

J2SE 1.3 → 2000

J2SE 1.4 → 2002

J2SE 5.0 → 2004

- Annotations
- Autoboeing
- Enumeration
- Enhanced for loop

Java 8 6 → 2006

Java 8 7 → 2011

- string in switch
- try with resources
- <> diamond operators

Java 8 8 → 2014

- lambda Expression
- support for T8 code
- Date and Time API
- Stream API

Java 9 - 2017

- modularity .
- Reactive Streams .
- JShell .

Java 10 - 2018

- local variable Type Inference .

Java 11 -

- Run source file .
- var for lambda .

Java 12 -

Java 13 -

- switch Expression .
- multiline strings .

Java 14 -

- Records .
- packaging tool .

Java 15

J6

Java 16 -

- sealed classes .

Java 17 -

Java 18 -

- virtual threads .
- Vector API

which version we should use .

LTS versions (Long Term Support)

→ Java 7

→ Java 8

→ Java 11

→ Java 17