CS & IT ENGINEERING

C Programming

Data Types and Operators

Lec- 02

By- Pankaj Sharma sir

Introduction to C Programming-2

Why PL ?

Translator ?

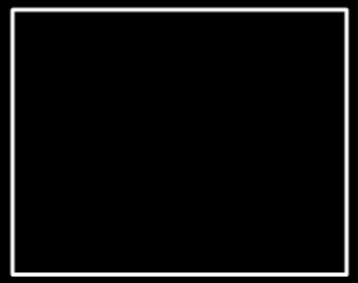
Abstraction/Interface → general

→ recording

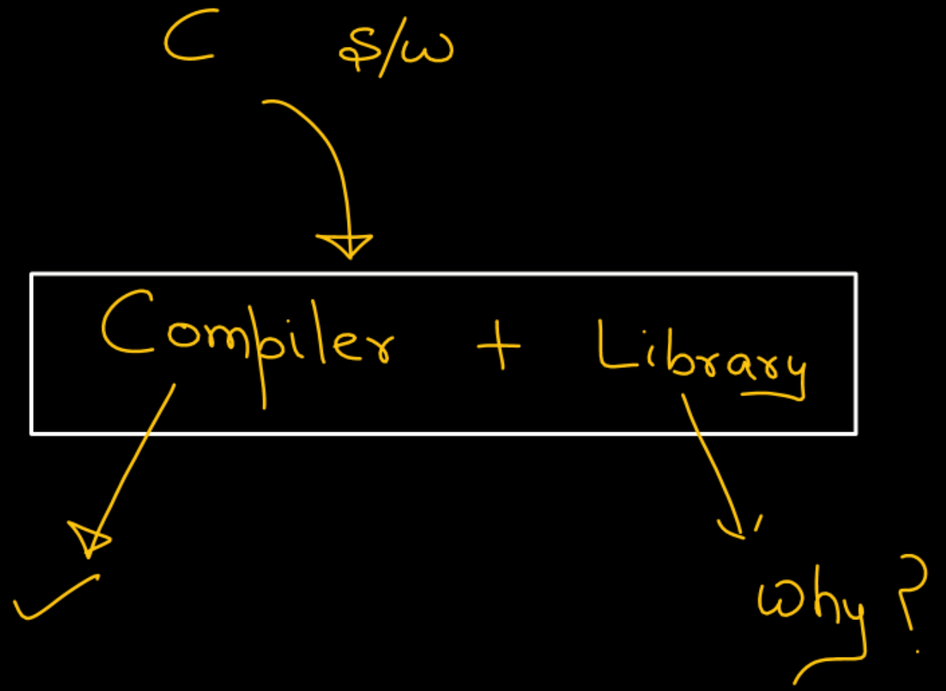
Presentation
Slide
→ powerpoint

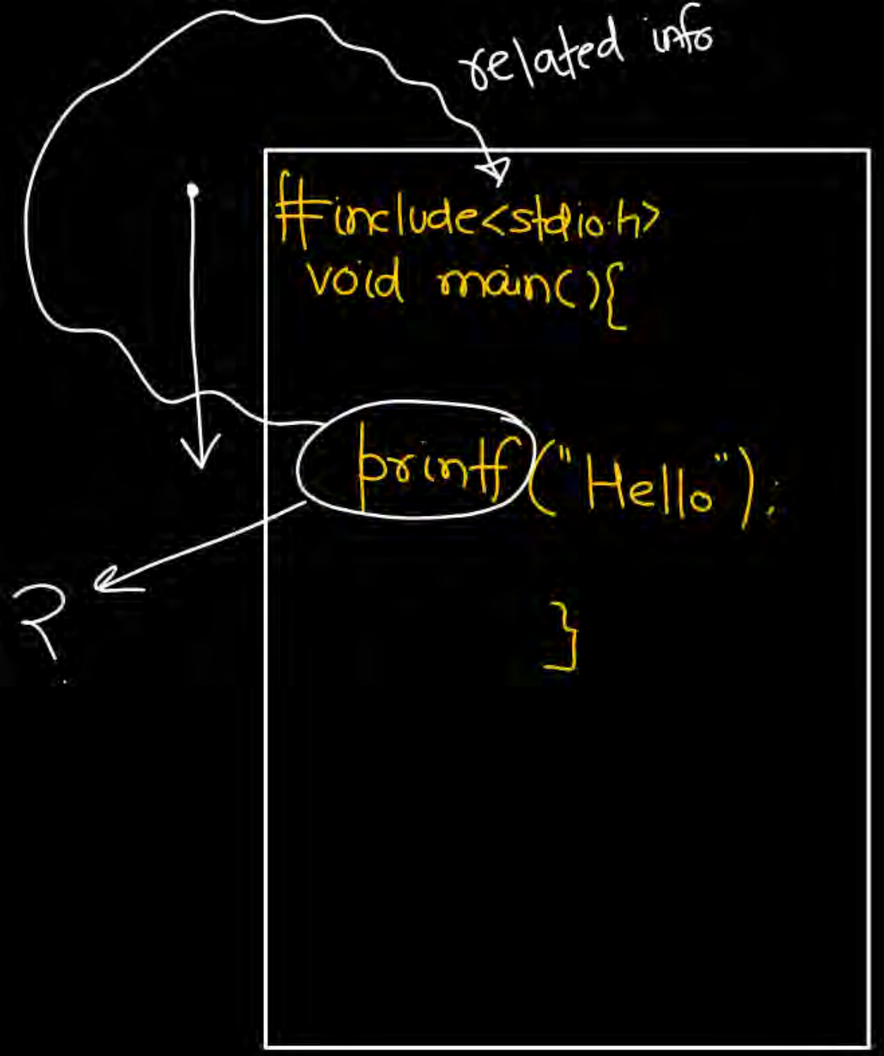
document
.doc
→ MS- word

Text file
.txt
→ NotePad

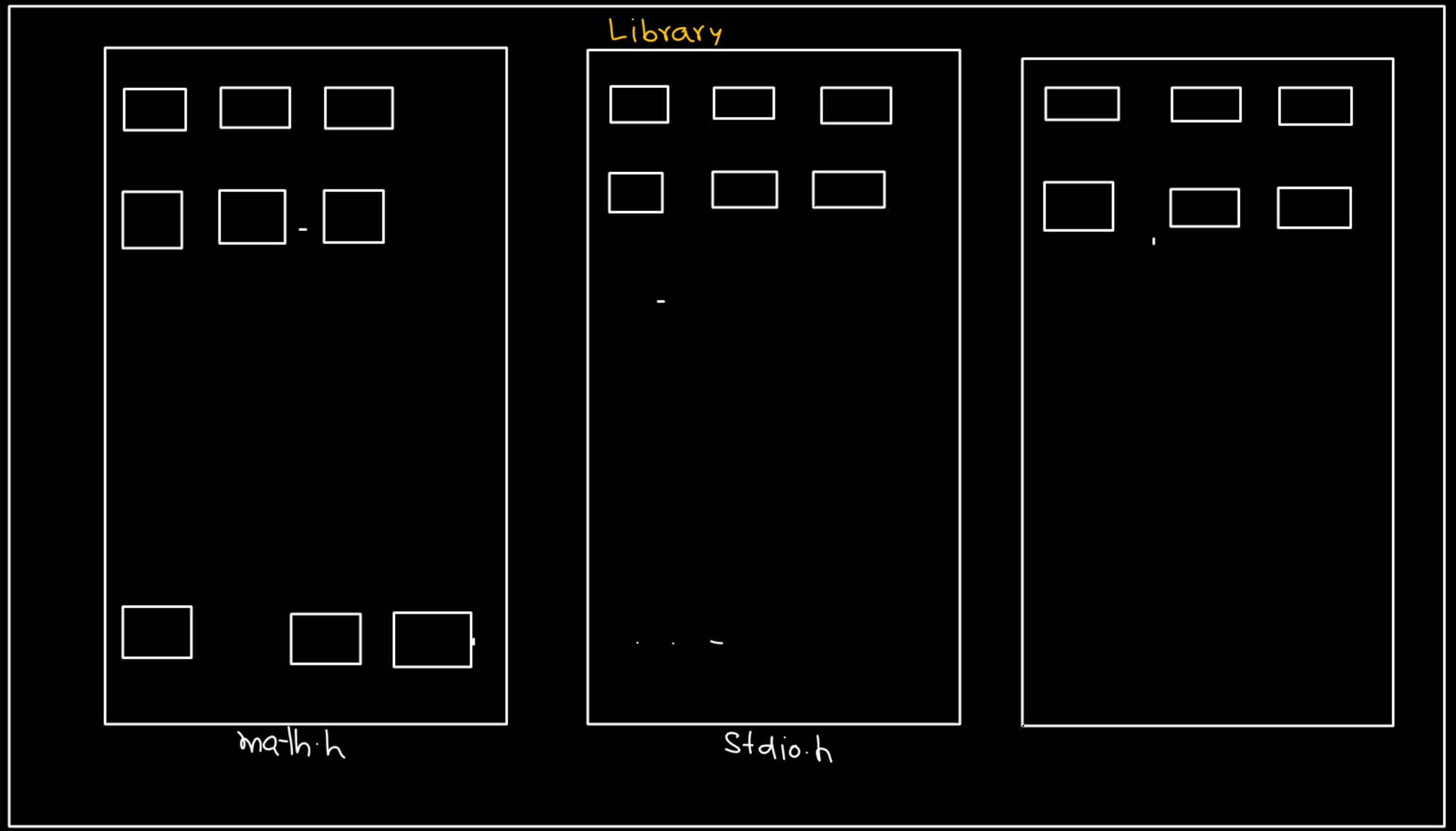
C Program
.c
→ 

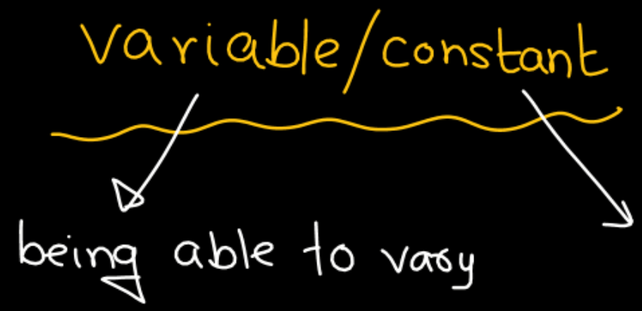
Vs-code
code blocks
Turbo C


C
s/w
Compiler + Library
why ?



Task:
Whatever u provide inside
" "
⇒ print on screen


Library
math.h
Stdio.h



email id :

Password :

$$y = 4x$$

$$\left\{ \begin{array}{ll} x = 1 & \text{then } y = 4 \\ x = 2 & y = 8 \\ x = 3 & y = 12 \end{array} \right\}$$

Container

Tea

Rice

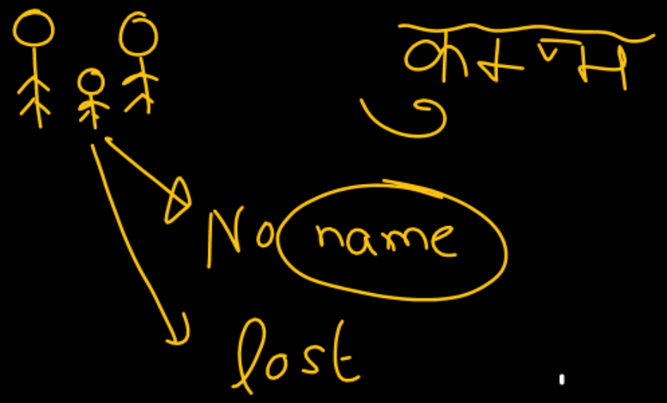
Sugar

Container type

goods

20 Randomly

retrieve :



1000

2016

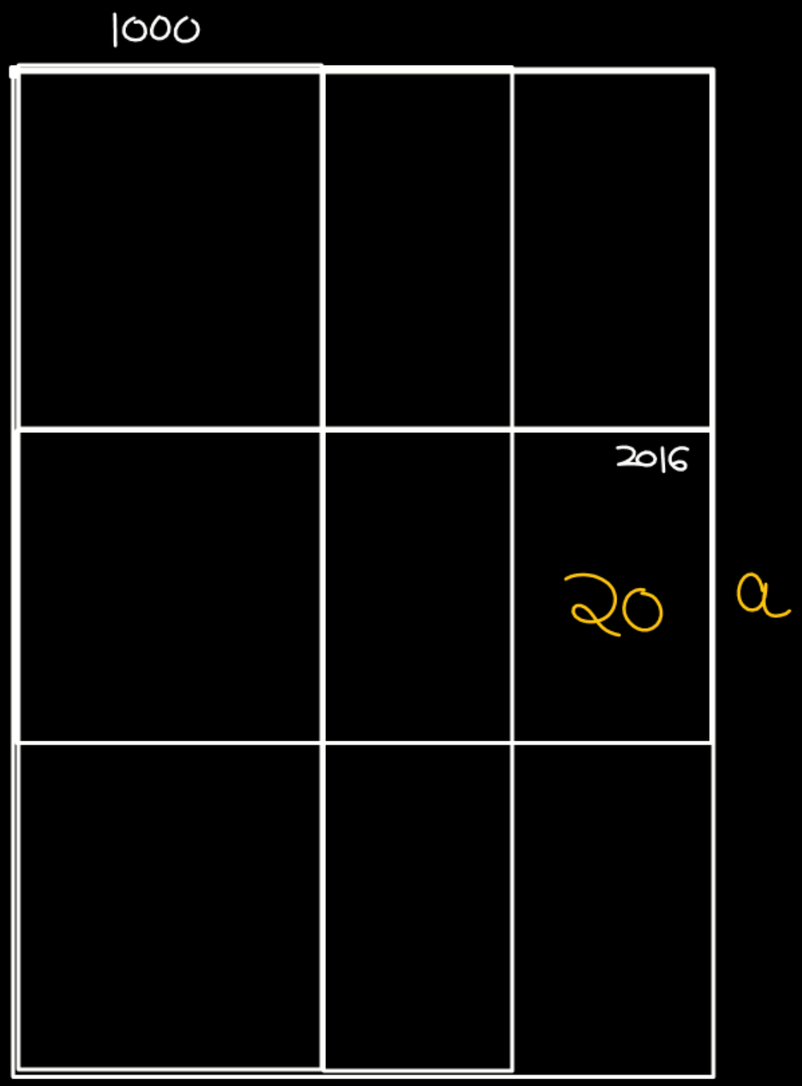
20

20 Randomly

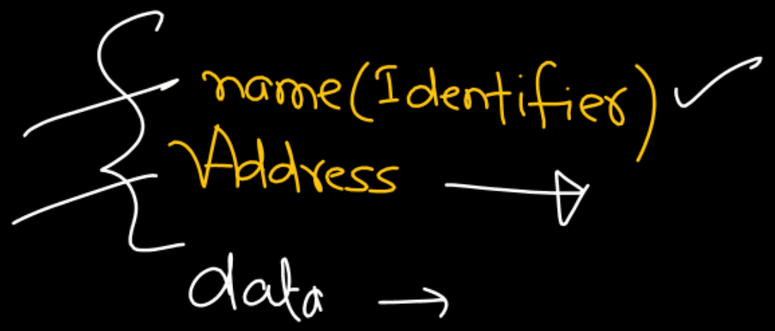
retrieve :

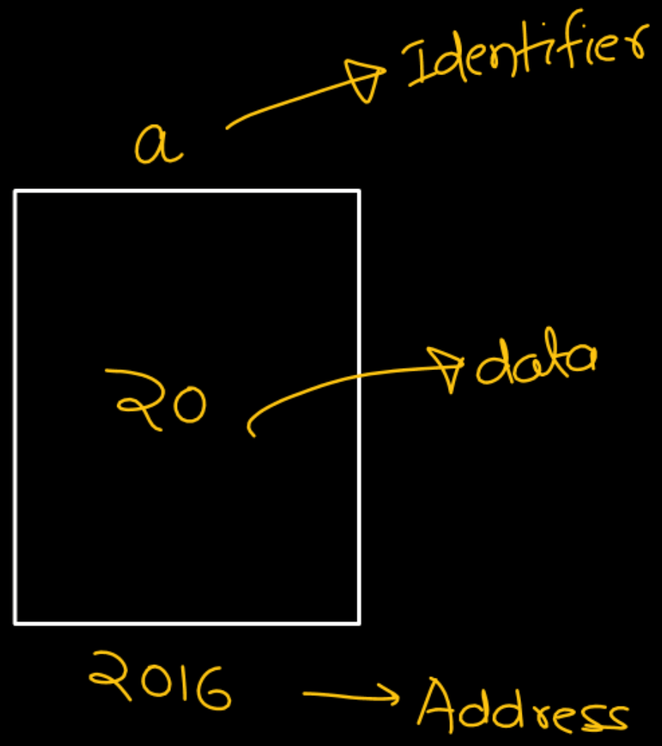
name

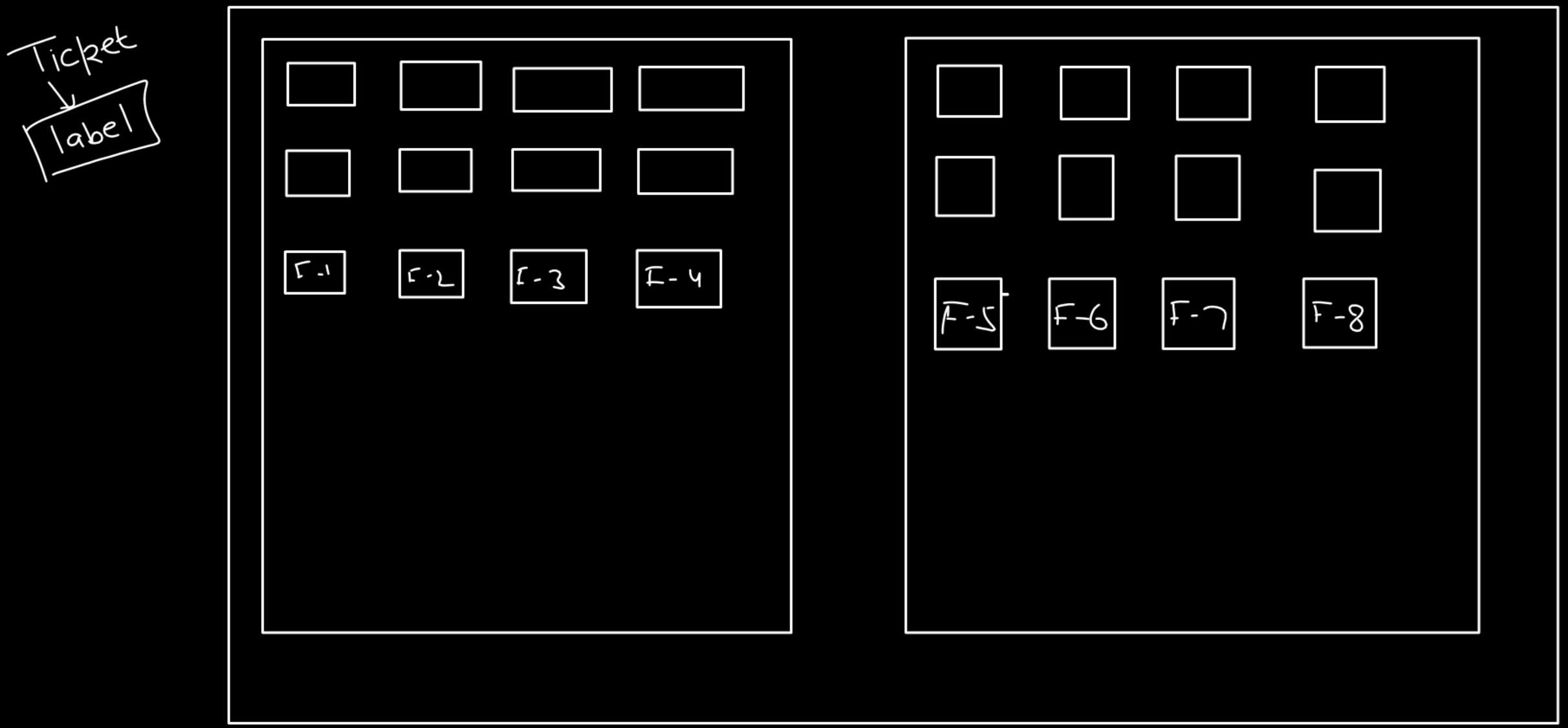
$a = 20$

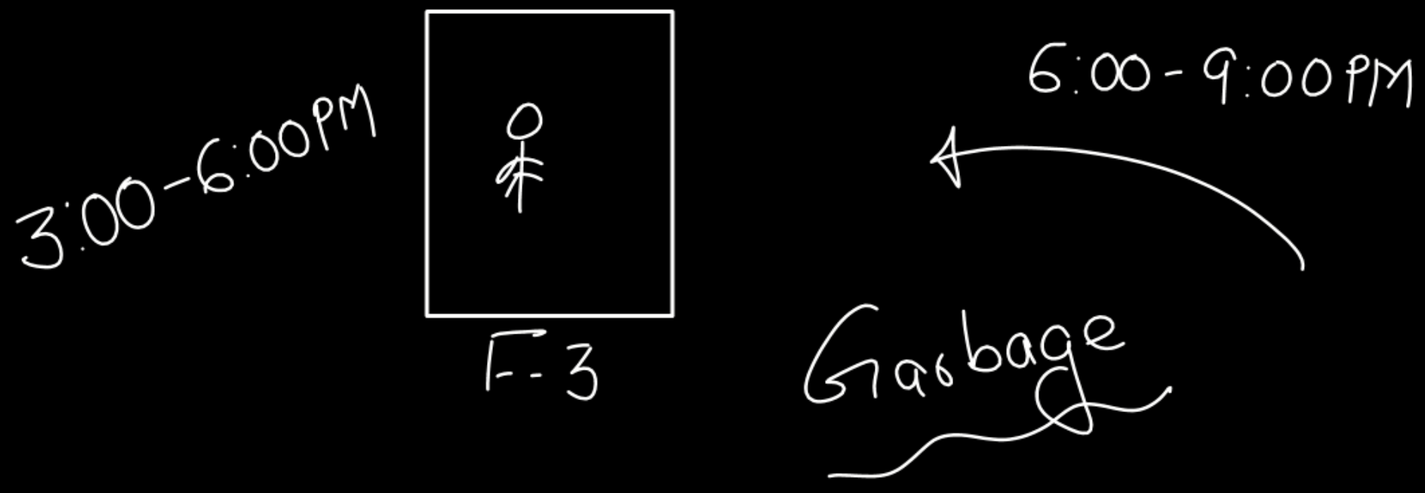


a : 20


name(Identifier)
Address
data


a
Identifier
20
data
2016
Address


Ticket
label
F-1
F-2
F-3
F-4
F-5
F-6
F-7
F-8


3:00-6:00PM
F-3
6:00-9:00PM
Garbage

Data

Google maps: source [1234X]

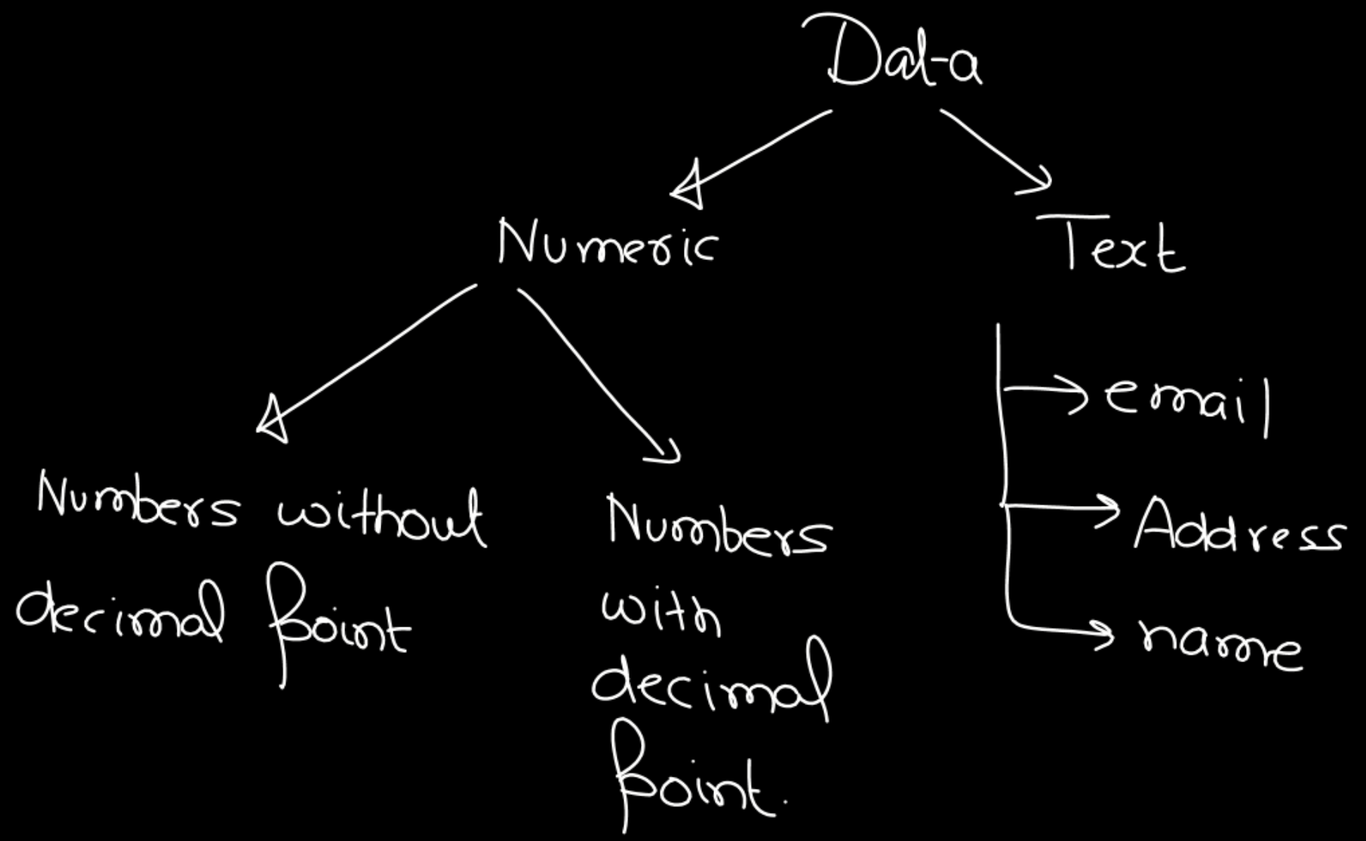
dest. [Goa] ✓ Text

ATM M/C

Pin [abcd] X

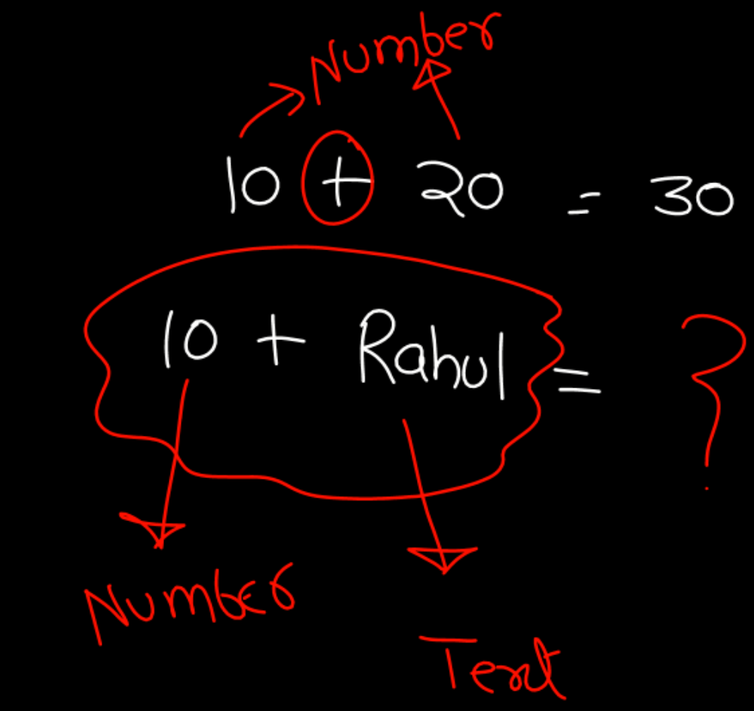
1234 ✓ Numeric

data types


Data
Numeric
Text
Numbers without decimal point
Numbers with decimal point.
email
Address
name

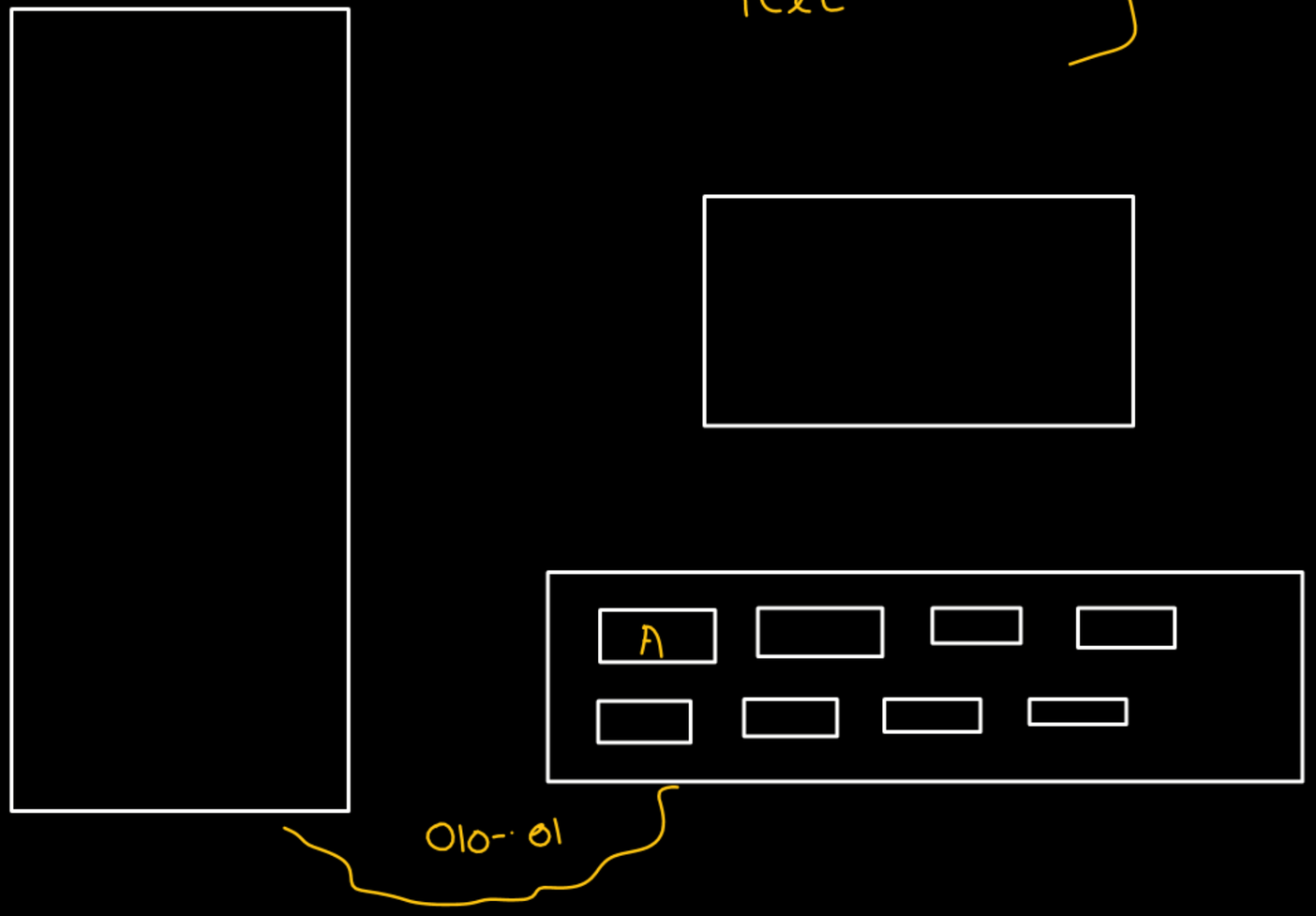
Number → +112, 98, -60, 0

Number → 9.8, -12.4, 3.14



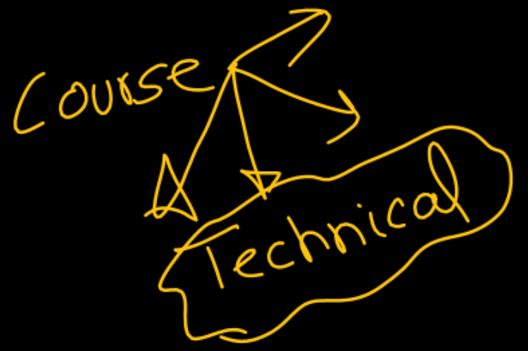
Text : Pankaj (Symbols)

Integers
Floating point
Text



whatever i/p u provide from KB

it is a seq of 0s & 1s



KB

Program

A

01000001

65

binary Number

Container-Type

Number without decimal point

int

a = 20 X

a = 20 ;

float b = 12.38;

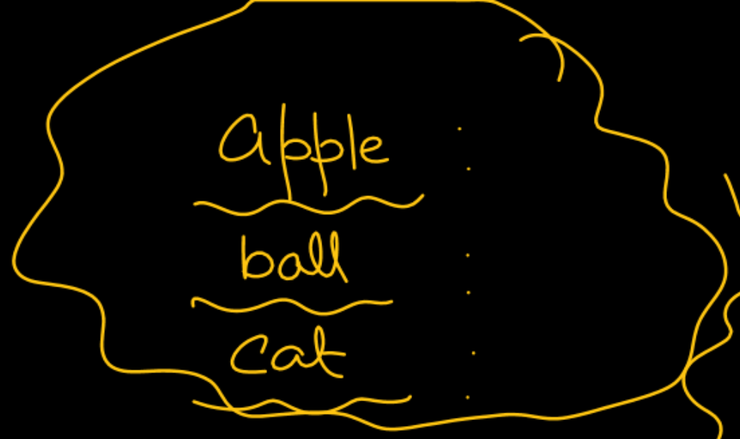
Numbers with a decimal point

Text : Pankaj

```
char c = '@';
```

int

float

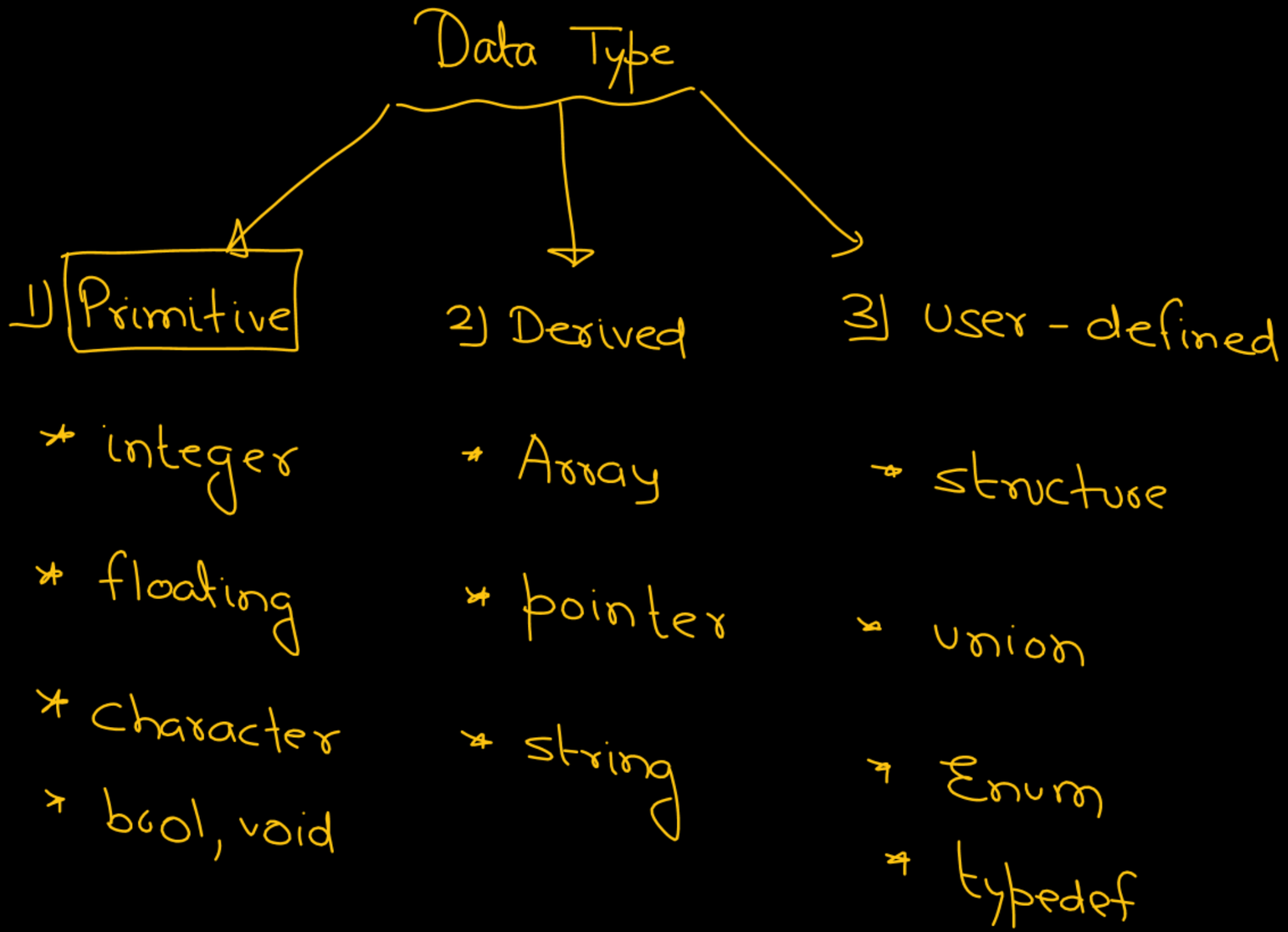


To store Symbol

Lacks predefined to human being

Pre-defined → meaning Compiler

Keywords


Data Type
1) Primitive
* integer
* floating
* character
* bool, void
2) Derived
* Array
* pointer
* string
3) User - defined
* structure
* union
* Enum
* typedef

Integer

-90, +80, 80, 90, 0, 12

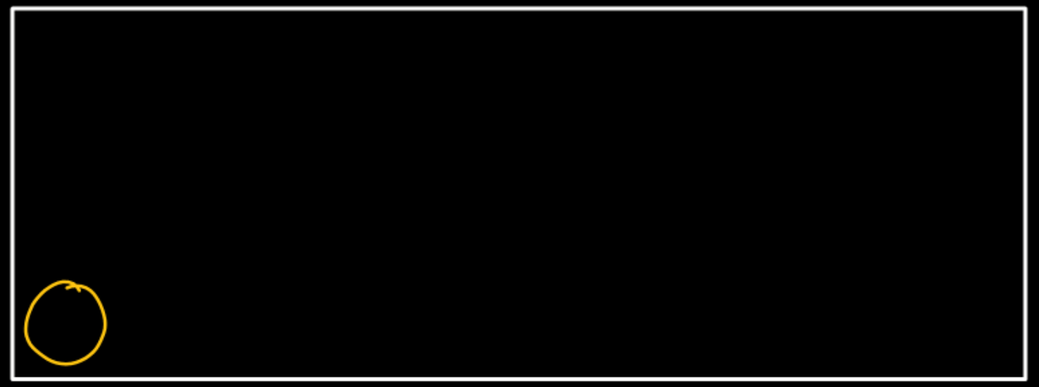
+ve, -ve

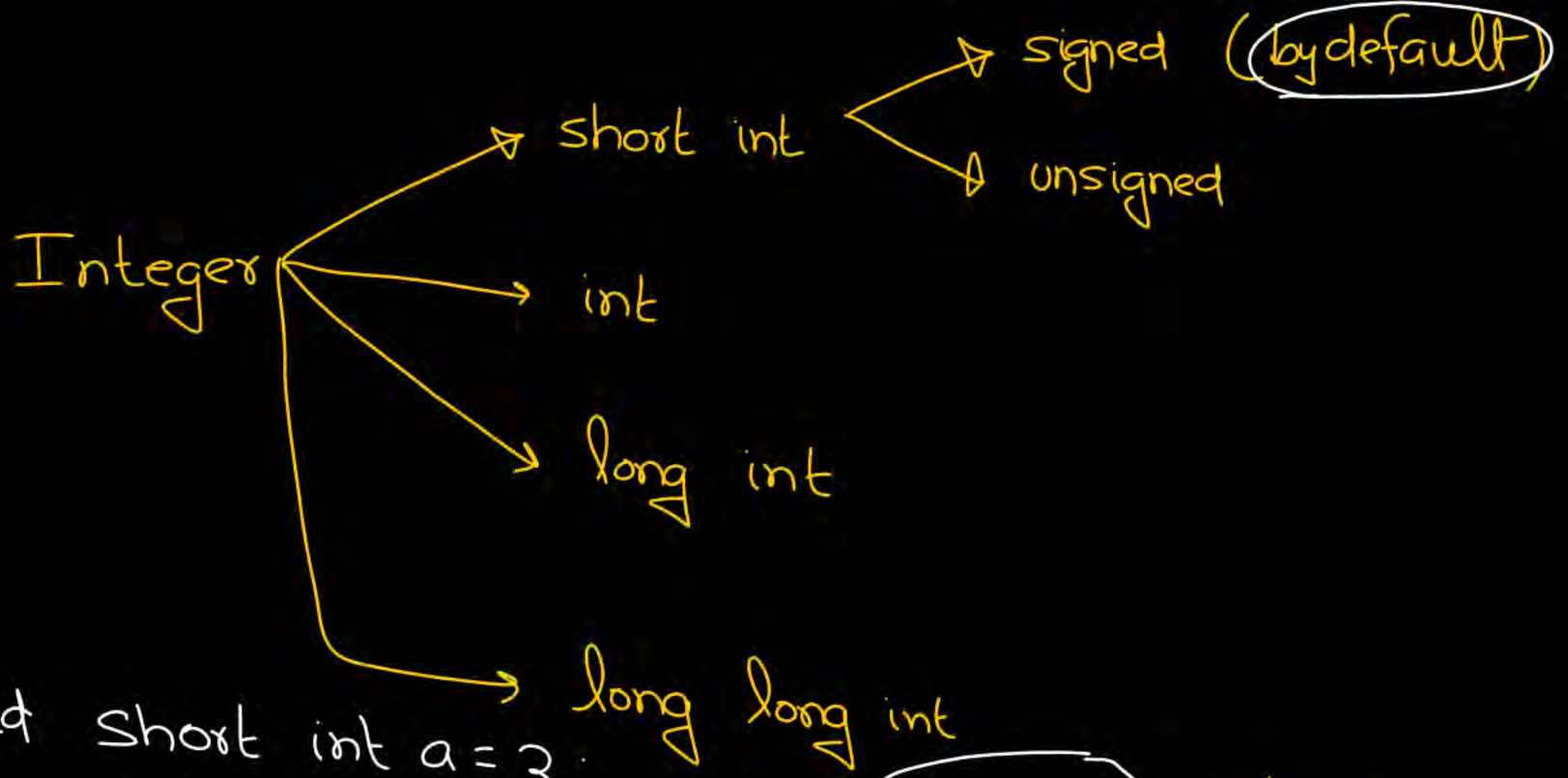
short int a = 3

int b = 4139;

long int = 23967839 21;

long long int = 212;;

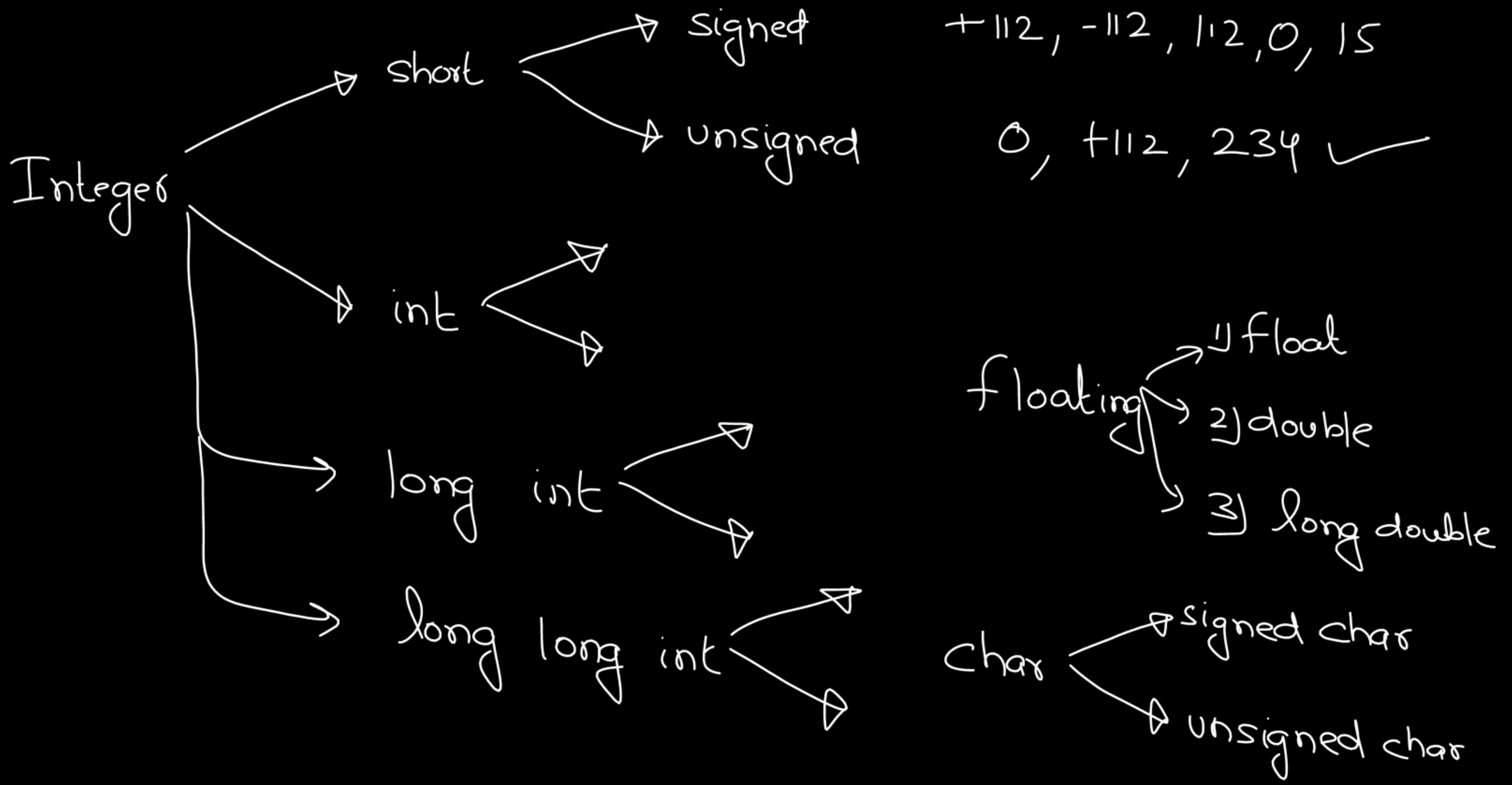




Same { signed short int a = 2;
OR
short int a = 2;

short int a = 2;

by default signed


Integer
short
signed
unsigned
+112, -112, 112, 0, 15
0, +112, 234
int
long int
long long int
floating
1) float
2) double
3) long double
char
signed char
unsigned char

decimal number system (0-9)

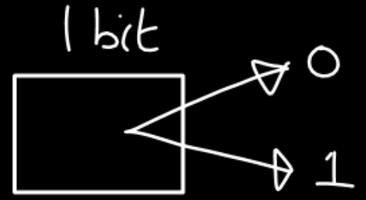
No. of ways $\Rightarrow$ 10×10

$= 100$ choices

00, 01, - - - 99

binary number System (0,1)

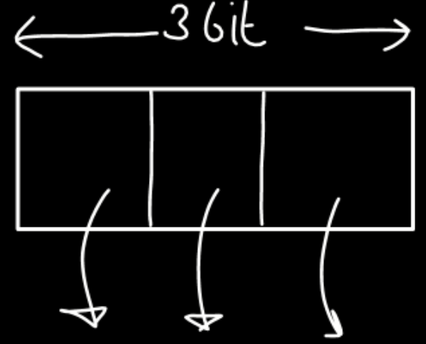
1 digit = bit



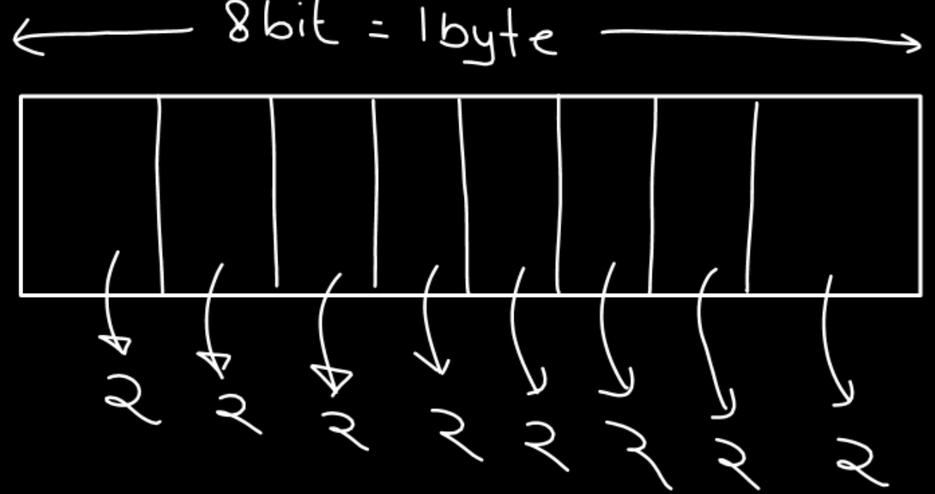
2 possibility

2×2

$= 2 \times 2 = 2^2 = 4$ possibility



$2 \quad 2 \quad 2 = 2^3$ possible values



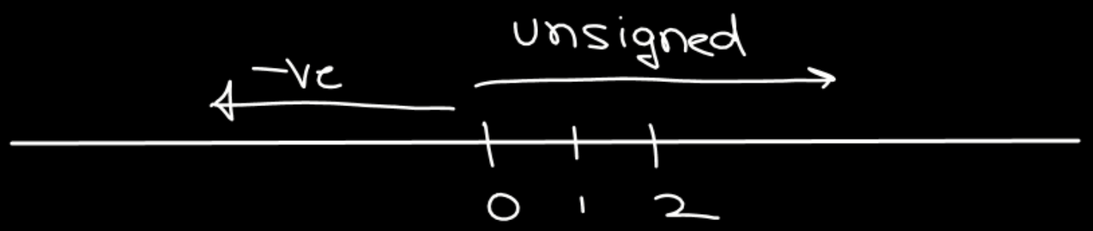
$= 2^8$ possible values

Unsigned

2bit

= 4 possible values
(0, 1, 2, 3)

0 0
0 1
1 0
1 1



unsigned

$\Rightarrow 2^4$ (16 possible)

0, 1, 2, - - - 15

C standards 1 byte = 8 bits

sizeX

Min size

short int 2 byte

assuming 2 byte

short int → unsigned
short int → signed

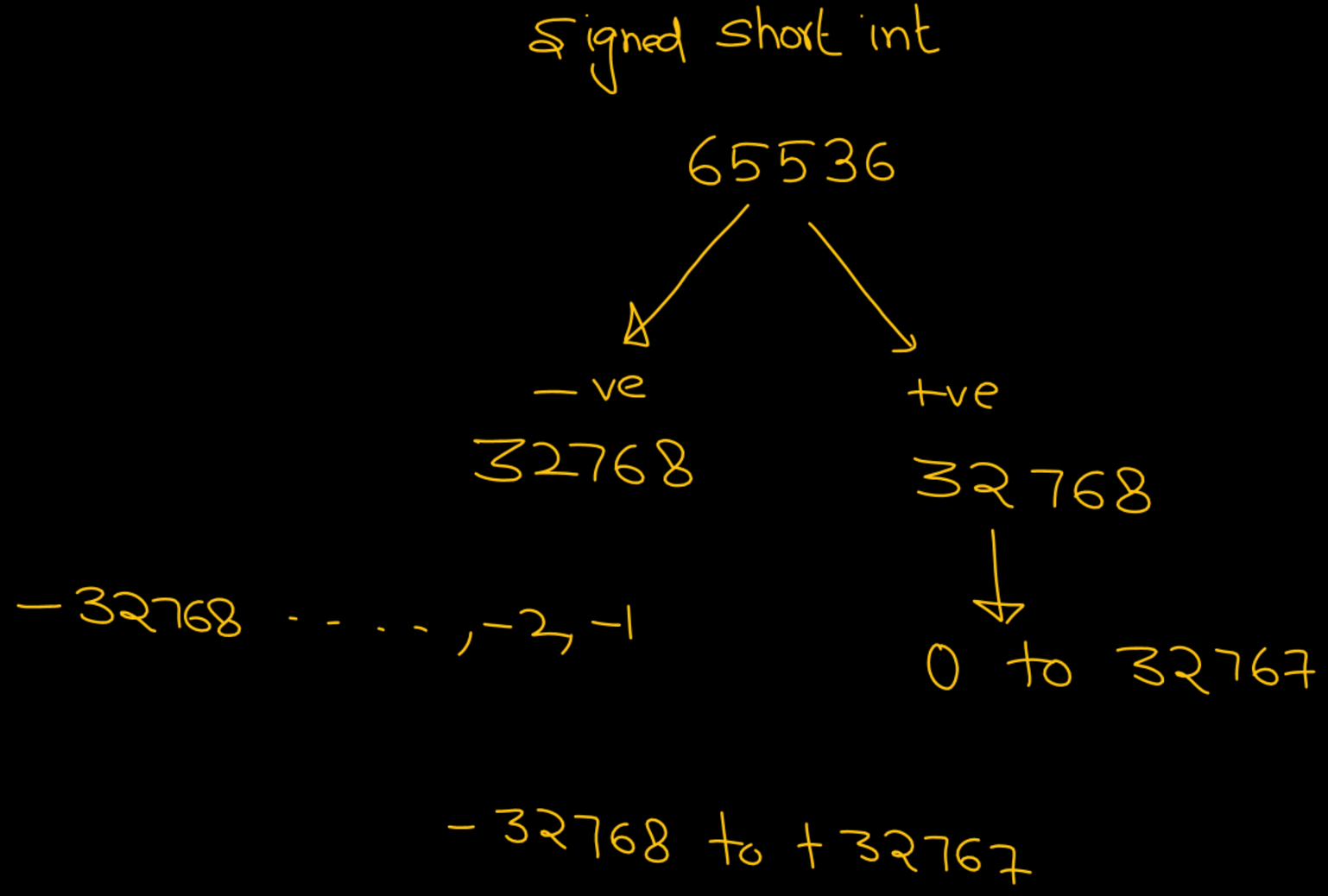
2 byte = 2 × 8 = 16 bits

unsigned

⇒ 2^{16} possible values

65536 values

Unsigned short int ⇒ (0 to 65535)


signed short int
65536
-ve
32768
+ve
32768
-32768 ..... ,-2,-1
0 to 32767
-32768 to +32767

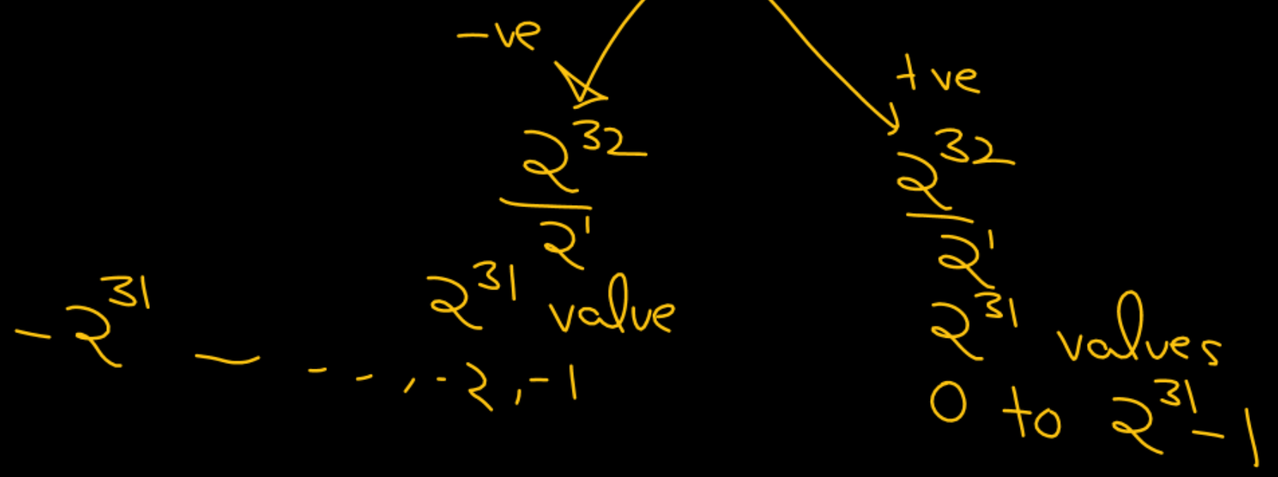
int ⇒ 4 byte

4 × 8 bits = 32 bits

2^{32} possible values

unsigned int : 0 to $2^{32}-1$

signed int : 2^{32} values ⇒ -2^{31} to $+2^{31}-1$



integer

3-4

n bits

2^n values

Unsigned

0 to $2^n - 1$

Signed

2^n

$\frac{1}{2}|2^n$ $\frac{1}{2}|2^n$

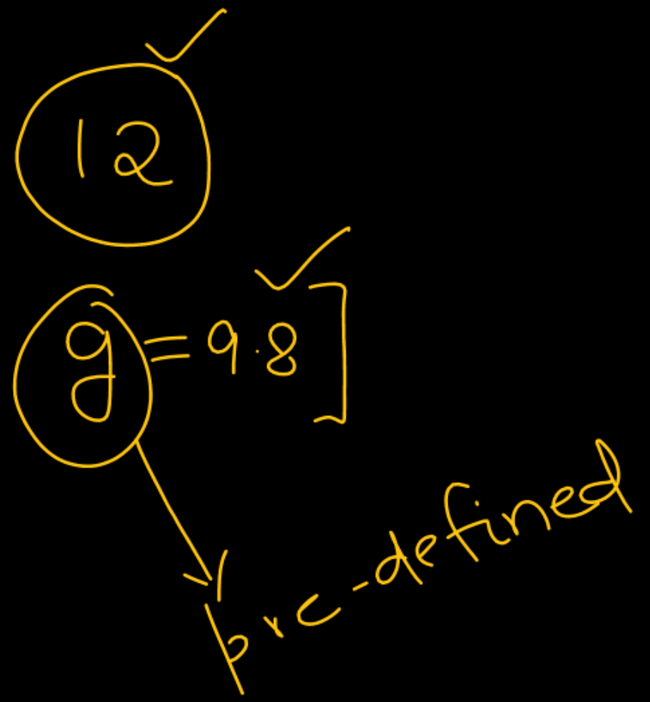
-2^{n-1} . . . , $-2, -1$ $\quad$ 0 2^{n-1} 1

0 to $2^{n-1} - 1$

$\Rightarrow$

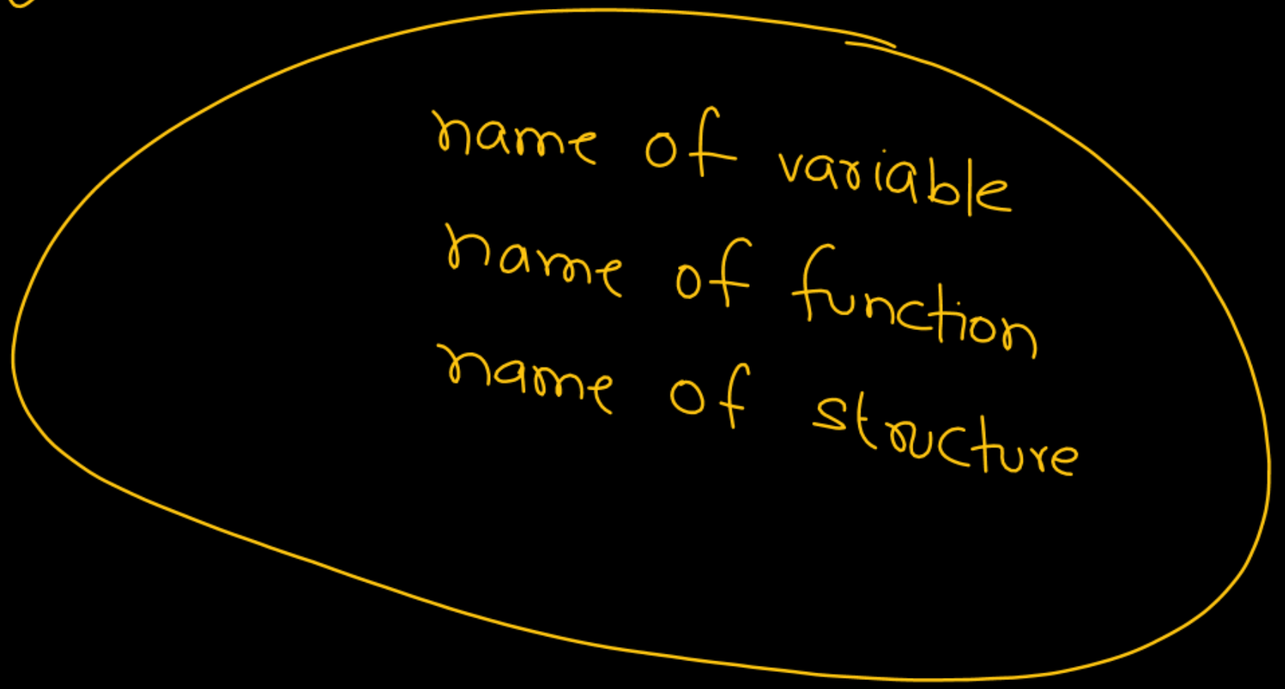
-2^{n-1} to $+2^{n-1} - 1$

10 min → doubt ?



C Standard ⟸ undefined behaviour

identifier



26 lecture

↓

26 question

String

Overflow in addition — Digital

Gate ✗✗

↓

practical lab

C → DS


THANK YOU GW
SOLDIERS !