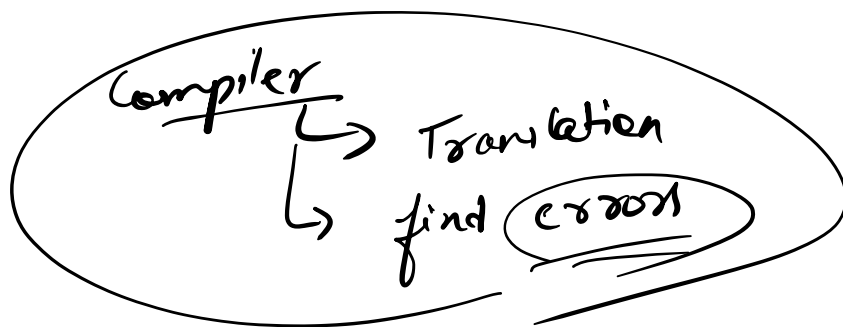
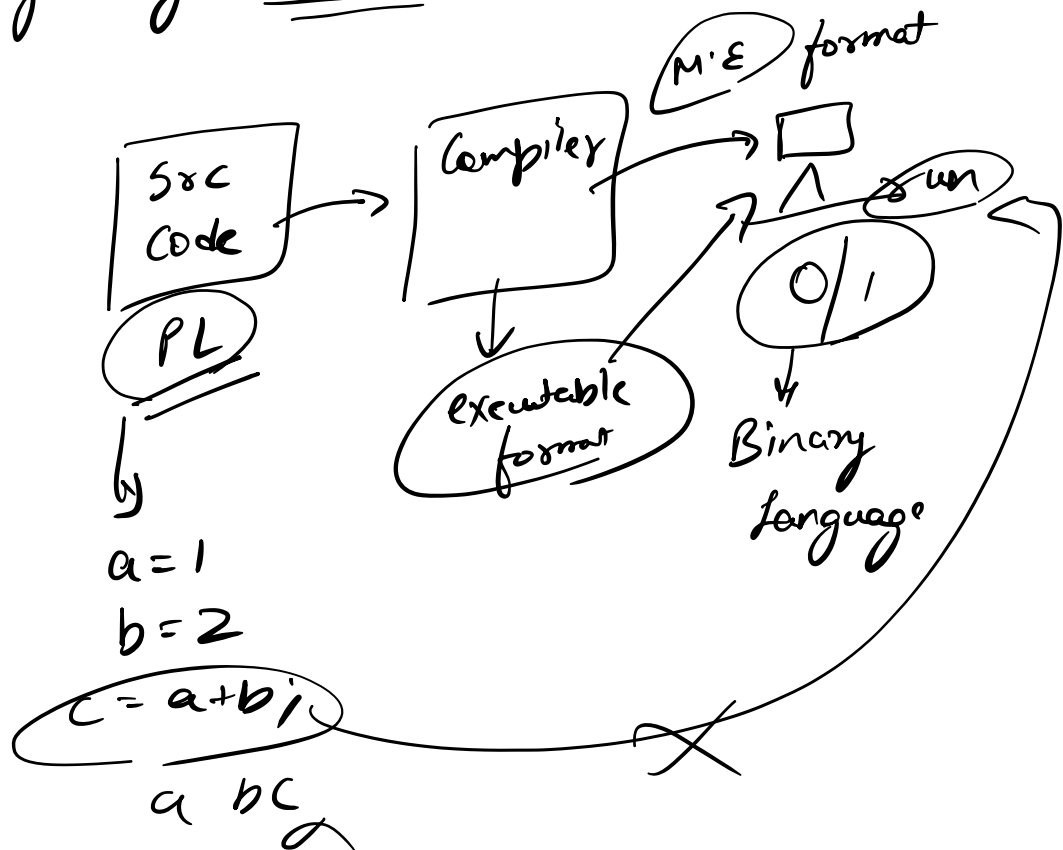


First Program

→ getting started

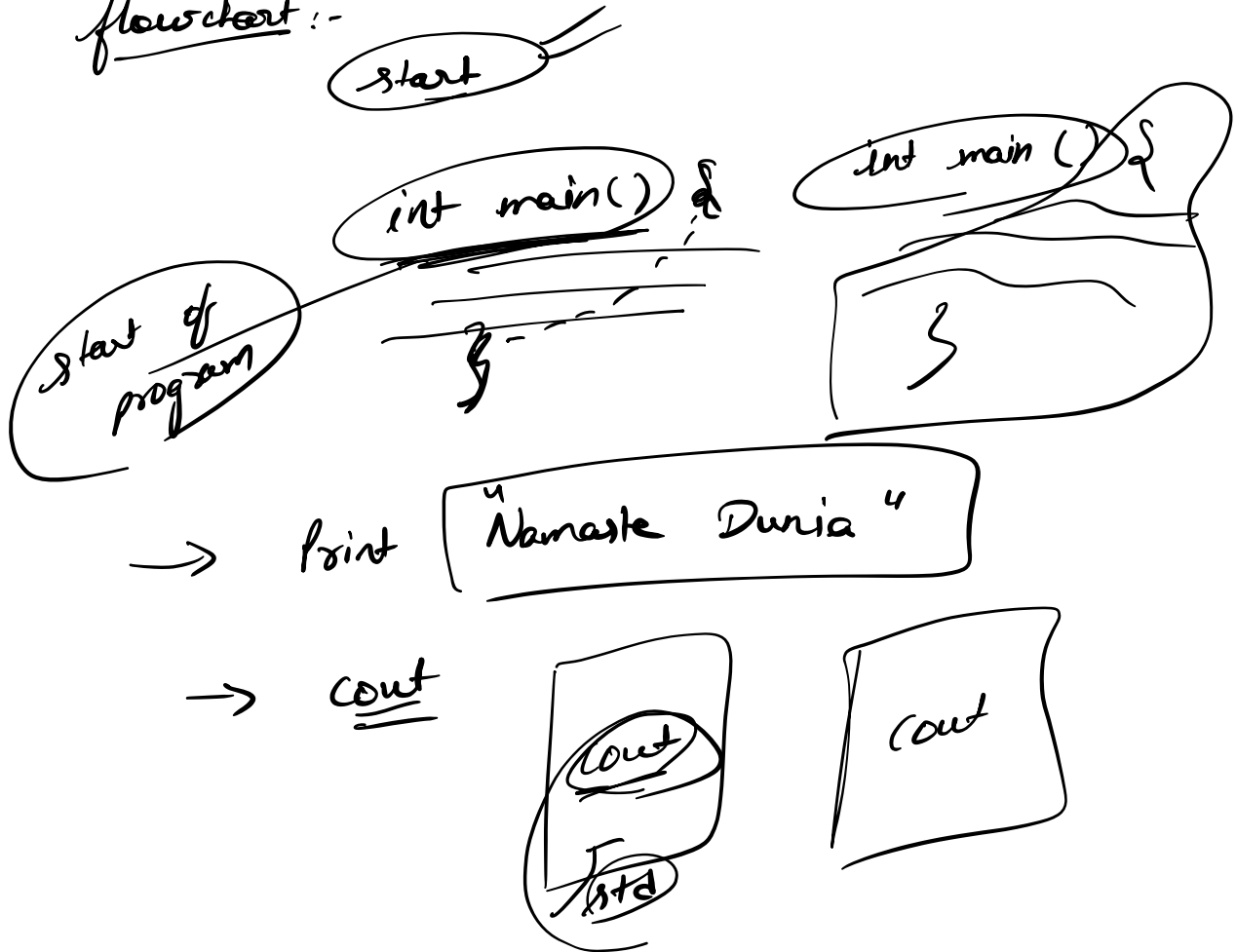


IDE → Code Blocks
VS Code → Youtube

Replit → C++ file → run

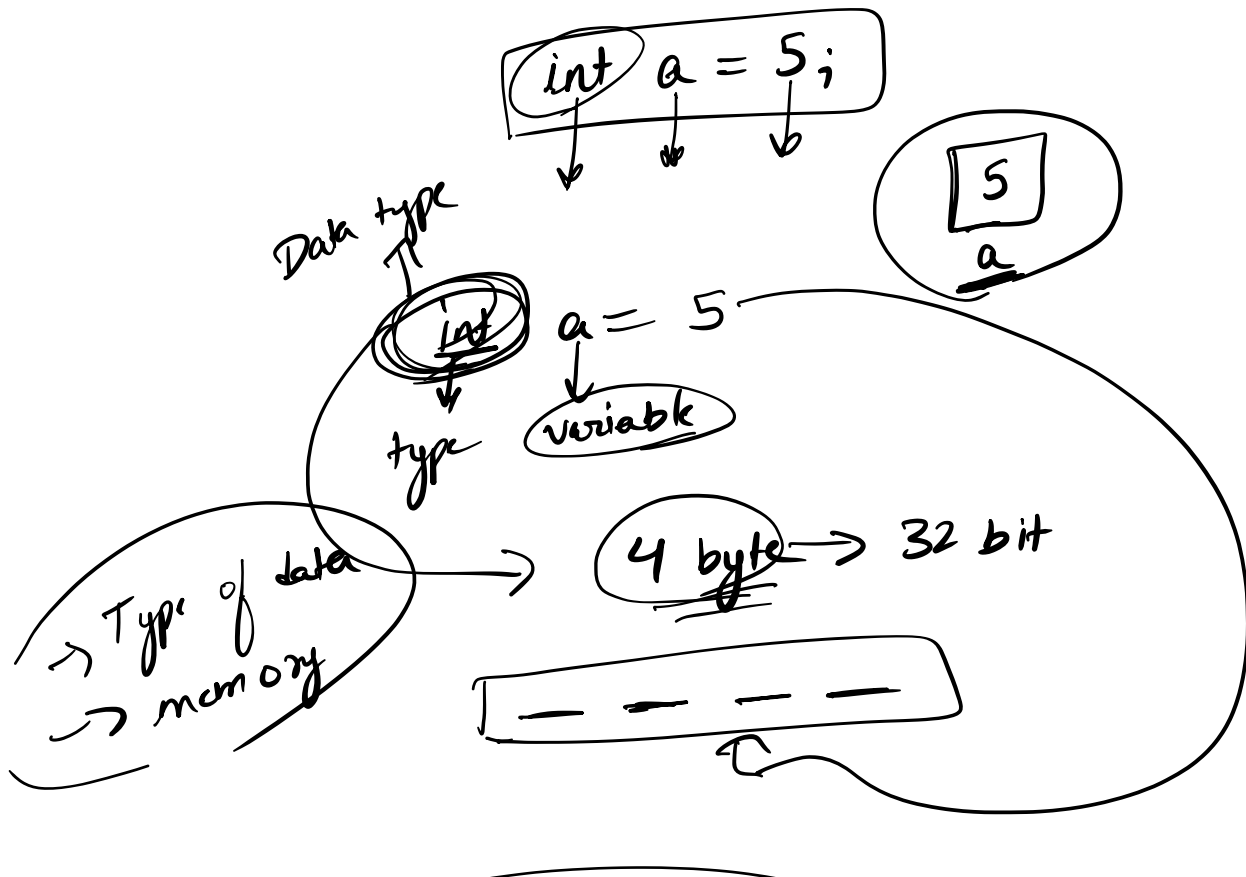
Namaste Dunia
Hello World X

flowchart :-

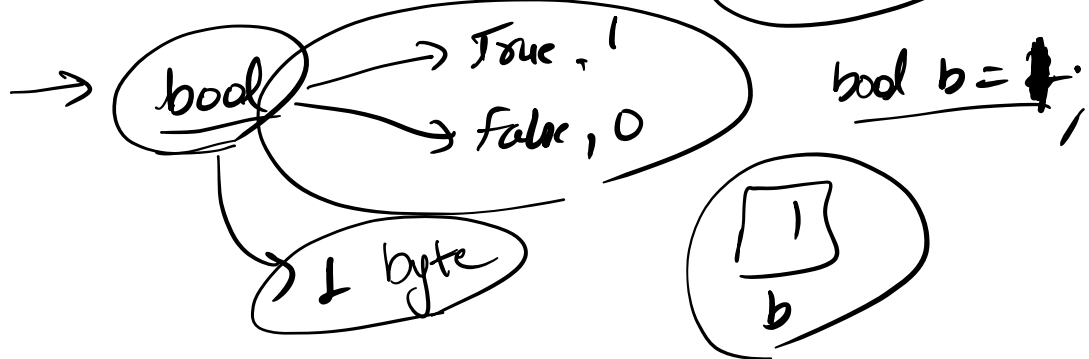
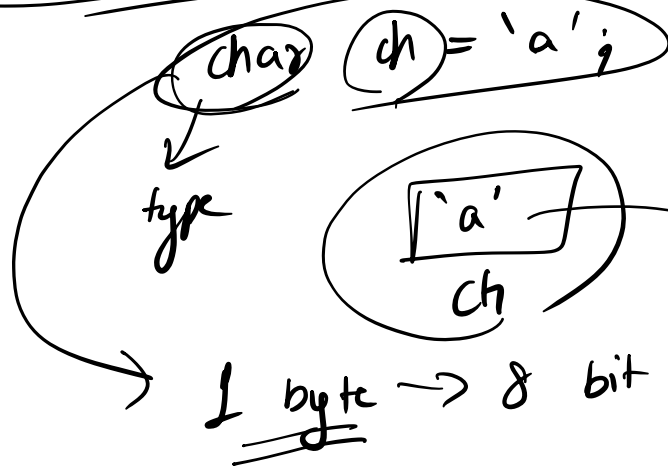


- << ? <<
cout << → to write
cout >> → compilation error
- endl → new line or enter
- \n → enter
- ; → to end line

Data types & Variables



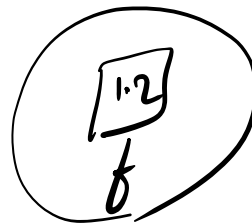
→ char ch = 'a';



→ float f = 1.2;

↓

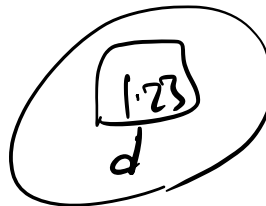
4 byte



→ double d = 1.23;

↓

8 byte



Variable name

abc
ABC
A1
A-1

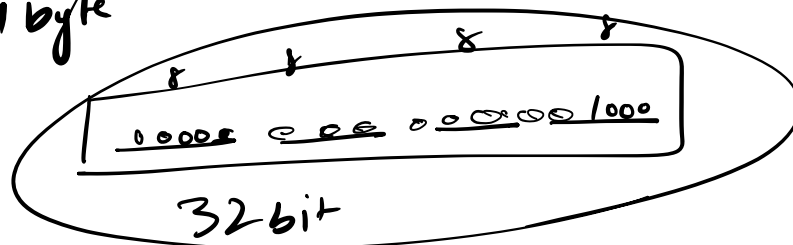
1abc ✗

abc1 ==

-a1 ==

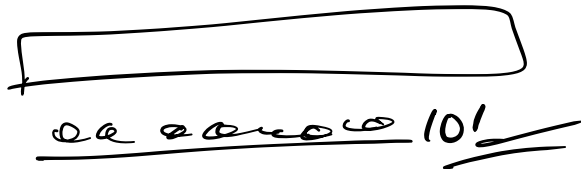
How data is stored?

int a = 8; → Binary 1000
4 byte



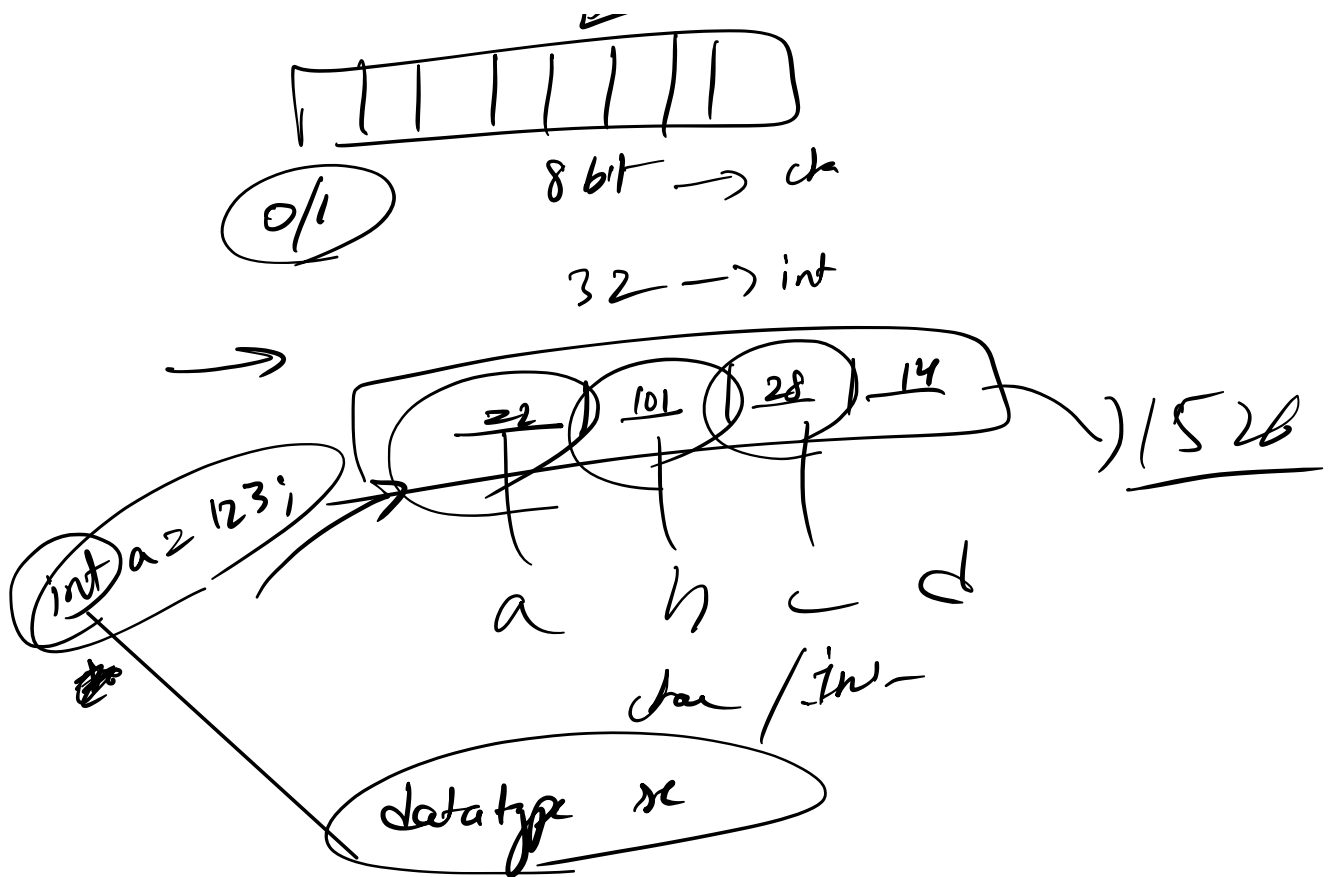
int a = 5; → 101
3 bit

32 bit



-ve number → ?

→ char ch = 'a';
1 byte
97
ASCII table → H/W
Binary



\rightarrow Type casting

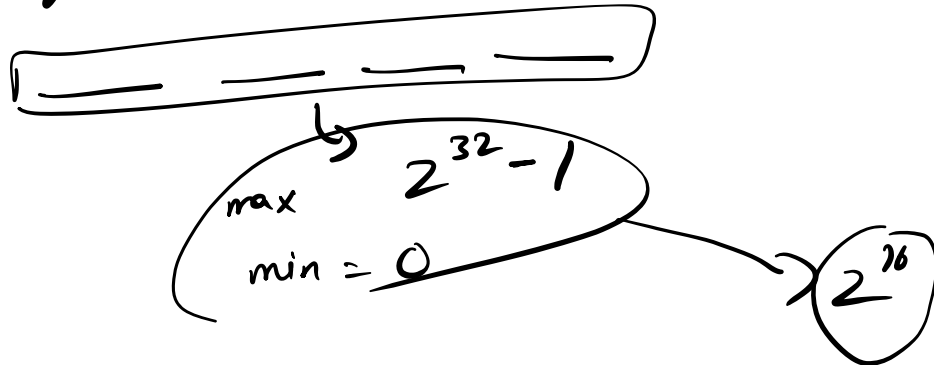
a \rightarrow 97

int a = 'a';

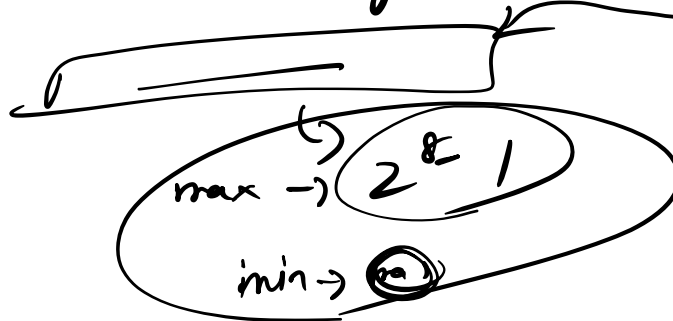
cout << a << endl; \rightarrow ?

char ch = 98

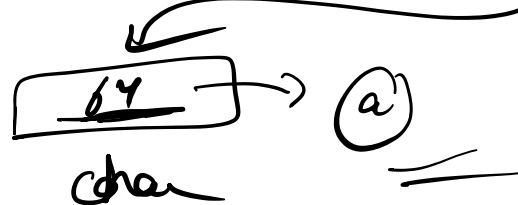
Integer \rightarrow 4 byte \rightarrow 32 bit



char ch \rightarrow 1 byte \rightarrow 8 bit



123456
int



How -ve no are stored

first bit

↳ +ve → 0
↳ -ve → 1

-ve no → (-5)

display

2's complement

↳ ignore the -ve sign
(5)

↳ convert into binary rep.

00000000 101

↳ take 2's complement
store

000 000 00 0000 000 101

1's compl. → 111 111 111 111 010
+1

2's comp
1 111 111 111 011
-ve
int a = -5

print a = 1

1st step $\frac{0022222222222222}{11}$

2^5 ops → | 01
→ 5

```
print(-5)
```

Compilation

→ Namaste Dunia

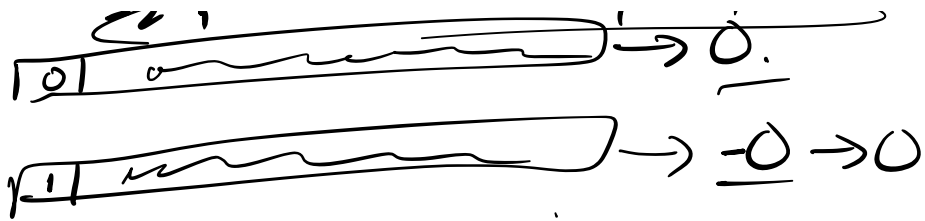
↳ Data types

↳ Variablen

$2^{31} - 1$

$-(2^{31}-1)$

~~$$-(2^{3^1}-1), \dots, 2^{3^1}-1$$~~



1 no → "0" → 2 representations

→ default

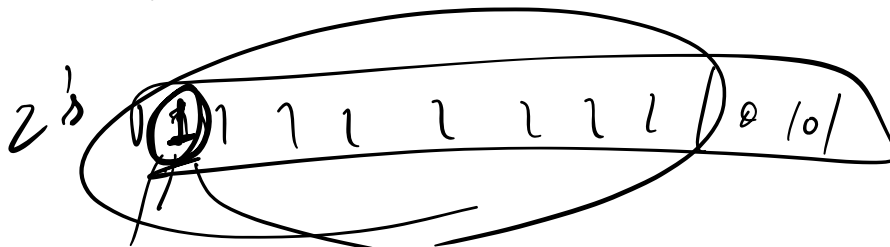
.int = 19-1



unsigned int = 2

-112

112



print a
unsigned

→ true value

Operators -

$\%$

modulo
op

Arithmetic

$+$, $-$, $*$, $/$

$2/5$

0.4

$0 = ?$

int/int

int

$0.4 \rightarrow 0$

float/int

float

double/int

double

$\frac{2.0}{5} = 0.4$

$\text{int } a = 0.4$

$a = 0$

→ Relational Operators

$=$
 $>$
 $<$
 $>=$
 $<=$
 $!=$

$a = 3$

$b = 4$

a equal to b?

$a == b$

$a > b$

$a < b$

$a >= b$

$a <= b$

$a != b$

$\text{int } a = 3;$

→ Logical op

$||$
 $!$

$a == 23$

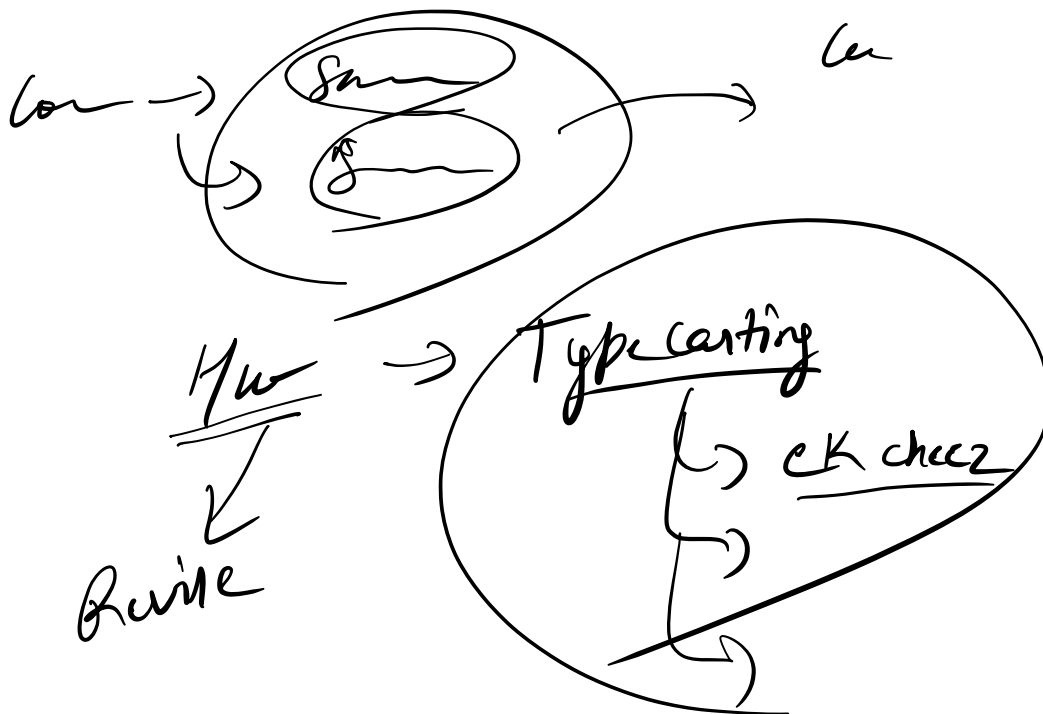
$!a$

0

$23 \quad 22 \quad 6 \text{ foot} \quad 22 \quad 80 \text{ kg}$
 \downarrow
 fit
 $23 \quad || \quad 6 \text{ foot} \quad || \quad 80 \text{ kg}$
 \downarrow
 fit and not

→ Bitwise operators

- compilation
- first prog → Namek Dunia
- Line
- Data types / Variable
- +ve / -ve / unsigned / signed
- op → Arith / Rel / logical



Programming Language