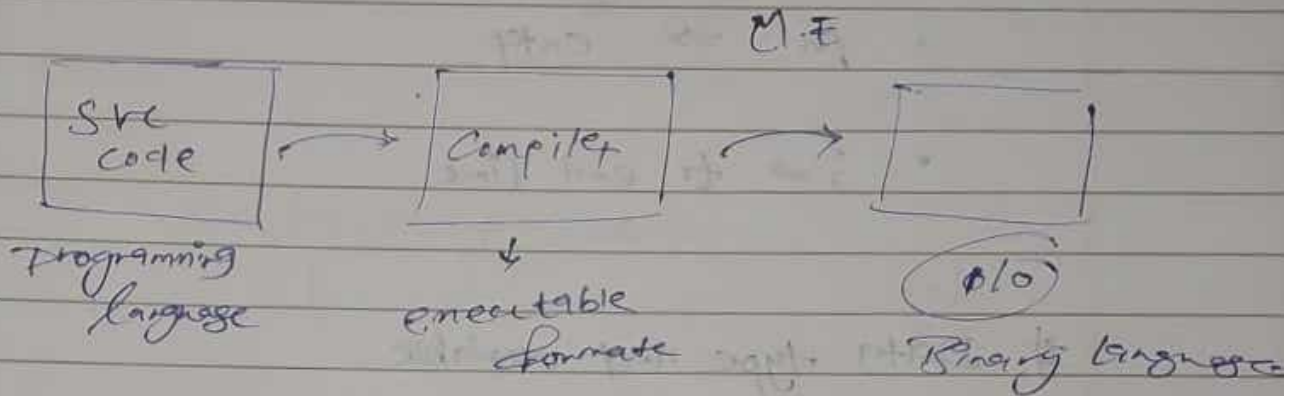


Lecture 2

First programming in C++.

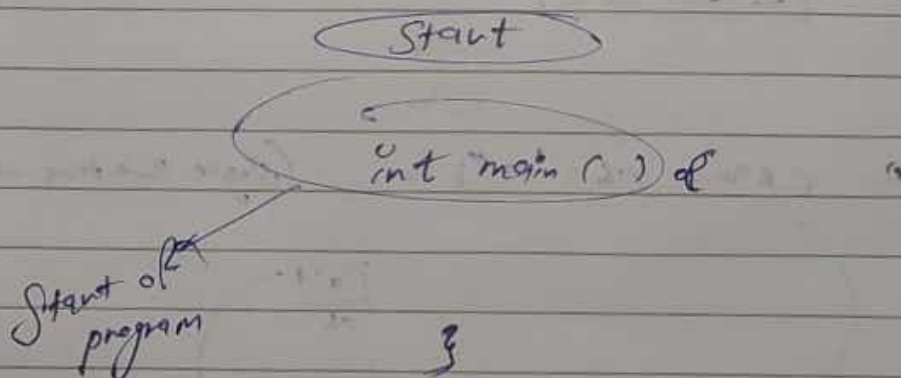


• Compiler

- ↳ Translation
- ↳ find errors

IDE \Rightarrow Integrated Development Environment.

• flowchart



• cout \Rightarrow used to print

• endl \Rightarrow new line or enter.

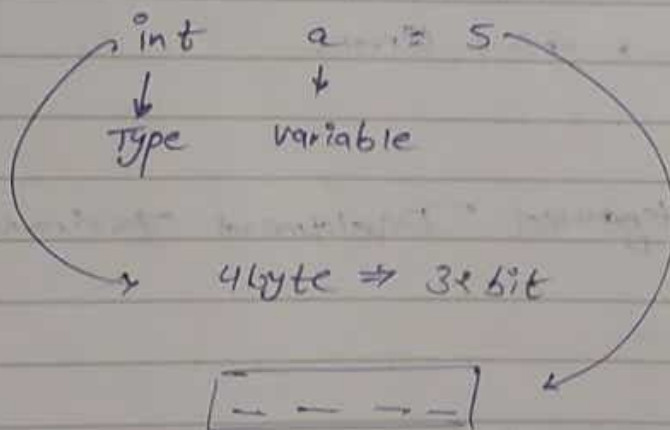
• \n \Rightarrow enter

• ; \Rightarrow to end line

Data type and variable

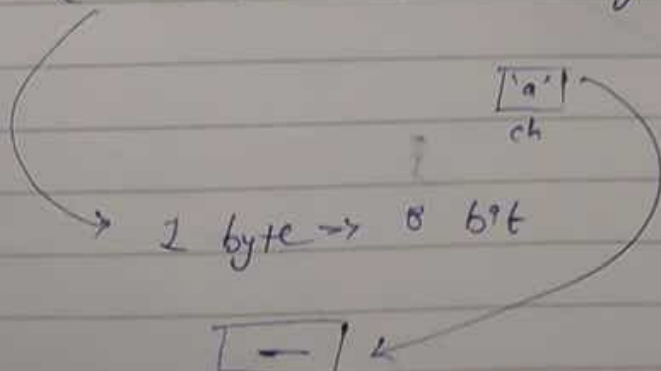
• `int a = 5;`

`5`
a



• `char ch = 'a';`

(single character constant)



• bool → true, 1
 → false, 0
 → 1 ~~bit~~ ~~byte~~ 1 byte

bool b = 1

1
b

• float f = 1.2;

→ 4 ~~8~~ byte

1.2
f

• double d = 1.23;

→ 8 byte

1.23
~~d~~

1 byte is smallest addressable size of CPU

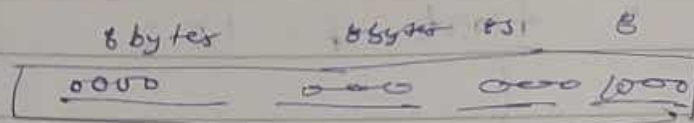
variable name

abc
ABC
A1
A_1

1 abc ✗
abc1 ✓
_ab ✓

How data is stored?

int a = 8;
 ↓
 4 byte
 Binary 1000
 4 bits



int a = 5
 (101) binary
 3 bit



char ch = 'a';

↓
 1 bytes

↘ 97

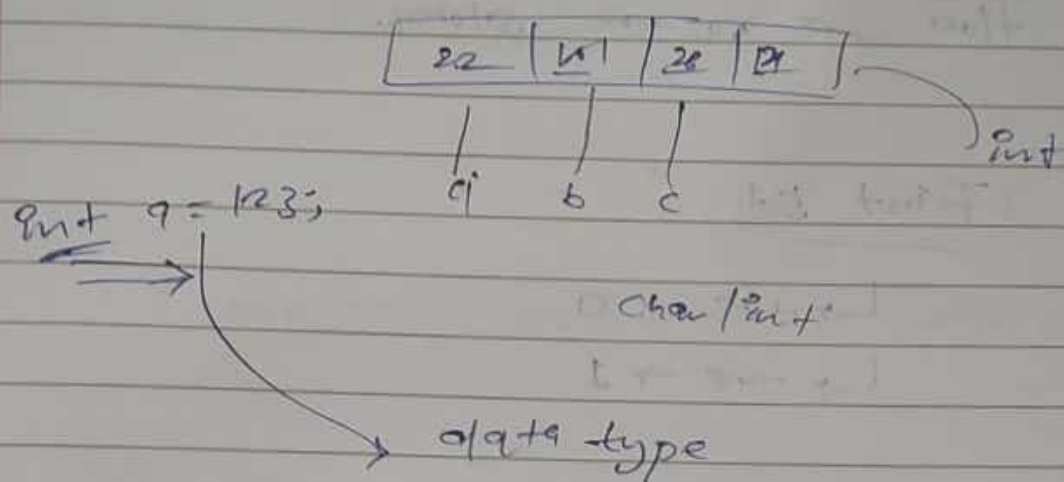
↘ Binary

H.W

ASCII Table



8 bit



⇒ Type casting

int $a = 'a';$
cout << a << endl;

char $ch = 98;$

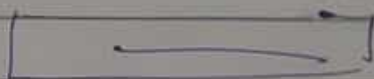
integer \rightarrow 4 bytes \rightarrow 32 bits



$$\text{max} = 2^{32} - 1$$

$$\text{min} = 2^0$$

Char \rightarrow 1 byte \rightarrow 8 bits



$$\text{max} \rightarrow 2^8 - 1$$

$$\text{min} = 0$$

How -ve no. are stored.

First bit

↳ +ve $\rightarrow 0$

↳ -ve $\rightarrow 1$

-ve no. $\rightarrow -4$ -5

↳ ignore the -ve sign
(5)

↳ Convert into binary rep
101

[00000000 101]

↳ take 2's complement
of store

0000 0000 0000 101

1st Complement $\rightarrow 1111 1111 1111 010$

2nd Complement $\rightarrow 1111 1111 1111 011$ (+1)

✓
-ve

11111 = $\sqrt{11111}$ = 1011

2's comp \rightarrow 00000000101

⇒ default

$$p_{in} + = 1, -1$$
$$6 \quad \dots \quad 2^{32} - 1$$

```
unsigned int = 2;
```

Operator

% \rightarrow modulo operator

Aithmetic

$\rightarrow +, -, *, /$

$(4/5) \rightarrow 0.4$

$\rightarrow 0$

$(int / int) \rightarrow int$

$0.4 \rightarrow 0$

$float / int \rightarrow float$

$double / int \rightarrow double$

$\frac{4.0}{5} = (0.4)$

$int \frac{4}{5} = (0.4)$

0

⇒ Relational operator.

=

>

<

>=

<=

!=

a = 3

a equal to b?

b = 4

a == b.

a > b

a < b

a >= b

a <= b.

a != b.

←

int a = 3;

→ assignment operator.

Logical operator →

&&

and

||

or

!

not