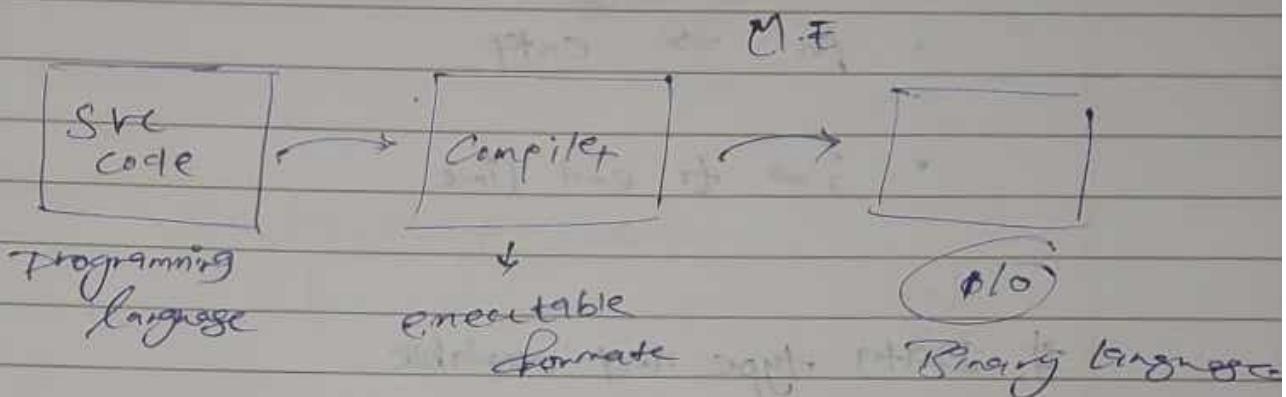


Lecture 2

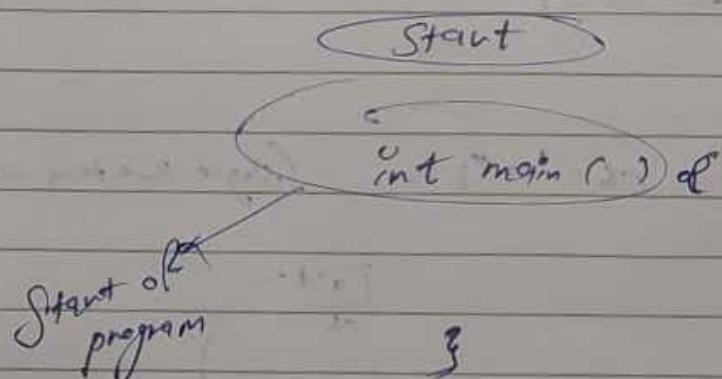
First programming in C++.



- Compiler
 - Translation
 - find errors

IDE \Rightarrow Integrated Development Environment.

Flowchart



• `cout` \Rightarrow used to print

• `endl` \Rightarrow new line or enter.

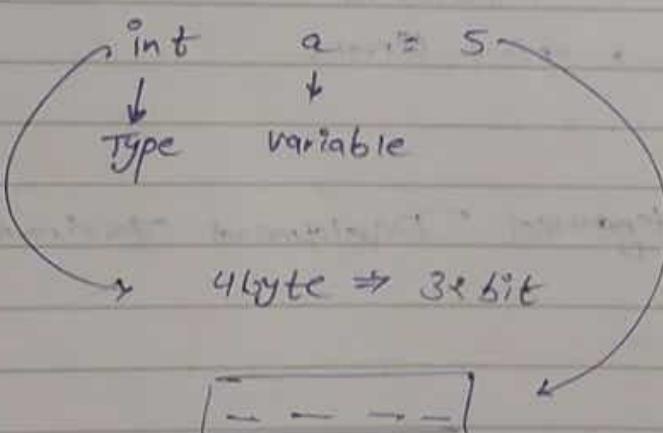
• `\n` \Rightarrow enter

• `;` \Rightarrow to end line

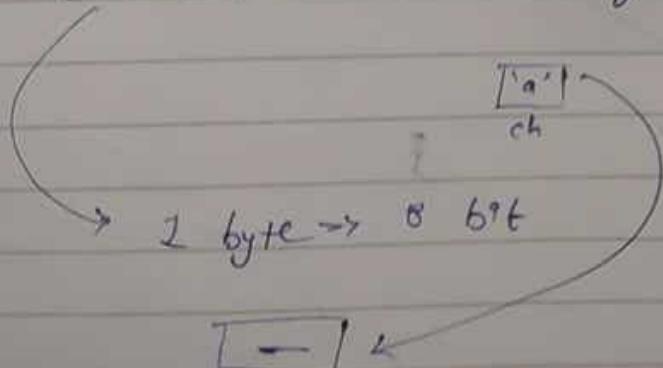
#. Data type and variable

• `int a = 5;`

5
a



• `(A char ch = 'a');` (single quoted character)



• `bool` → true, 1
 false, 0
 2 bits ~~byte~~
 1 byte

`bool b=1`

`[1]`

b

• `float f = 1.2;`

`[1.2]`

f

4 bytes

• `double d = 1.23;`

`[1.23]`

d

8 bytes

1 byte is smallest addressable size of CPU

variable name

abc

ABC

A1

A-1

abc X

ABC Y

A1 Z

A-1 Z

How data is stored?

• int a = 8;
↓
4 bytes

Binary

1000

4 bits

8 bytes 8 bytes 8 bytes
0000 000 000 1000

32 bits

• int a = 5;
3 bits

32 bits [000 000 00 101]

• char ch = 'a';

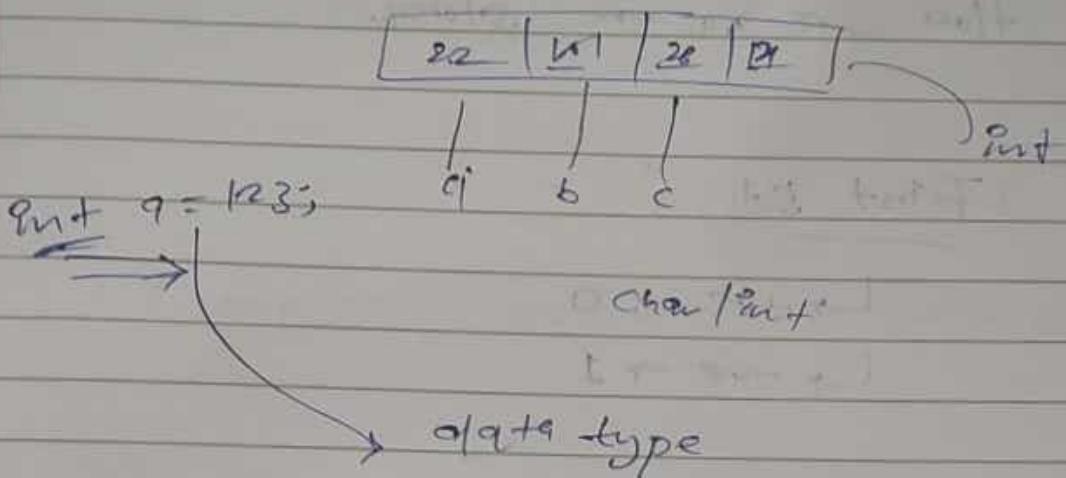
↓
1 byte

↳ g.f

↳ binary

A.S.C.I.I Table

[]
8 bit



→ Type casting

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

char ch = 'B';

integer → 4 bytes → 32 bits



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

$$\text{min} = 0$$

Char → 1 byte → 8 bits



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

$$\text{min} = 0$$

How -ve no. are stored.

First bit

↳ +ve → 0

↳ -ve → 1

-ve no. ~ -5

↳ ignore the -ve sign
⑤

↳ Convert in + binary rep
for

[00000000 101]

↳ take 2's complement
of store

0000 0000 0000 000101

1's Complement → 1111 1111 1111 010

2's Complement → ① 1111 1111 011 (+ 1)
↙
-ve

Operator.

% → modulo operation

Arithmetic

→ +, -, *, /.

$$\text{int} \rightarrow 0.4$$
$$0.4 \rightarrow 0$$

int / int → int

0.4 → 0

float / int → float

double / int → double

$$\frac{2.0}{5} = 0.4$$
$$\text{int } 4 = 0.4$$
$$0.4 \approx 0$$

⇒ Relational operator.

=

>

<

>=

<=

!=

a = 3

a equal to b?

b = 4

a == b,

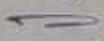
a > b

a < b

a >= b

a <= b,

a != b.



int a = 3;

→ assignment operator.

Logical operator →

|| and
or

! not