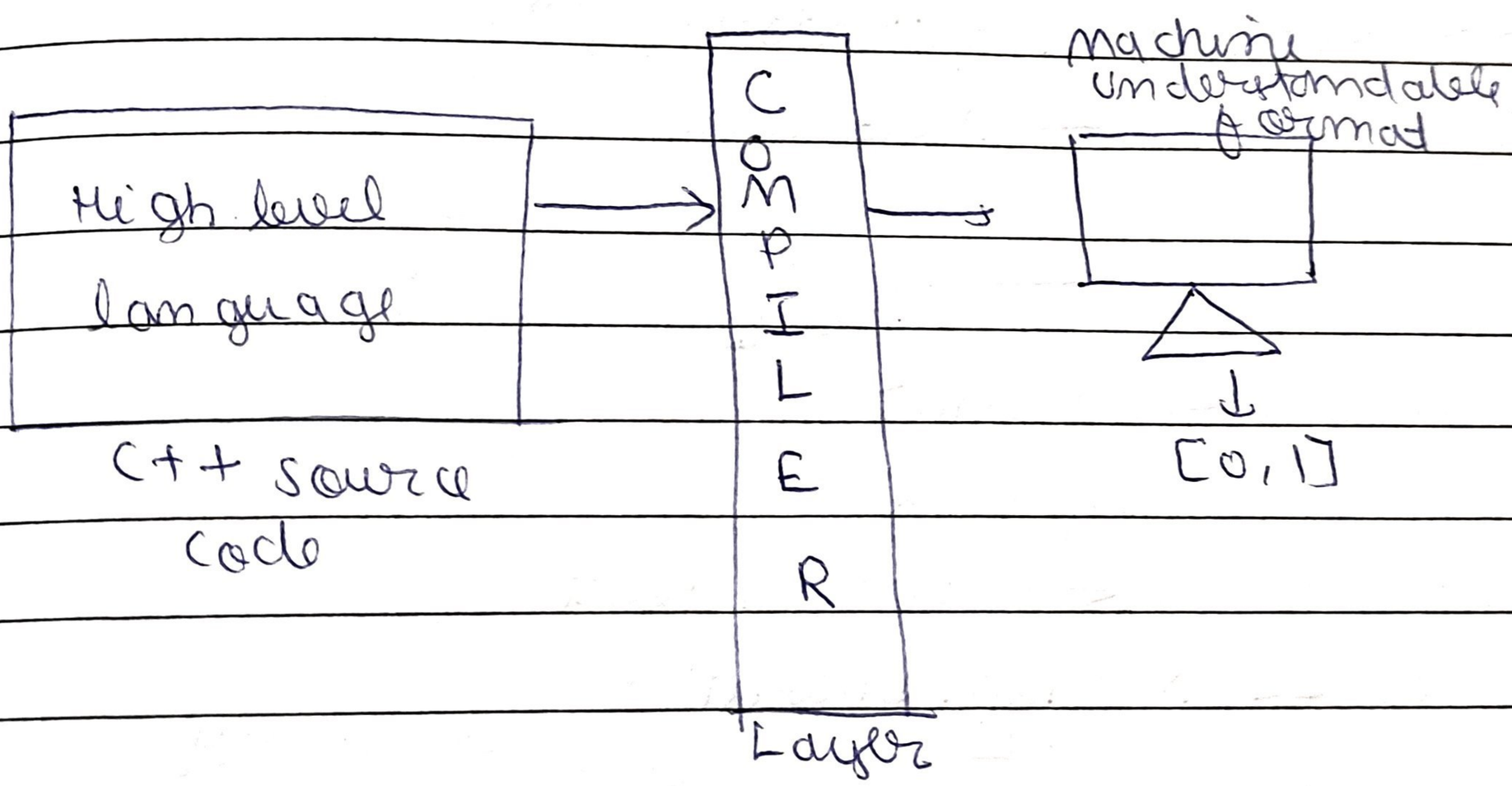
compilation process



→ We can instruct the computer to carry out real life tasks and computations

IDE (Integrated development environment

Compiler :- compiler scans the whole program
or
Convert source code into object code

DATE: / /
PAGE NO.:

Interpreter: Translate the program one statement at a time
→ It does not convert source code into object code instead scans line by line

Let's code your 1st program

code
(i) → int main() {
(ii) execution start
}
(ii) cout → Output (Print)
(iii) std → Standard Namespace
(iv) cin → Input
(v) << (Insertion operator) → to print into standard display
(vi) >> (Extraction operator)
(vii) endl or \n → next line

preprocessor → # include <iostream> → header files
using namespace std;
int main() { ← Starting Point
cout << "Namaste Bharat"
; ← ending point
→ string → terminator (end line)

Comments

Ignore to the Browser
// Pranjal Rastogi

DATE : / /
PAGE NO. :

variables

Variables are containers for storing data values

```
int a = 5
↑   ↑   ↑
data  Variable  Value
type
```

Memory

[5]
a

void → empty data

int → return type

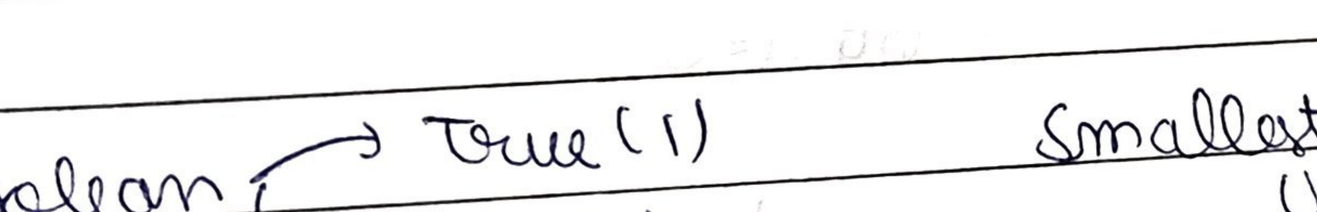
Datatype

which type of data is will store

1 byte → 8 bits
(0 or 1)

$2^{32}-1$

integer → int → 4 byte → 32 bits



2^{32}

character → char → 1 byte → 8 bit

2^8-1

char ch = 'a'; char → ASCII value

'a' → 97 'c' → 99

'A' → 65 B → 66

'b' → 98 C → 67

[a]
ch

2^8

Boolean → True (1)
→ False (0)

Smallest Memory Unit → 1 byte

1 byte → 8 bits

1 → Negative
0 → Positive

storage different 2

float → 4 byte → 32 bits b = 10 2 $2^{32}-1$

double → 8 byte → 64 bits b = 10 2 $2^{64}-1$

DATE : / /

PAGE NO. :

1 → 1

2 → 10

2^2-1 → 3 → 11

4 → 100

5 → 101

6 → 110

2^3-1 → 7 → 111

8 → 1000

9 → 1001

10 → 1010

⋮

2^4-1 → 15 → 1111

Rule → 8 4 2 1

8 4 2 1

1 1 1 → 7

4+2+1 → 7

8 4 2 1

1 0 1 0

8+0+2+0 → 10

Variable Naming conventions

(i) Pascal Case — Eg ThisIsPascal

(ii) Camel Case — Eg thisIsCamelCase

(iii) Snake Case — Eg this_is_snake_case

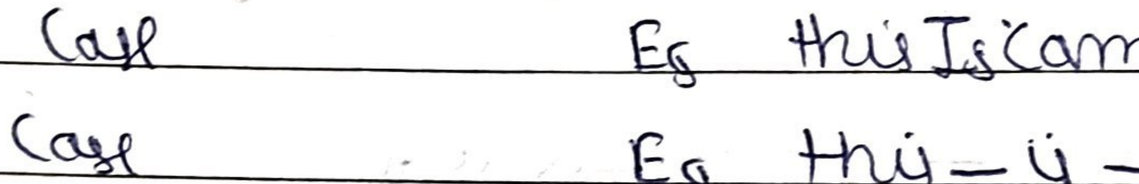
(iv) Kebab Case — Eg this-is-kebab-case

How data is stored?

Positive vs negative

5 → 101

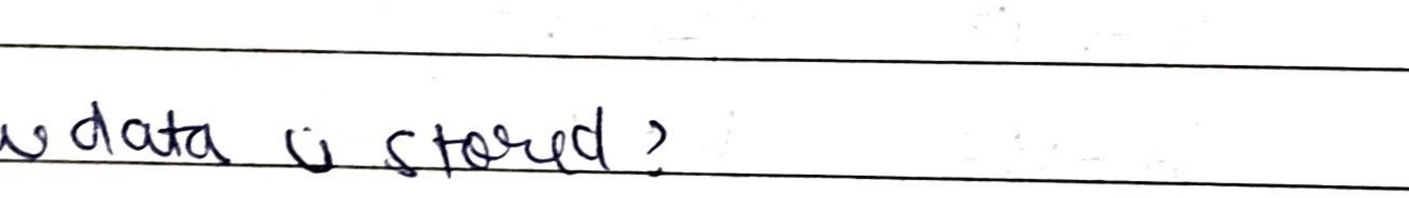
int a=5



char ch = 'a'

a = 97

1 1 0 0 0 1 0 1



DATE : / /
PAGE NO. :

+ive MSB

→ [0 |]

first Bit

-ive → [1 |]

↑ first bit

How -ive no. are stored in Memory?

↳ 2's complement

2's complement ↓

1's complement +1

00101011

1's ↳ 11010100

+1

2's → 11010101

1's complement → flip bits

1→0

0→1

7 → 111

7 → 0000 111

1's → 11111000

+1

11111001

unit a=-5

① Ignore -ive sign

[5]

② Binary equivalent

0000 - - - - 00101

③ 2's complement

000 - - - - 00101

1's → 111 - - - - 11010

+1

2's → 111 - - - - 011 ← -5

-ive

Read

1's comp → 000 - - - - 00100

+1

2's 000 - - - - 00101

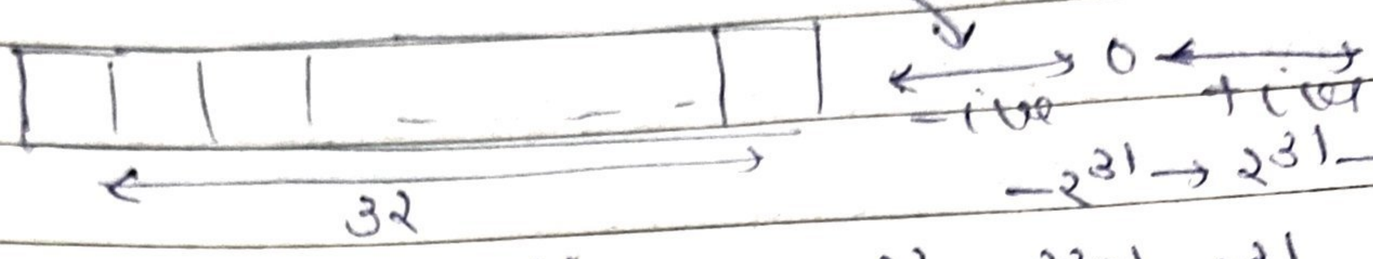
DATE : / /
PAGE NO. :

Signed vs unsigned data

```
[ int a = 5 ]
    a = -5
```

int → 4 byte → 32 bit

Signed → -ive, 0, +ive
unsigned → +ive
↓
0 → $2^{32}-1$



Total count → 2^{32}

-ive → $-2^{31} \rightarrow 2^{31}-1$

+ive → $\frac{2^{32}}{2} = 2^{32-1} = 2^{31}$

Short → 2 byte → 16 Bit

char → 1 byte → 8 bit

Total count → $2^8 \rightarrow 256$

unsigned → $0 \rightarrow 2^8-1$

signed → $-2^7 \rightarrow 2^8-1$

xyz → 6 byte → 48 bit

Total → 2^{48}

unsigned → $2^{48}-1$

signed → $-2^{47} \rightarrow 2^{47}-1$

n bits

signed → $(-2^{n-1} \rightarrow 2^{n-1}-1)$

unsigned → $0 \rightarrow 2^n-1$

Type casting

→ Type casting refers to the conversion of one data type to another in a program.

DATE : / /
PAGE NO. :

Eg char ch = '97'; Output → a
cout << ch; char → int

int a = 'b'; Output <-> 98
cout << a; int → char
↑

Implicity type conversion → automatically

Explicit type conversion → manual type
float a = (float) x;
(type) expression

Eg double d = 5.7; Output
int x = (int) d + 2; 7
cout << x;

char ch = 1 0 0 1 0 0
— — — — 0 0 0 0
ch → 0 0 1 0 1 0 0 → 1024
display 10 bit
1 byte → 8 bit

Operators

→ Arithmetic (%, +, -, *, /)
→ Relational (>, <, >=, <=, !=, ==)
→ Assignment (=) int a = 5
→ Logical && and both conditions true
|| or one condition can be true
! not False → true or true or False

→ Bitwise

float → float double → double
int int

double → double
float