(lecture - 2) 25-08-23

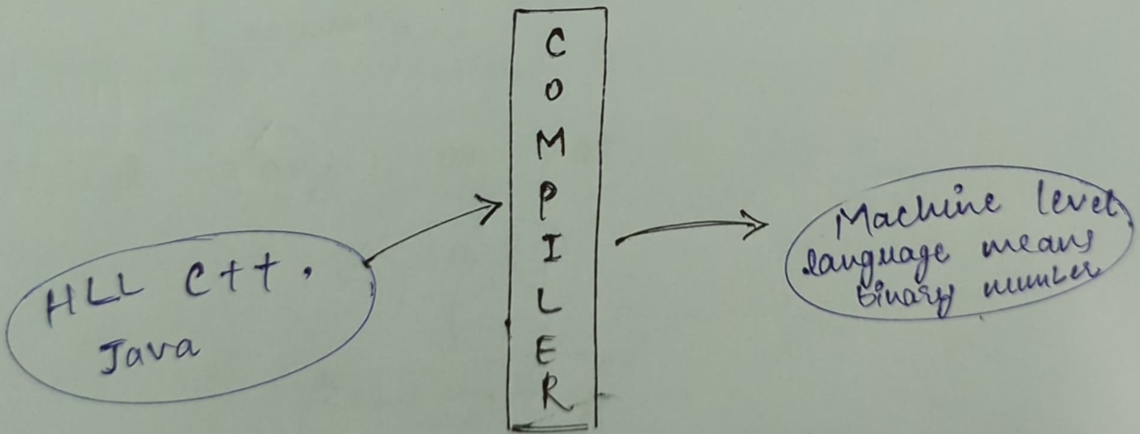
Progoamming languages :

why do we need it ?

→ To instruct the computer to carry out the tasks or to communicate with computer we need progoamming language

→ Progoamming languages can be convert into machine understandable language which is Machine level language (MLL) with the help of compiler.

→ MLL are nothing but the binary number 0/1.



IDE : Integrated development enviooment like vs code, code-blocks, xcode, sublime

Our 1st code :

int main() ← function

Start of code OR entry point of code → int main()

function ↓ It is a block of code to which we are provide input & it gives output.

int main () → main: function name

int → return type

{ } → scope of main function

```
# include <iostream>
using namespace std ;
int main {
    cout << "Love Babbar" ;
    cout << endl ;
}
```

# include → pre-processor directive

<iostream> → file name

cout → "character output"

"Love Babbar" ↓ String

cout ↓ used to print

endl ↓ end of line which is use to print in new line.

Our 1st code :

int main() ← function

Start of code OR entry point of code

function: It is a block of code to which we are provide input & it gives output.

```
int main ()
{

}
```

function name

int → return type

{ } → scope of main function

```
# include <iostream>
using namespace std ;
int main {
    cout << " Love Babbar " ;
    cout << endl ;
}
```

include → pre processor directive

<iostream> → file name

cout → "character output"

" Love Babbar " → String

cout → used to print

endl → end of line which is use to print in new line.

Eg:- Ek bada room hai → iostream hai
Room ke andar tijori hai → namespace std
Tijori ke andar paise hai → cout hai

which means to use "cout" first include the iostream file and then specify the namespace std

Taking input in C++ :-

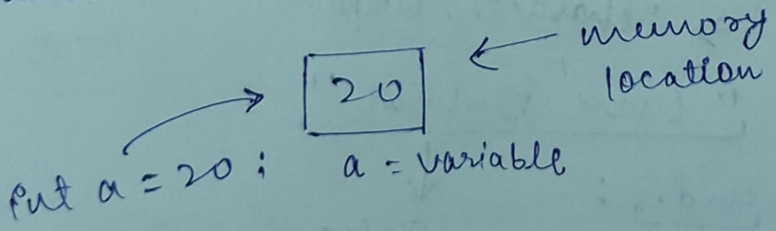
cin which stands for character input used to take input from the user.

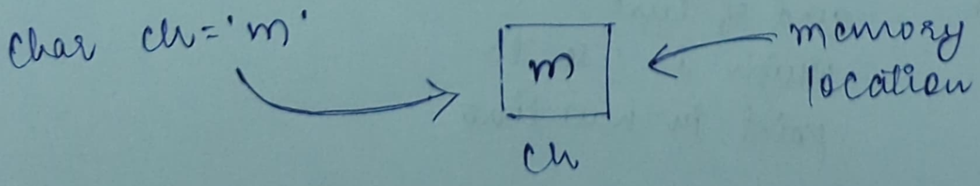
Eg:- cin >> marks ;

Datatype & Variables :-

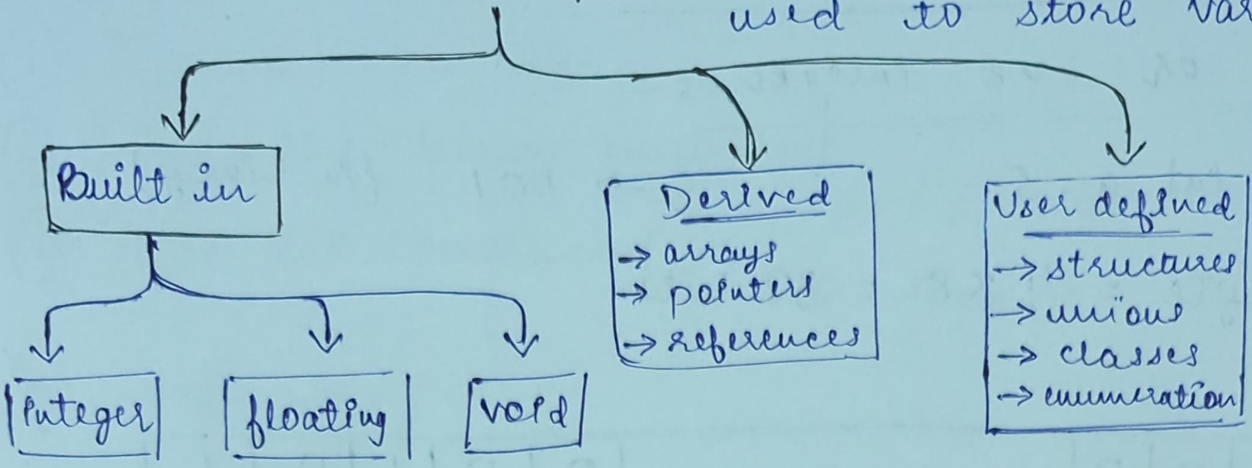
→ Variable is a name given to a memory location.

Eg:-





Datatype : Types of data which is used to store variable.

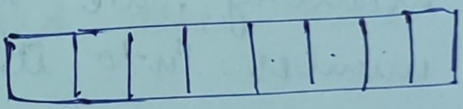


Derived : Datatype which is derived with the help of built in datatype.

User defined : Datatype which is defined by user to create their own datatype.

Size of Datatype :-

1 bytes = 8 bits



int - 4 bytes

char - 1 bytes

float - 4 bytes

double - 8 bytes

long - 4/8 bytes depends on machines

long long - 8 bytes

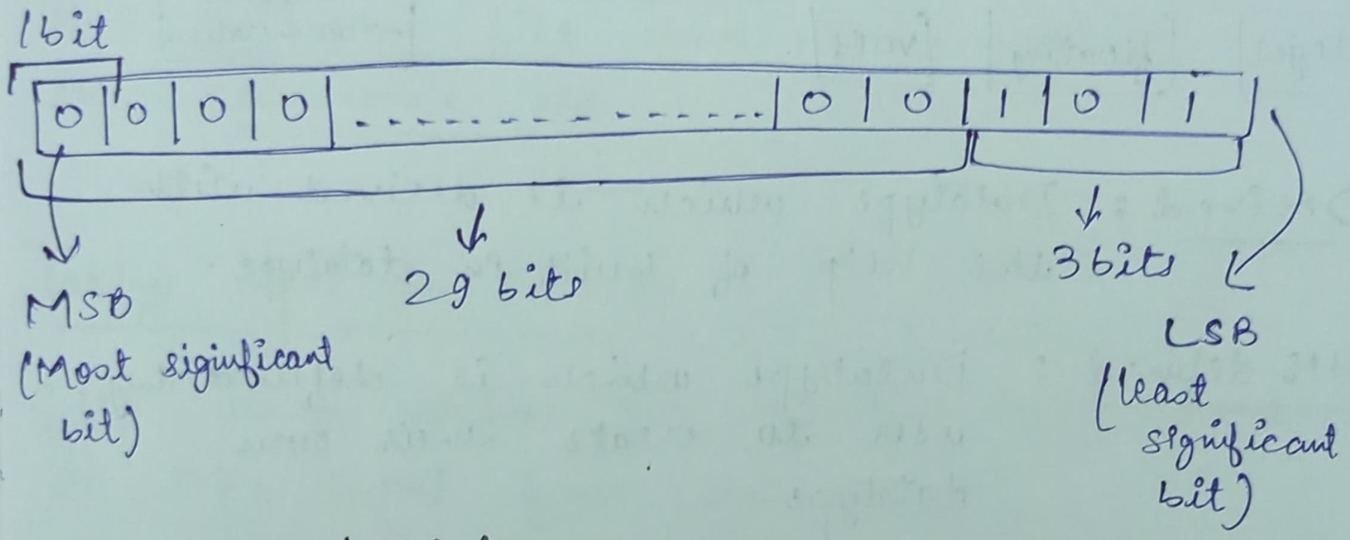
short - 2 bytes

How data is stored ?

+ve or -ve integer :-

eg:- int a = 5.   5 → 101 (in binary)

4 byte = 4 × 8 = 32 bits



+ve numbers :-

→ To store the positive number in the memory, we need to convert the number into its binary form.

→ After converting into binary, check the size of the datatype. The size of datatype is in bits.

→ The block of respective bits created in the memory and the number is stored in the form of bits.

-ve numbers :

(i) Ignore -ve sign.

(ii) Find the binary equivalent

(iii) Take 2's complement → 1's complement
→ then +1

eg:- int a = -5

(i) ignore -ve sign :-

a = 5

(ii) Binary equivalent :-

```
5 = 00000000 00000000 00000000
                               00000101
```

(iii) 2's complement :-

To find 2's complement first we need to find 1's complement and then add +1.

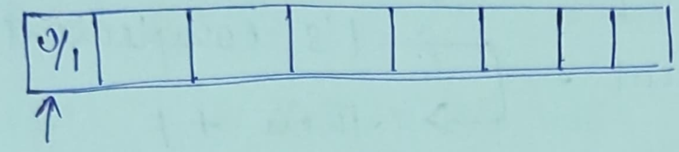
```
5 = 11111111 11111111 11111111 11111010
                                     +1
    -----------------------------------
    11111111 11111111 11111111 11111011
    -----------------------------------
```

↗

-5 will store in memory like above shown

Range of Datatypes :-

Char → 1 byte → 8bit



Each block consists of two choices either 0 & 1

Total combination = $2^8 = 256$

Possible combination = $0 \rightarrow (2^8-1)$

$= 0 \rightarrow 255$

In general the combination is 2^n-1

int → 4bytes → 32 bits

Total combination = 2^{32}

possible combination = $2^{32}-1$

Range = $0 \rightarrow 2^{32}-1$

long → 8bytes → 64 bits

possible combination = $2^{64}-1$

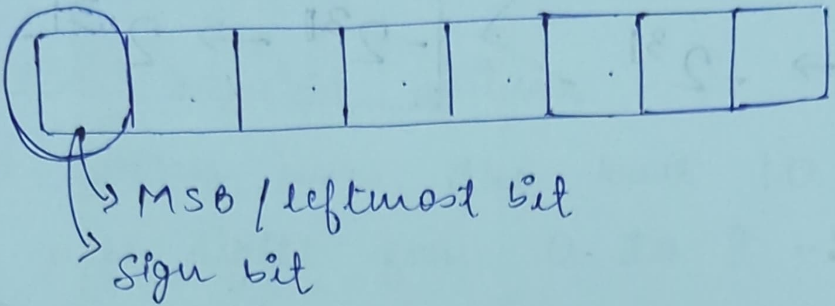
Total combination = 2^{64}

Range = $0 \rightarrow 2^{64}-1$

Signed vs Unsigned Data :-

Signed - It can store +ve, -ve, 0 value.

Unsigned - It can store only +ve value.



Note : * → If sign bit is 0 the number is positive..

* → If sign bit is 1 the number is negative.

to find range :-

unsigned = $0 \rightarrow 2^{48} - 1$

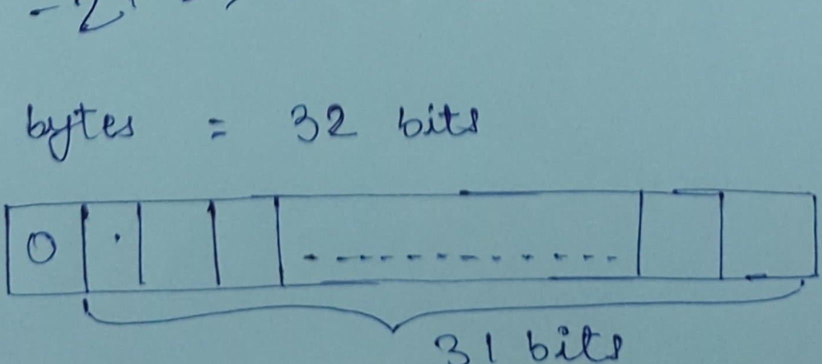
signed = $\frac{2^{48}}{2} = 2^{47}$ [we need to divide in two equal parts]

$\therefore -2^{47} \rightarrow 2^{47} - 1$

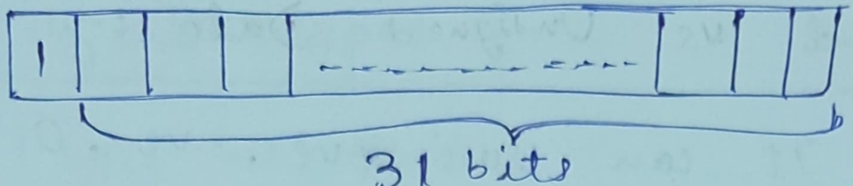
$-2^{47} \rightarrow 0 \rightarrow 2^{47} - 1$

int = 4 bytes = 32 bits

+ve =



-ve :



Range

+ve = 0 → $2^{31}-1$

-ve = -1 → -2^{31}

$\boxed{-2^{31} \to 2^{31}-1}$

Operator :-

(i) Arithmetic → +, ×, −, /, %

(ii) Relational → >, <, <=, >=, !=, ==

(iii) Assignment → &&, ||, !

logical and, logical or, logical not

(iv) logical

→ =

(v) Bitwise