Lecture - 2

Q. What is Programming Language?

Ans:-

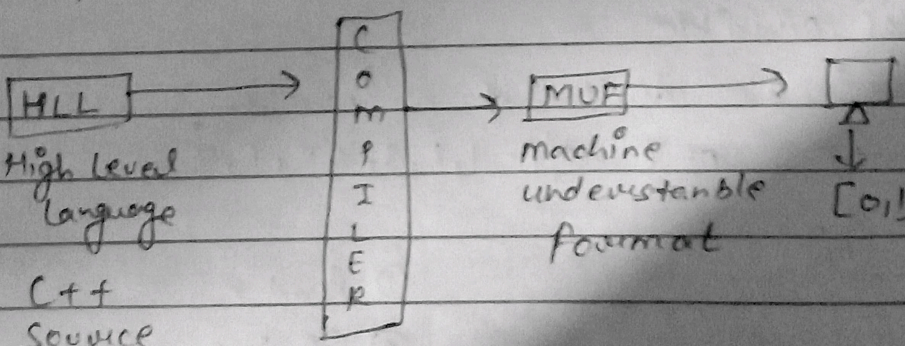
① A language using which, we can instruct the computer to carry out real life tasks & computation is called programming Lang.

② Programming language has a fixed set of rules according to which program could be written in it.

③ Every language has its own compilers & interpeters

④ once program is compiled & linked, its execution is created & computers run a program.

Compilation Process:-



C++ :- (Programming language)

IDE :- Integrated Development Environment

Various IDE's are:-

① VS code
② Sublime
③ X code
④ Codeblocks

int main () {
}

It is a function which start & end the code. Compiler firstly find int main then execute the code.

* cout :- cout is used to print in c++

* cin :- cin is used to take input in c++

* # include <iostream> :- It is header file where cout & cin implementation file stored

using namespace std; :- (It is a generic way).

① namespace is a particular region where scope of identifier define.

② std is the standard version of iostream ont cout

Q- Print "Namaste Bharat" ?

Ans/ # include <iostream>
using namespace std;
int main () {
cout << "Namaste Bharat" << endl;
}

* cout << "Hi" ;
↓ ↓ ↳ end
To print ↳ string
↳ To print into standard display

* endl ;- It is used for cursor goes to new line.

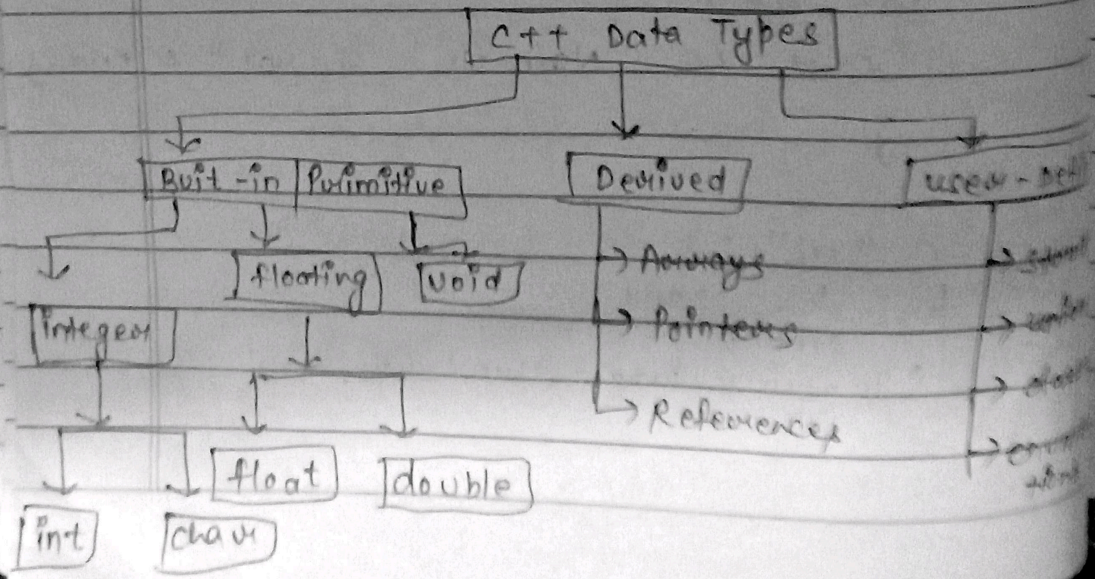
* '\n' :- It is also used for cursor goes new line.

* (//) or (/* → */) → It is used for comment use. Comment is non-executable line in a program

* >> :- insertion operator
* << :- extraction operator

Ex:- cout << "Enter a number:" << endl;
int n;
cin >> n;
cout << "you enter" << n << endl;

Data Types & variables :- A variable in C is a memory location associated with some name in order to store some form of data & retrieve it when * Th
& data types are used to tell the variable the type of data they can store.



Q. What is variable?

Ans: Variable is a container for storing the data values.

Ex:- int a = 5

Data type ↓ Variable ↓ value

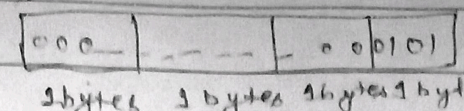
Data types are "Types of Data"

Data types	32-bit CPU		64 bit CPU	
	size (Bytes)	Range	size (Bytes)	Range
char	1	-128 to 127	1	-128 to 127
short	2	-32,768 to 32,767	2	-32768 to 32767
int	4	-2^{31} to $2^{31}-1$	4	-2^{31} to $-2^{31}-1$
long	4		8	
long long	8		8	
float	4	3.4E+/-38	4	3.4E+/-38
double	8	1.7E+/-308	8	1.7E+/-308

* Data type is which type of data is store.
* These are pre-defined data types called primitive data types.
* void means empty

* Important Points:-
  1. 1 bytes = 8 bits
  2. int → 4 bytes → 32 bits

1) If int a = 5; 5 - 101



8) char ch = 'a';

1 byte → [| | | | | | | |] → 2^8 → 256

0 1 8 bit

256 characters can define in different character define.

Each characters is mapped from ASCII value.

→ ASCII :- American standard code for Information Interchange

→ ASCII TABLE :-

a to z → 97 to 122

A to Z → 65 to 90

0 to 9 → 48 to 57

* char ch = 'a';

[97] → [0100 0011]

0

*3) Boolean Data Types :- It contain 1 byte

It has two types :-

(a) True → 1

(b) False → 0

Syntax :-

bool flag = 1;

DataType VariableName ← value

* 1 bytes is a minimum addressable memory in CPU.

4) Float :- It is decimal types values used.

Syntax:-

```
float f = 1.2 ;
Double f2 = 1.2 ;
```

Float	Double
Take 4 bytes	Take 8 bytes
Take normal no.	Take language number
Take less memory	Take large memory

Decimal to Binary Rule :- (8421)

1 → 1
2 → 10
3 → 11
4 → 100
5 → 101
6 → 110
7 → 111
8 → 1000
9 → 1001
10 → 1010
15 → 1111

General Formula

$2^n - 1$ $n = 1, 2, 3 \cdots$

short :- It take 2 bytes

Interview

Q:- What a variable naming conventions?

Ans:- The general rules of naming variables are:-

1. Names can contain letters, digit & underscore(_)
2. Name begin with letter & underscore(_) not number
3. Name are case sensitive (my var & my var are different)
4. Name connot contain white space or special character like !, #, %, etc.

⑤ Reserved words (e.g. int) cannot be used as names.

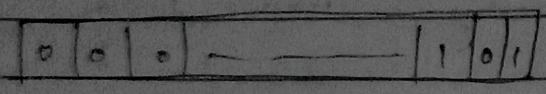
* size of() :- It is used for finding the size of data types of a variable.

Syntax:- int a = 5
cout << sizeof(a) << endl;

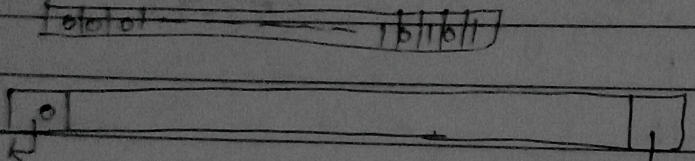
Q-1- How data is stored? (Positive / Negative)

Ans:- int a = 5 5 → 101

+ve



char ch = 'a' a → 97
↳ 1010 1



Note:- +ve no. MSB is 0
-ve no. MSB is 1

→ -ve no. stored by doing 2's compliment

Eg:- 00 1001
1's → 11 0110
2's → +1
11 0111

Step to -ve no. store → int a = -5

Step ① ignore -ve sign
② Binary equivalent of number
③ Find 2's compilement

int a = -5

int a = 5

5 → 101

0000 ---- 0101

1's → 1111 ---- 1010

2's → +1

1011 ---- 1011

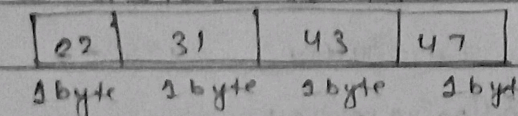
-ve

To Read again it

① find 2's compilement

② check MSB (First Bit)

Q- How I decide it is int, char in 4 bytes?



It decide upon data type & scope of memory which was taken by data type

Data type always provide → Type of Data, Size of Data

* signed or unsigned :-

signed : It store +ve or -ve no.

unsigned : It store only +ve no.

signed Range :- -2^{31} to $2^{31}-1$

unsigned Range :- 0 to $2^{32}-1$

Compiler take by default is signed

→ short → 2 bytes → 16 bit → 2^{16}

Range:-

unsigned → 0 to $2^{16}-1$

signed → -2^{15} to $2^{15}-1$

→ Range Generic Formula:- n = bits

signed → -2^{n-1} to $2^{n-1}-1$

unsigned → 0 to 2^{n-1}

Type Casting:- Type Casting refers to the conversion of one data to another in a program.

Type casting can be done in two ways

① automatically by computer

② manually by programmer.

Typecasting also known as Type Conversion

* Implicit Type Casting:-

① known as Automatic type casting

② Automatically converted from one data to another

③ It can only apply in a program of both variables are compatible with each other.

④ All data type is automatically upgraded to the largest type without losing any info

* Explicit Type casting:-

① known as manual type casting

② manually cast by programmer to change from one data type to another type of a program.

③ It do not require checking the compatibility of the variables.

④ It use the cast() operator to change the type of variable.

Ex:- char ch = 97;
cout << ch << endl;

o/p → 'a'

Ex:- int num = 'b';
cout << num << endl;

o/p → 98;

} Implicit T.C

Ex:- float = float(a) + 2;
type (expression)

} → Explicit T.C

Ex:- double d = 5.7;
int x = int(d) + 2;
cout << "value of x" << x << endl;

o/p → 7

** OPERATORS :-

① Arithmetic :- %, +, -, /, *, //
② Relational :- >, <, >=, <=, !=, =, ==
③ Assignment :- := Ex:- int a = 5
④ Logical :- &&, ||, !
⑤ Bitwise :-

All condition are true

1	0	0
0	1	0
0	0	0
1	1	1

If one condition is true then true

1	0	1
0	1	1
0	0	0
1	1	1

* Logical operators used when we have multiple conditions

Q- If char = 234234;
& char take 1 byte → 2^8 → 255
& then what print?

Ans/- 000- ------ 11000+

It take last 8 digits & these 8 digit map to ASCII table value that displays

→ Always remember in Divisions-

① $\frac{int}{int}$ = int ② $\frac{int}{float}$ = float

③ $\frac{float}{int}$ = float ④ $\frac{float}{float}$ = float

⑤ $\frac{double}{int}$ = double ⑥ $\frac{float}{double}$ = double.

⑦ $\frac{double}{float}$ = double

Q- What is precedence & Associativity & it's table?

Ans/- Operators precedence specify the order of operations in expression that contain more than operators

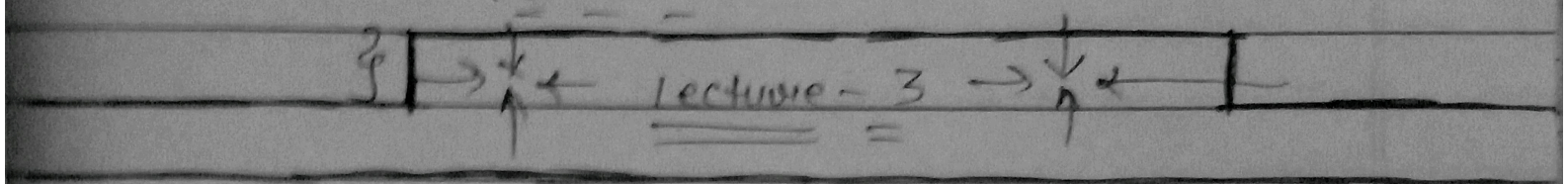
operator Associativity, in an expression that contains multiple operator with same precendence, an operand is grouped with the one on it's "left" or in "right"

→ check table in Github Repo

Q: How to create own namespace?

Ans:- C++ allows us to define our own namespace via the "namespace" keyword. Namespace that you create for your own declaration are called user-defined namespace. Namespace provided by C++ (as global namespace) or by libraries are not considered user-defined namespace.

Ex:- namespace do something
{ int dosomething ( )
{
}
}



Conditional :-