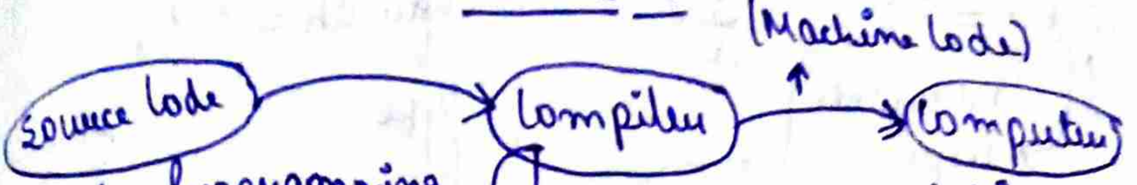
Subject:- DataStructure & Algorithm

Lecture:-2



(using Programming language)

a=1
b=2
c=a+b

This will not be understable by a computer directly.

It create executable file. (Binary lang.)

using compiler we can convert source code into machine code.

which understands 0/1

Compiler:- •) Translates source code to machine code.
•) It helps to find errors.

IDE:- Integrated development enviornment in which where we are able to write & execute our code.
ex:- Codeblocks, vscode & etc.

Prog:-1 Namaste duniya

```
#include <iostream>
int main() → from here our program starts.
{
  cout << "NamasteDuniya";  } → this will give error.
                            } → ending a line
}
```

→ functionality provide by C++ to print the things

```
#include <iostream.h> } This is a file where code of (cout) is already written by C++.
using namespace std;
int main()
{
  cout << "Namaste Duniya" << endl;  → for getting new line
  cout << "Hello" << '\n';  → for new line
}
```

<< → we use to display something.

There are so many namespace of cout but we required cout of std.

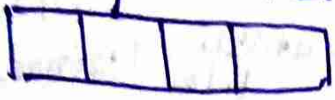
Datatypes & variables :-

int a = 5;

↳ variable

↳ type

↳ 4 byte → 32 bit



$\boxed{5}$
a

char ch = 'a';

↓

type

↳ 1 byte = 8 bit

$\boxed{a}$
ch

bool → true (1)
→ false (0)

bool b = 1

$\boxed{1}$
b

float f = 1.2;

↳ type

$\boxed{1.2}$
f

double d = 1.23;

↳ 8 byte

variable name :- abc, ABC, A1, B1 --- ✓

1ab ×

How data is stored?

int a = 8; → Binary (1000)

↳ 4 byte

$\boxed{0000 \ 0000 \ 0000 \ 1000}$

(32bit)

int a = 5 → (101)

$\boxed{0000 \ 0000 \ 0000 \ 0101}$

char ch = 'a';

↳ 1 Byte

↳ 97 (ASCII Table)

(0/1) (8Bit)

22	32	21	16

↳ Integer / character

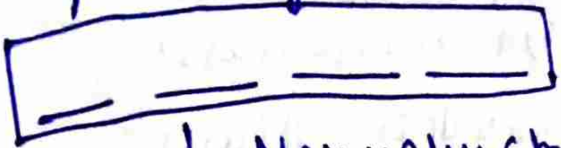
↳ using datatype

Type casting:- To convert from one type to another type.

ex:- int a = 'a'; (a→97)
cout << a << endl;

This character typecast into a integer.

Integer → 4 byte → 32 bit



↳ Max value store ⇒ $2^{32}-1$

char → 1 byte → 8 bit

↳ max value ⇒ 2^8-1

2^{16} → Integer but I want to store in a character. It will take last bits & whatever the value will be it will shown

How -ve no's are stored?

first bit
↳ +ve ⇒ 0
↳ -ve ⇒ 1

-ve no → 5 → simple take 2's complement to print.
↳ Ignore -ve sign (5)
↳ convert into binary no

0000 0000 0000 0101

↳ Take 2's complement & store

0000 0000 0000 0101

1's complement:- 1111 1111 1111 1010 ↓ (Just change 1→0 & 0→1)

+1

2's complement:- 1111 1111 1111 1011
↳ -ve

operators:- +, -, *, %, /   Relational operator:- ==, <=, >=, !=

2/10 ⇒ 0