Day- 02

Date
Page No.

14/9/23

Lecture - 02

Data type & Variable

Compiler:- It convert the source code machine excuatable formate.

Compiler :-
- Translation
- fixed error

= **IDE :-** Integrated development Enviroument.
Ex:- Codeblock, VS code.

A Enviroment that help to you write, run and even debug (in some cases) code in programming language

- **first program :-**

```
int main() {
}
```
→ These brackets show the scope of the main function

- **cout :-** It is used for print anything.
  Eg : `cout << "NABASTE";`

This cout is already defined in file.

- **using namespace std; :-** to write the standard output.
- **using cout :** use '<<' after 'cout' to display some

thing to standard output with in std name space.

\n → maped with endl;

endl :- used for enter in a new line. (enter)

```
#include <iostream>
using namespace std;

int main() {
        cout << "Nabastey" << endl ;

}
```

g → used for show to end line.

Data types & variable :-

- int a → variable
  - integer type.
  - → 4 byte → 32 bit
- char ch = 'a' → value of ch
  - 1 byte → 8bit.
  - character
  - variable.
- Boolean : bool
  - true → 1
  - false → 0.
  - 1 bit.
- float : float f = 1.2; → 8 byte - 64 bit
  - data type
  - variable
  - value
- double d = 1.23 → 8-byte. — 64 bit.
  - data type
  - variable
  - value

= Variable name :-

- abc
- ABC
- A1
- A_1

} create variable.

Sizeof :- It is used for what is space of data typ

= How data store :- convert in binary

```
int a = 8;
```

↓

4-byte.

→ char ch = 'a' ——— Ascii table

97 → Binary → store.

= Type Casting

conversion of one data type to another data type.

```
char ch = 98 ——> B
int a = 'a' ——> 97
```

integer 4-byte -32 bit

Range of integer $-2^{32}-1$, min = 0

= How can store negitive number :-

first bit

- 0. → ignore the -ive
- 1. → convert into binary
  → take 2's complement & store.

Operators :-

Arithematic → % modulo operator
+, -, *, /

int/int → int float/int → float

double/int → double

Relational operator :- ==, >, <, <=, >=, !=

Logical operator :- && || !

&& → all condition should be true

|| → only one condition true

! → not
true → false