

Programming language :- to convert the source code to the binary format which is computer understandable language to get output

Compiler  
 ↗ translation  
 ↗ find errors

IDE - integrated development environment

Ex:- code blocks, vs code — Replit

According to our flowchart we have very first start  
 So in programming there is start like

Int main(l) {

}

first program :-

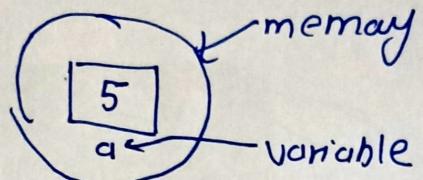
```

#include <iostream>
int main()
{
    using namespace std
    int main()
    {
        cout << "Hello World" << endl;
    }
}
  
```

`cout << **` → to write, to display anything  
 under heading `cout`.  
`<n` → new line - enter.  
`\n` → new line - enter  
`;` → to complete our line.

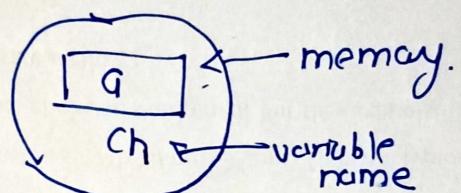
## Data types & Variables :-

`int a = 5;`  
 ↓ type  
 ↓ variable



`int - 4 byte → 32 bit`

$\rightarrow$  `char ch = 'a';`  
 ↓ type



`char → 1 byte → 8 bit`

$\rightarrow$  `bool` → True - 1  
 False - 0  
 ↓ 1 bit

`bool b = 1;`

$\rightarrow$  `float f = 1.2;`

`Float - 8 byte`

$\rightarrow$  `double d = 1.23;`  
 ↓ 8 byte

Variable name :-

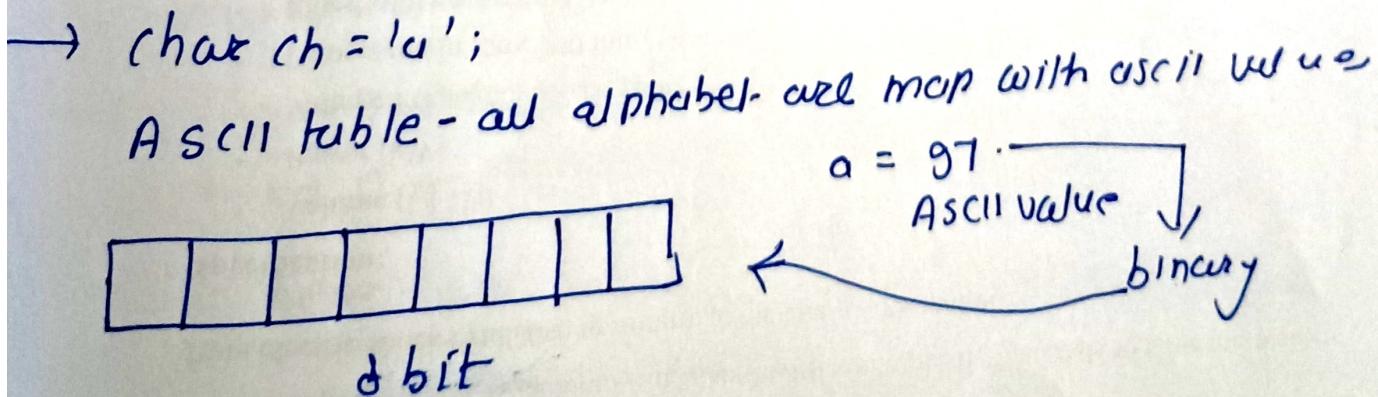
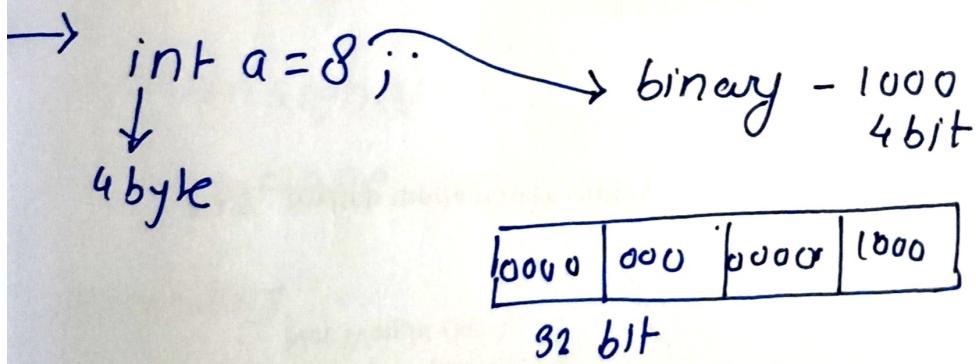
- abc ✓
- ABC ✓
- A1 ✓
- 1abc ✗, never start name by number.
- ab1 ✓
- a1 ✗

```

#include <iostream>
using namespace std;
int main() {
    int a = 123;
    cout << a << endl;
    char b = 'v';
    cout << b << endl;
    bool bl = true;
    cout << bl << endl;
    float f = 1.2;
    cout << f << endl;
    double d = 1.23;
    cout << d << endl;
    int size = sizeof(d);
    cout << "Size of a is :" << size << endl;
}

```

How data is stored?



Type casting :- to convert one datatype to another one

Ex:-

```
int a = 'a';  
cout << a << endl; → output - 97
```

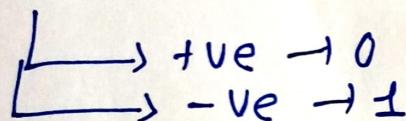
- it give value by the ASCII table value and map it.

```
char ch = 98;  
cout << ch << endl; → output - b
```

- char ch1 = 123456; → implicit conversion for ch1 will  
cout << ch1 << endl; to 123456 it become last  
value 64

→ in case of large datatype into small datatype  
there are some bit are converted no at all.

First bit →



→ unsigned - gives only positive number

→ unsigned int = 112;

operators :-

% - modulo operator.

→ 215 → 0.2 - normally in math  
→ 0 - programming because -  $\frac{\text{int}}{\text{int}} \rightarrow \text{int}$

here  $\frac{\text{int}}{\text{int}} \frac{2}{5} \Rightarrow \frac{0.2}{\text{float}} X$

hence it gives zero

→ Relational operators :-  $=, >, <, \geq, \leq, \neq$

Ex :- int a = 2;  
int b = 3;

bool first = (a == b);  
cout << first << endl; — 0

bool second = (a > b);  
cout << second << endl; — 0

bool third = (a < b);  
cout << third << endl; — 1

Assignment operator :-

int a = 3;  
└ assignment

Logical operators - &&, ||, !