L2-Basics of Programming

Special class

Love Babbar • Apr 3, 2025

L2 - Basics of Programming

- Variables and Operators
- Write your 1st Program

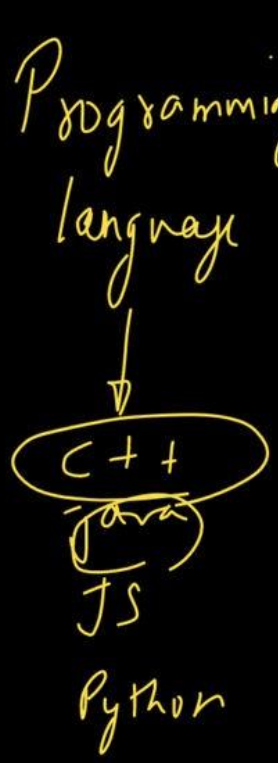
—> by Codehelp

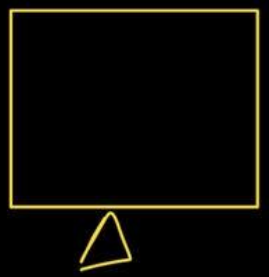
Programming Language ?

What ?

Why ?

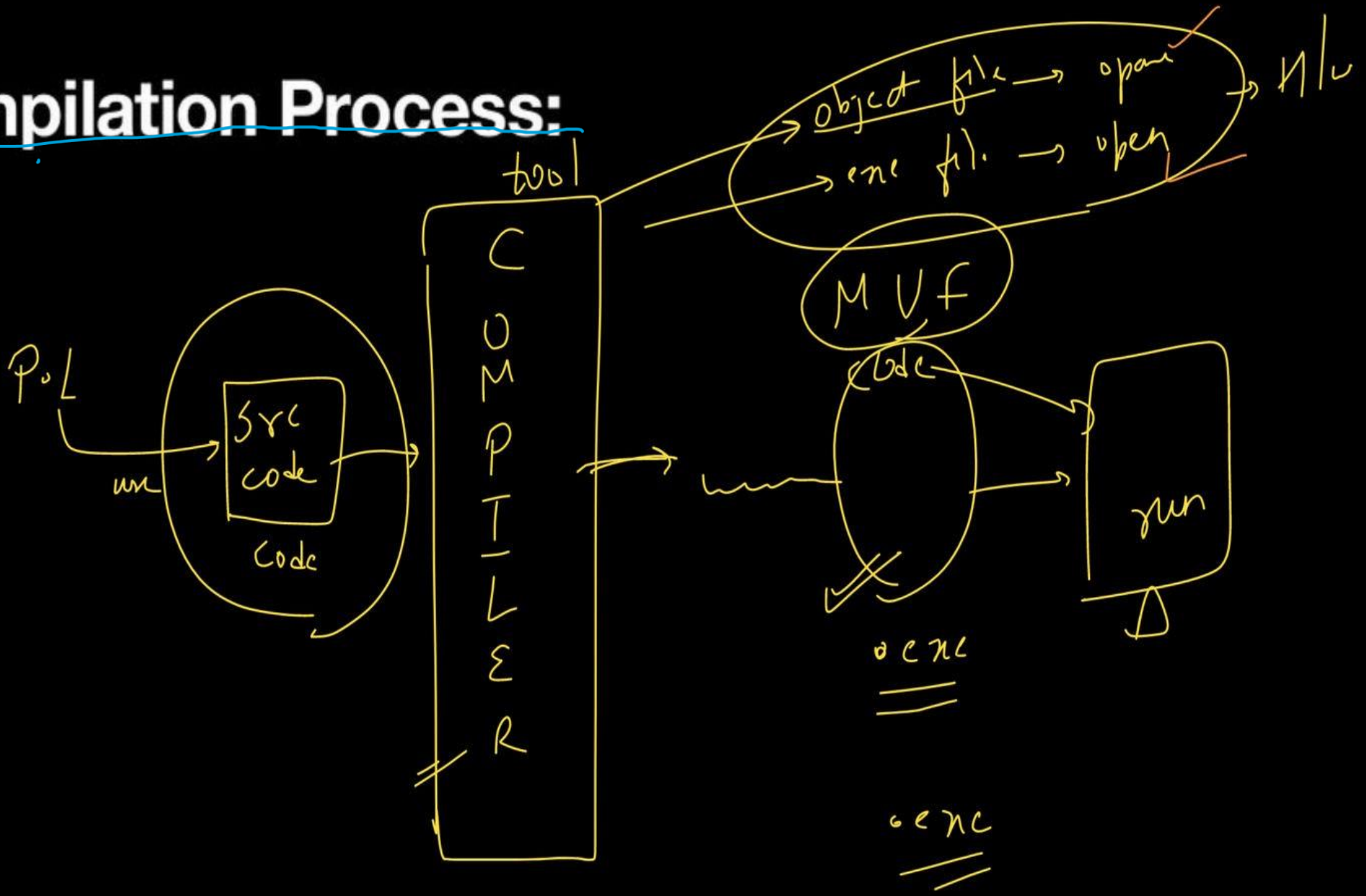
Standard



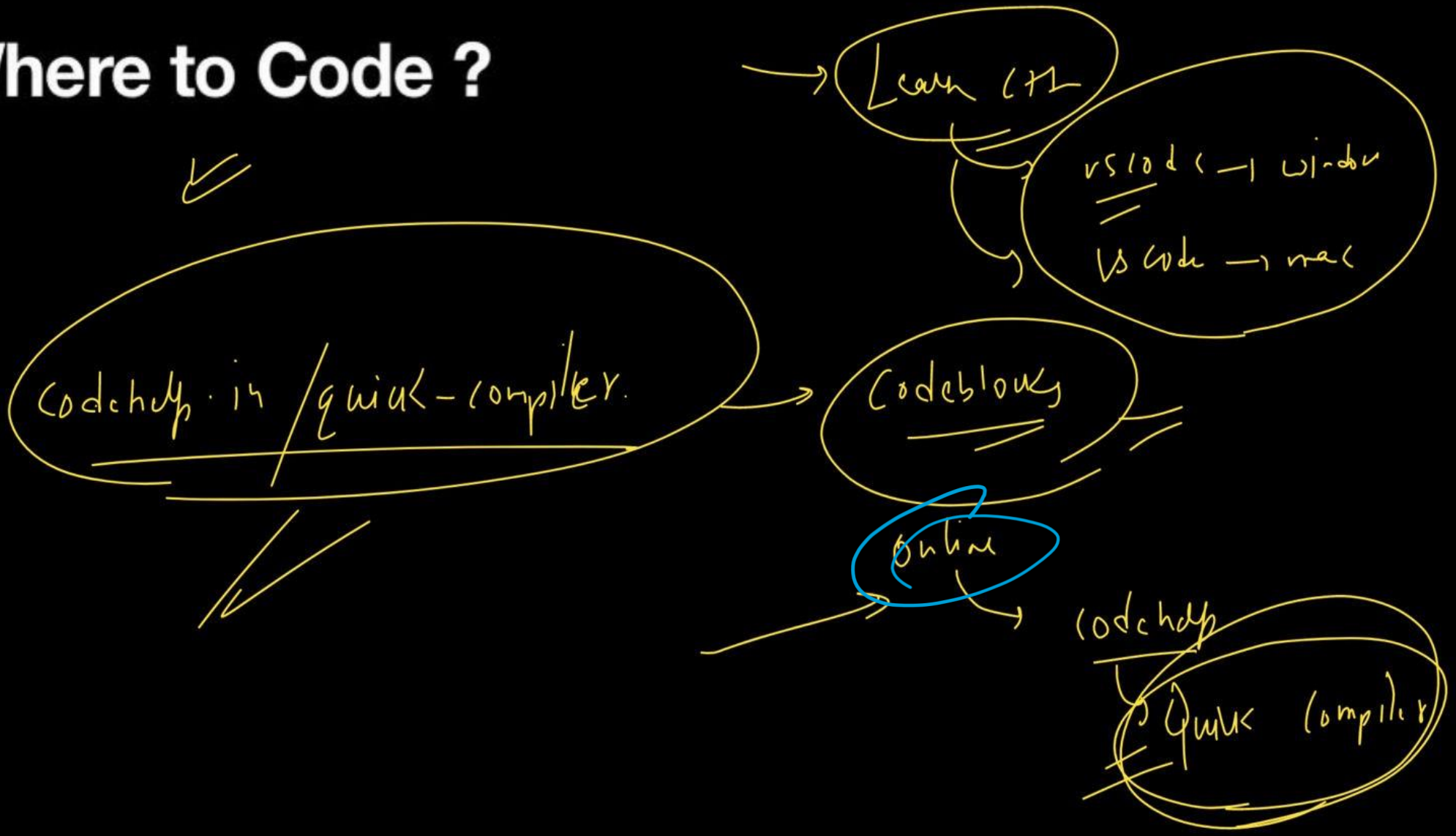


task

Compilation Process:



Where to Code ?



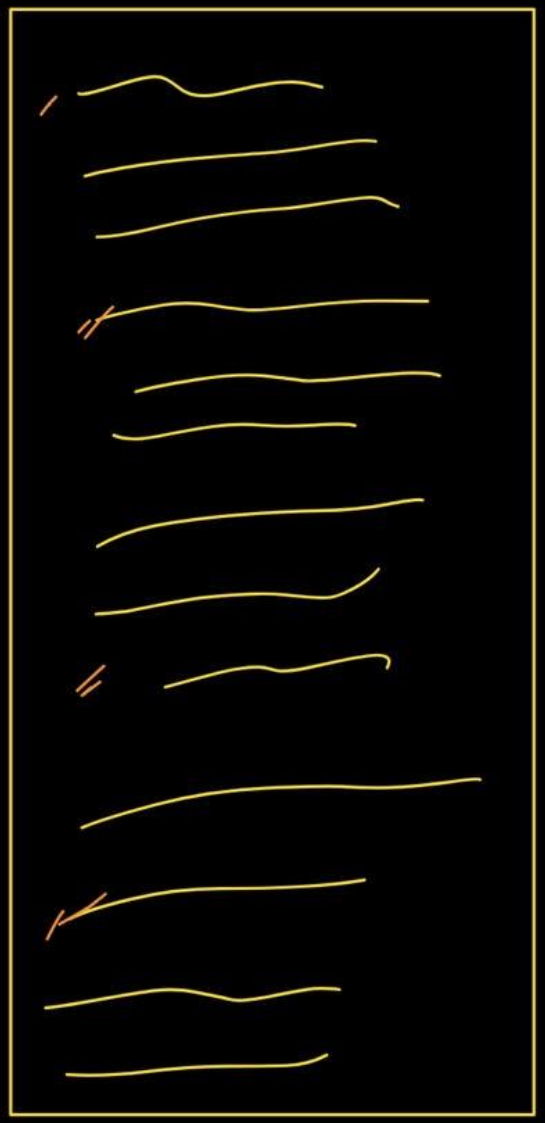
Your 1st Code:

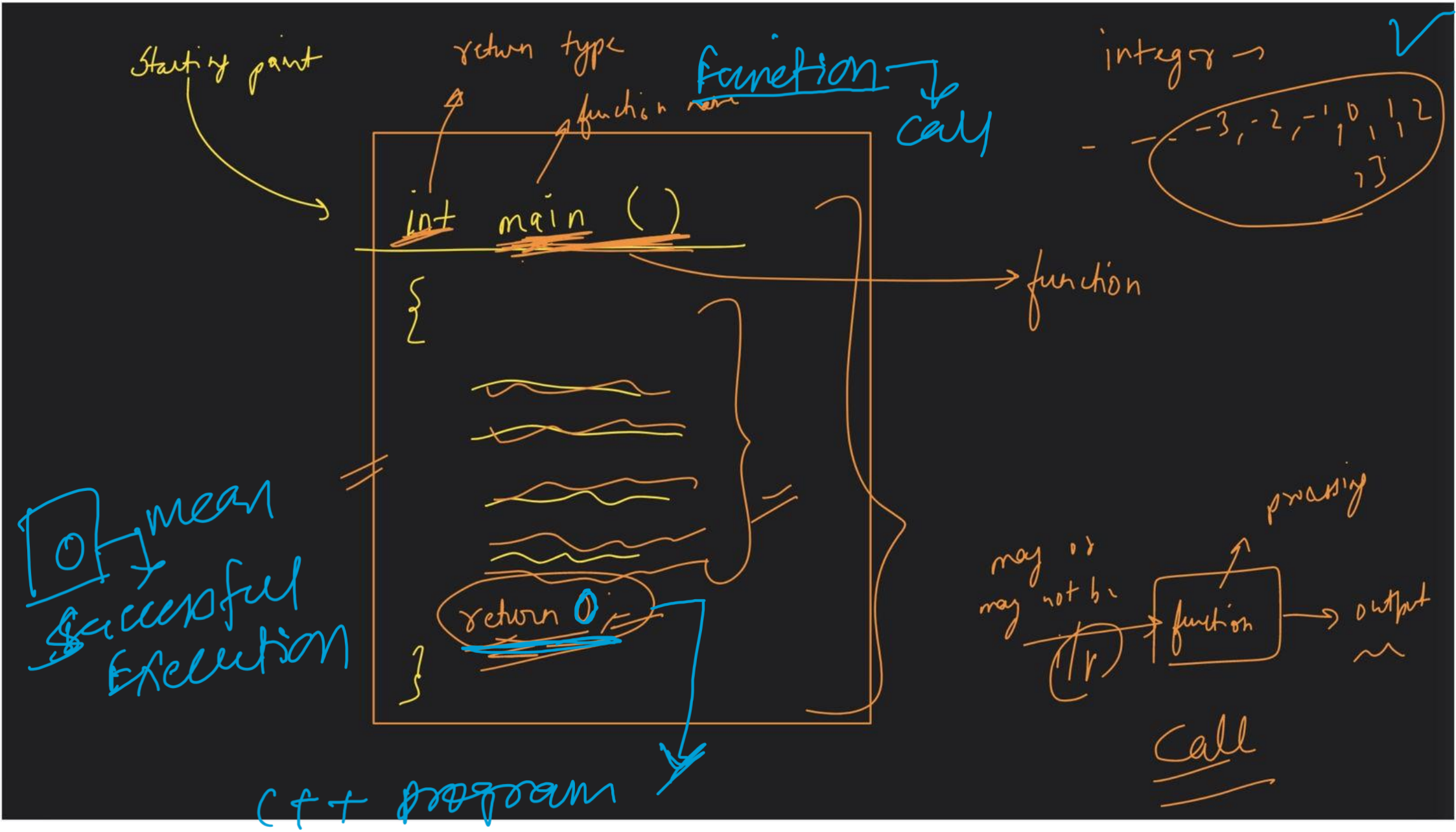
C++

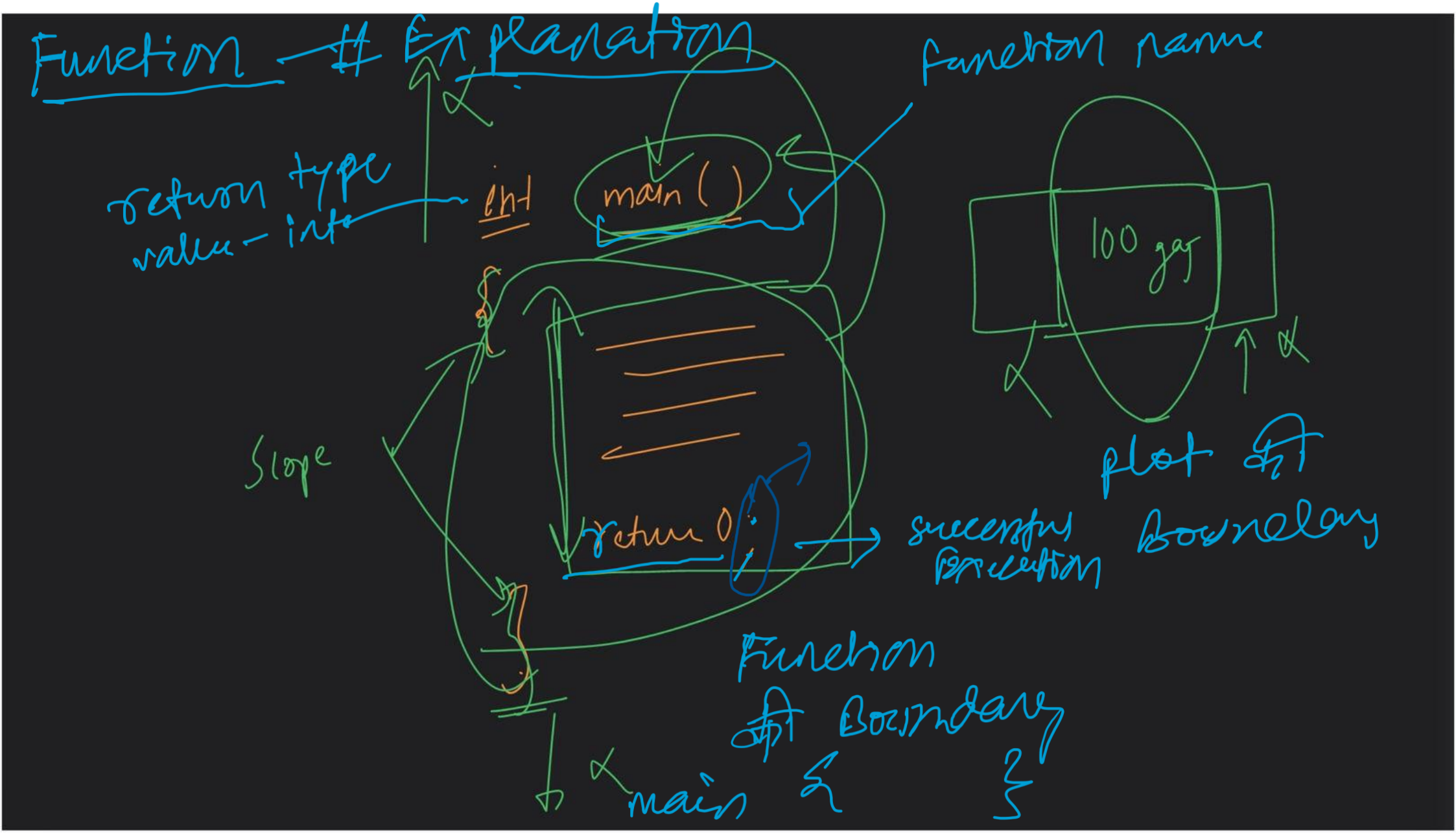
↳ starting point

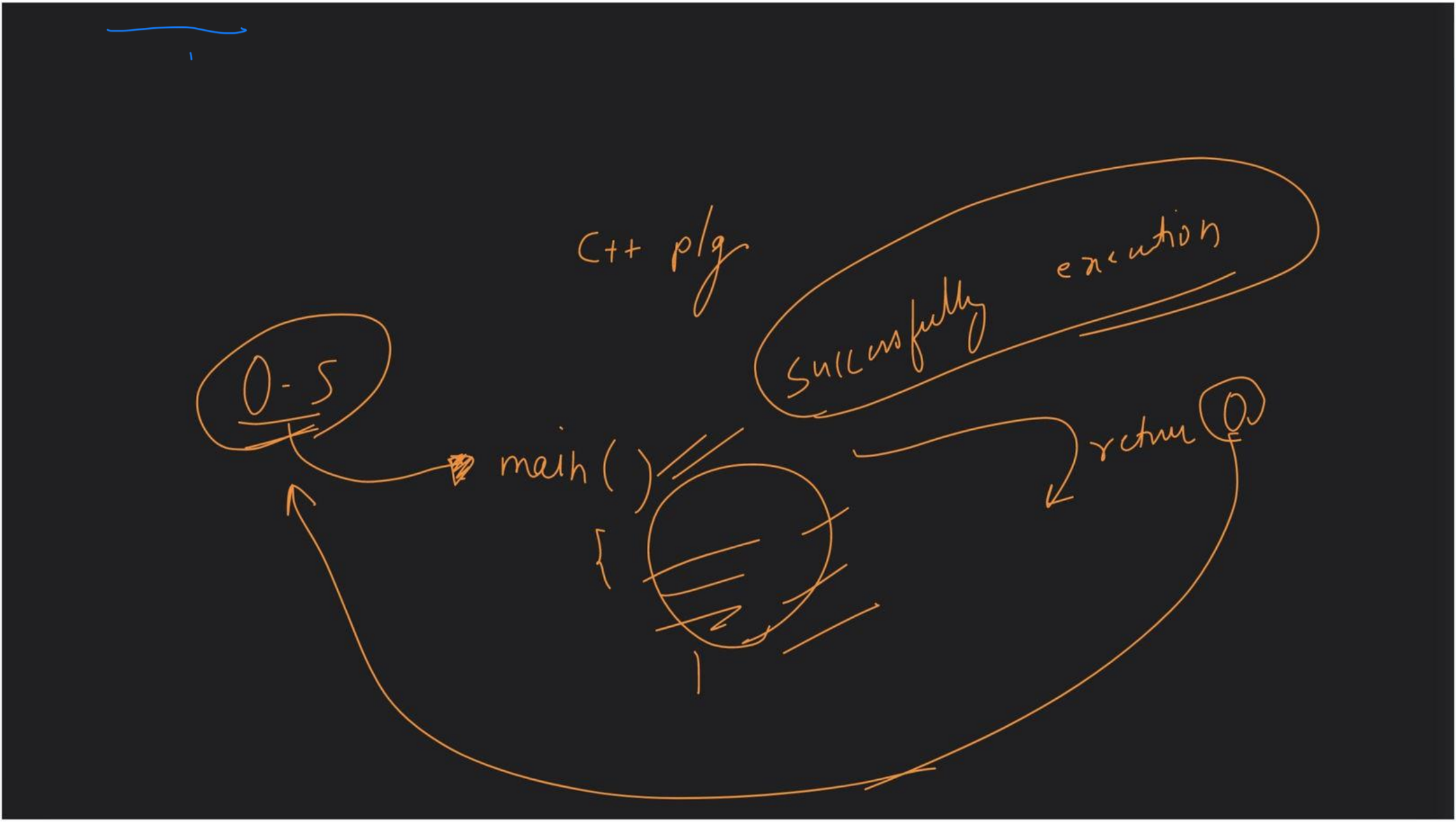
int main

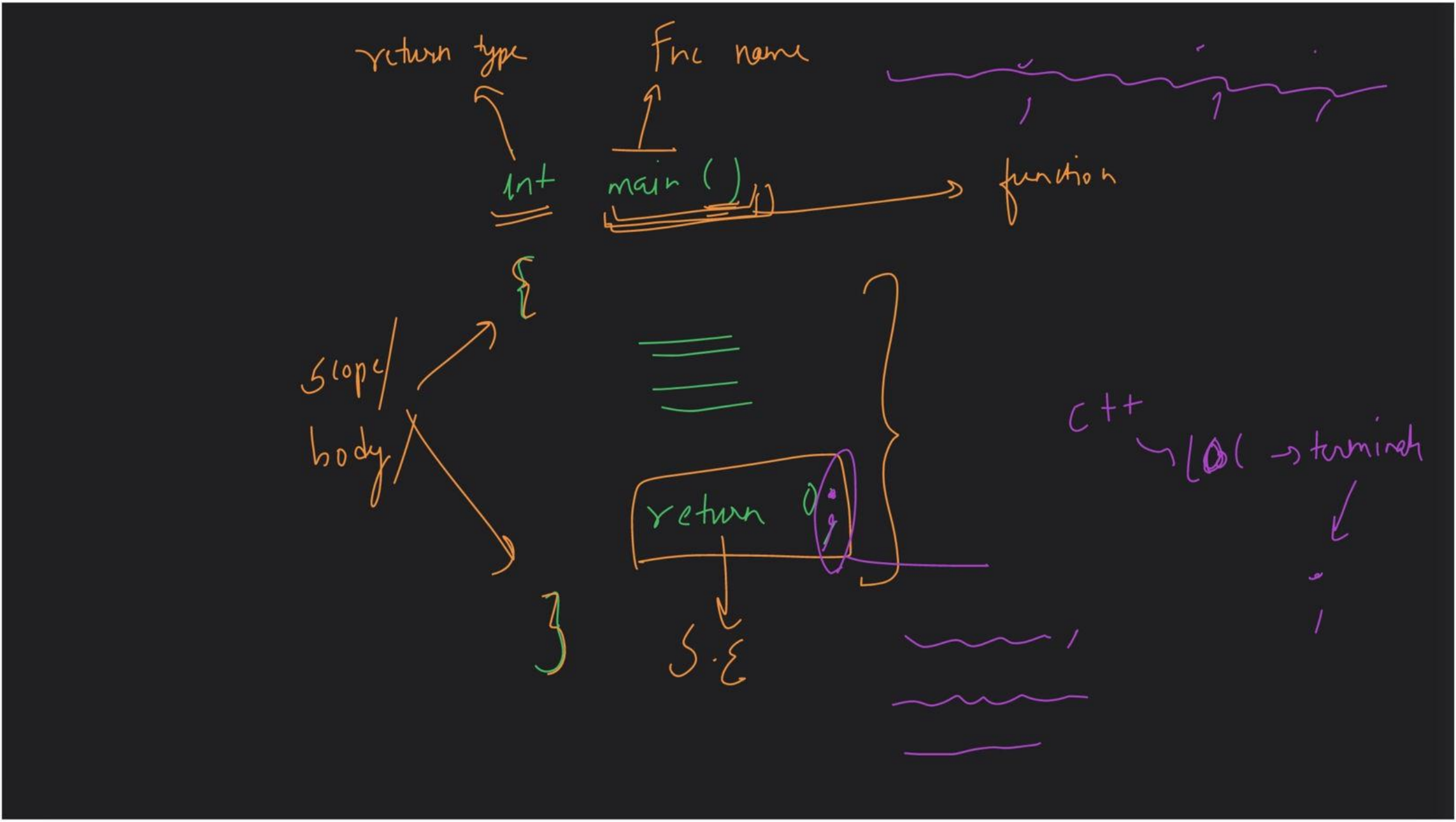
starting point

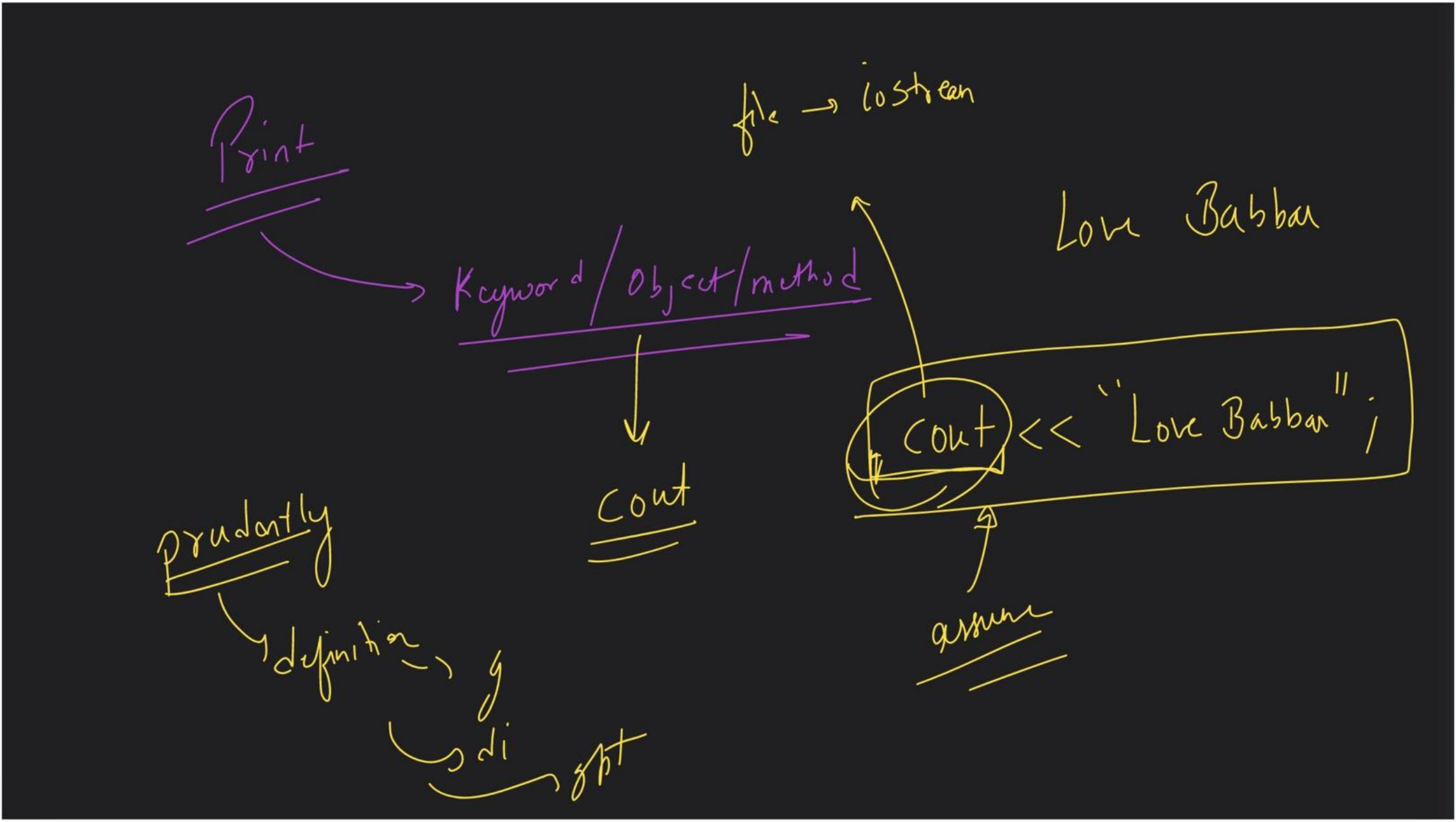


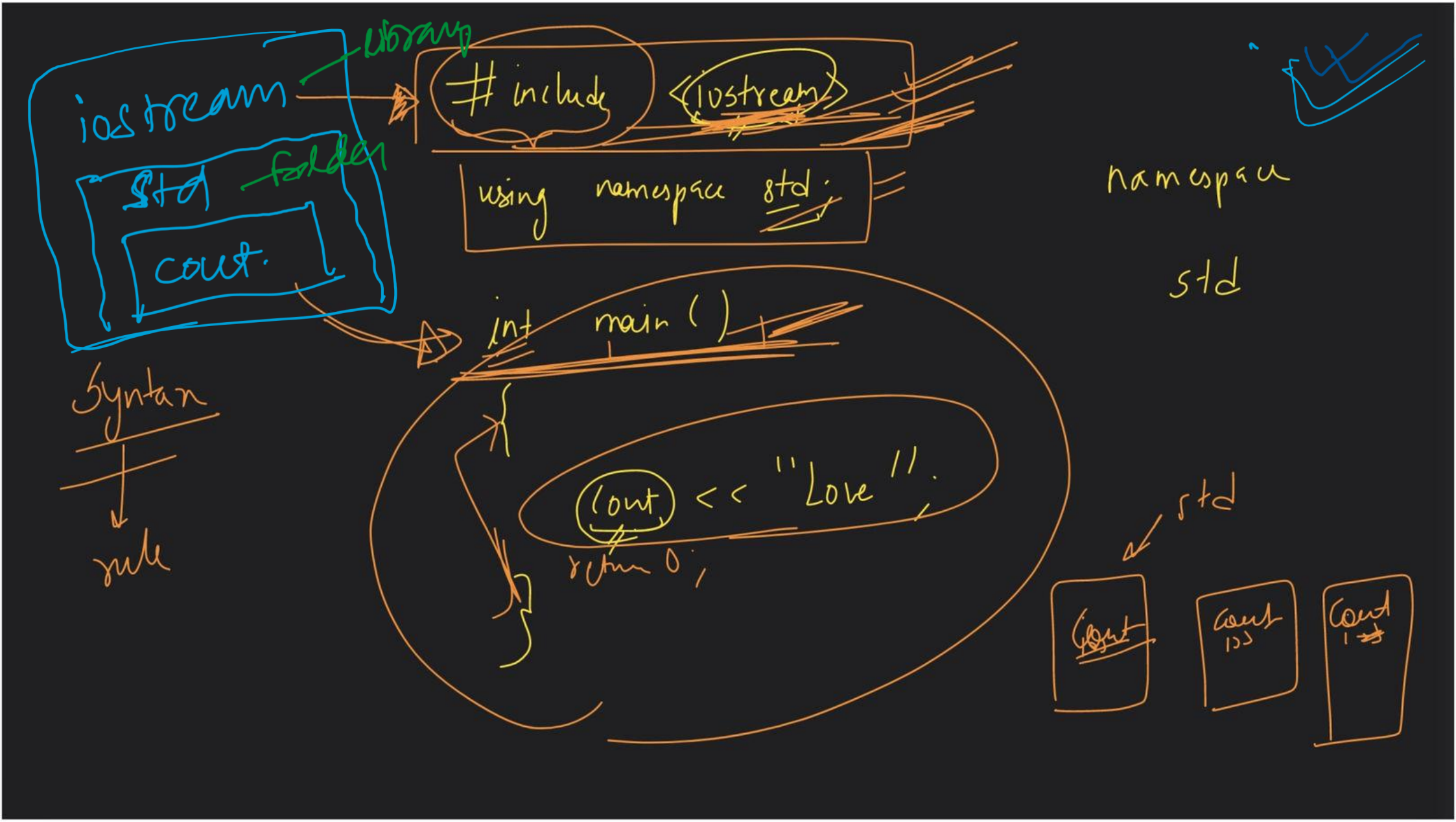

Starting point
return type
function name
function call
integers →
-3, -2, -1, 0, 1, 2
int main ()
{
function
return 0;
}
0 → mean successful Execution
C++ program
OS → main()
may or may not be
function
output
passing
Call


Function - # Explanation
function name
return type value - int
int
main ( )
100 gaj
Slope
plot of Boundary
return 0;
successful execution
Function of Boundary
main { }


C++ p/g
O.S
main( )
{
}
Successfully execution
return 0;

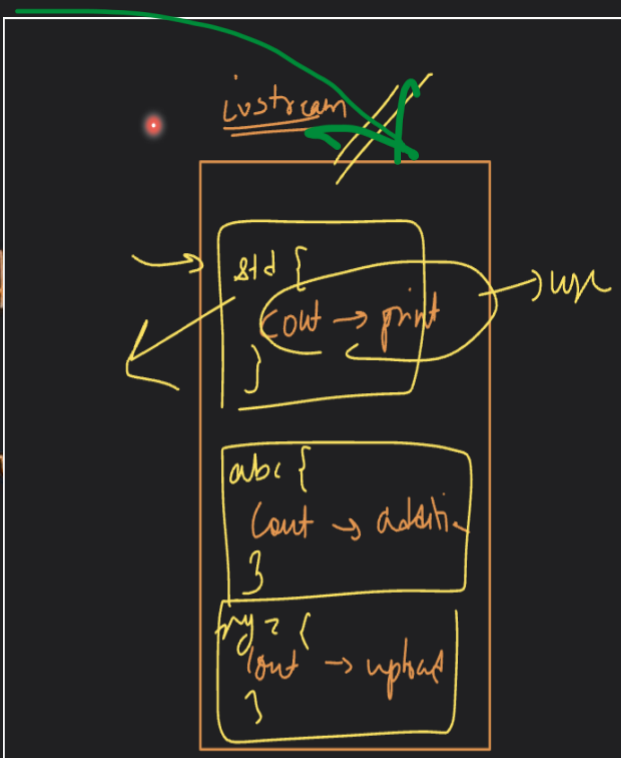

return type
Func name
int
main ()
function
{
scope/
body
return 0;
}
S.E
c++
\0(→ terminate


Print
file → iostream
Keyword / Object / method
Love Babbar
cout
cout << "Love Babbar" ;
Prudantly
definition
gpt
assume

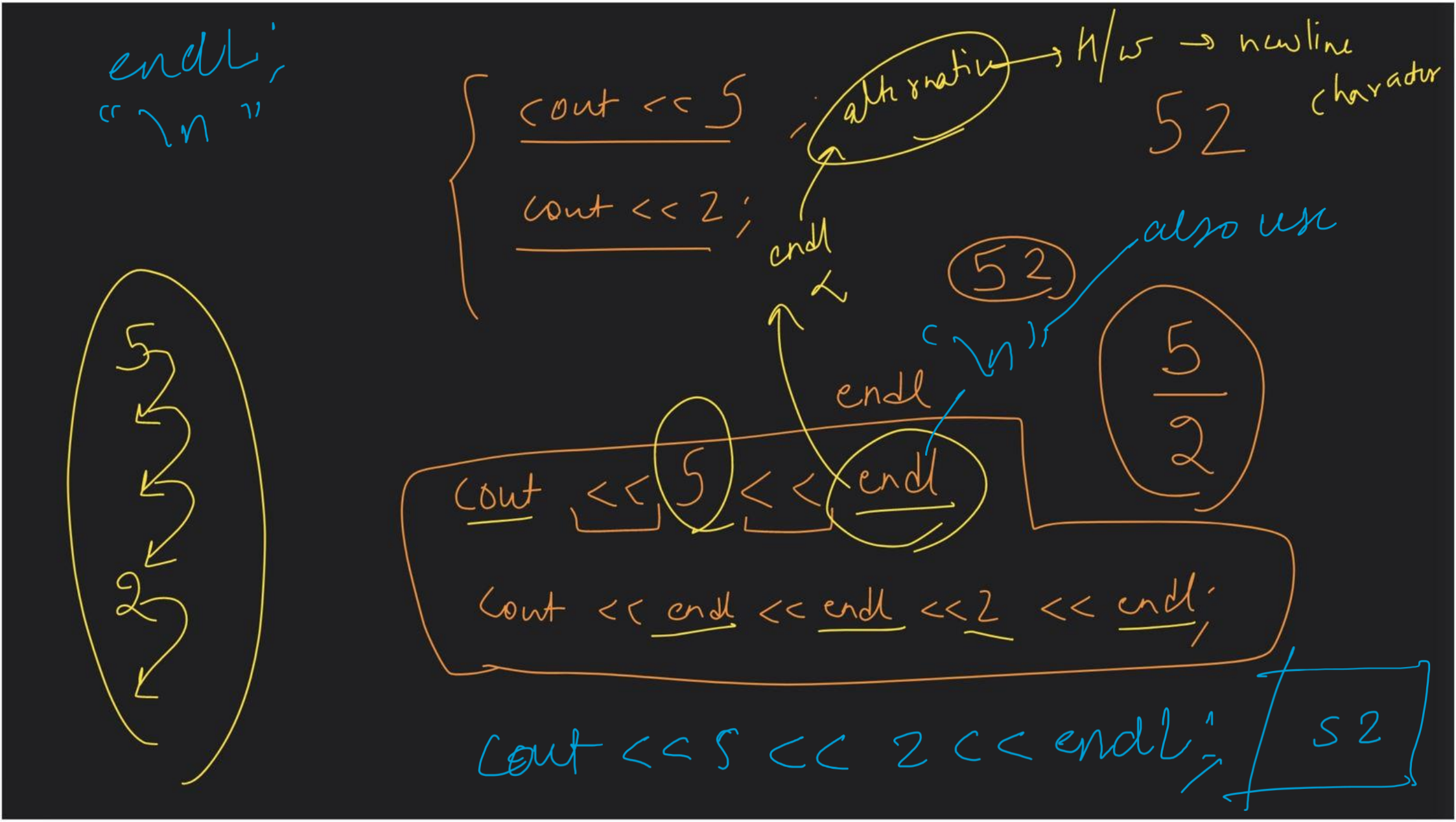

iostream
library
std
folder
cout.
#include
<iostream>
using namespace std;
int main()
{
cout << "Love";
return 0;
}
Syntax
rule
namespace
std
std
cout
cout
cout

iostream

folder
folder
cout

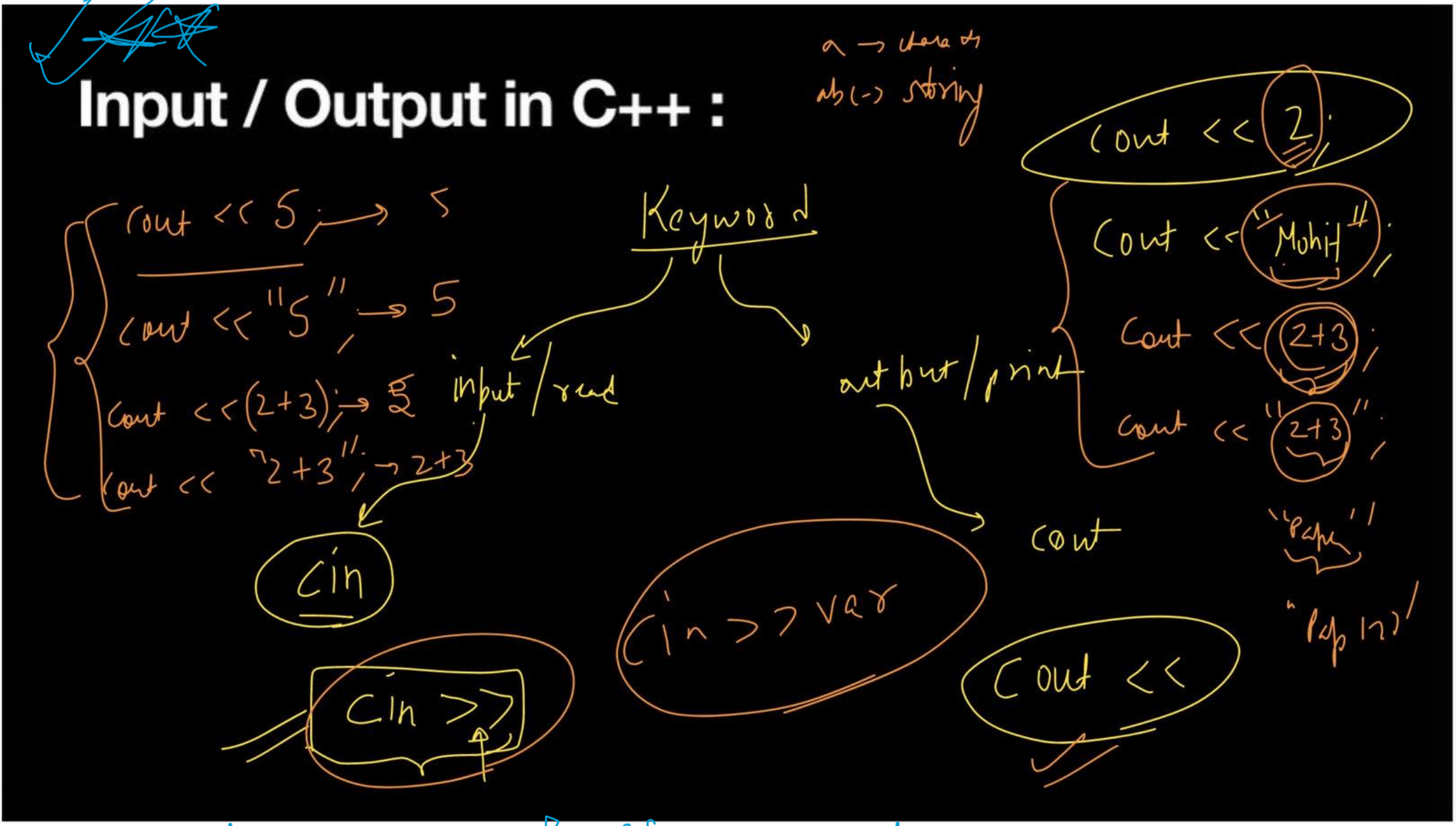



```
#include <iostream>
using namespace std;

int main()
{
    cout << "Love";

    return 0;
}
```

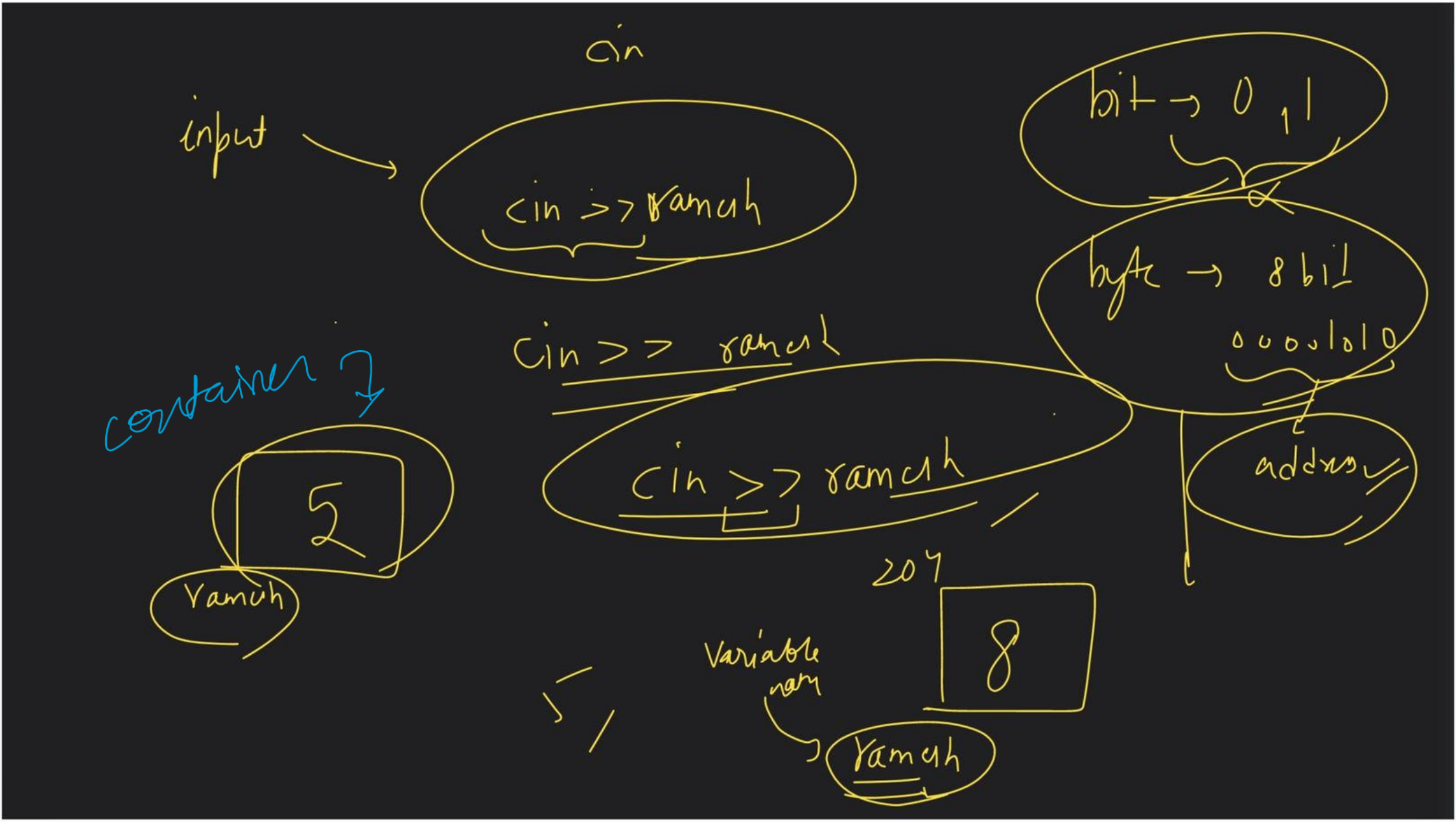

endl;
"\n"
cout << 5;
cout << 2;
alternative → H/w → newline character
52
endl
also use
"\n"
5
2
5
2
endl
cout << 5 << endl
cout << endl << endl << 2 << endl;
cout << 5 << 2 << endl;
52

cout << "line 3 \n";

cout << "Line 4 \n"; }

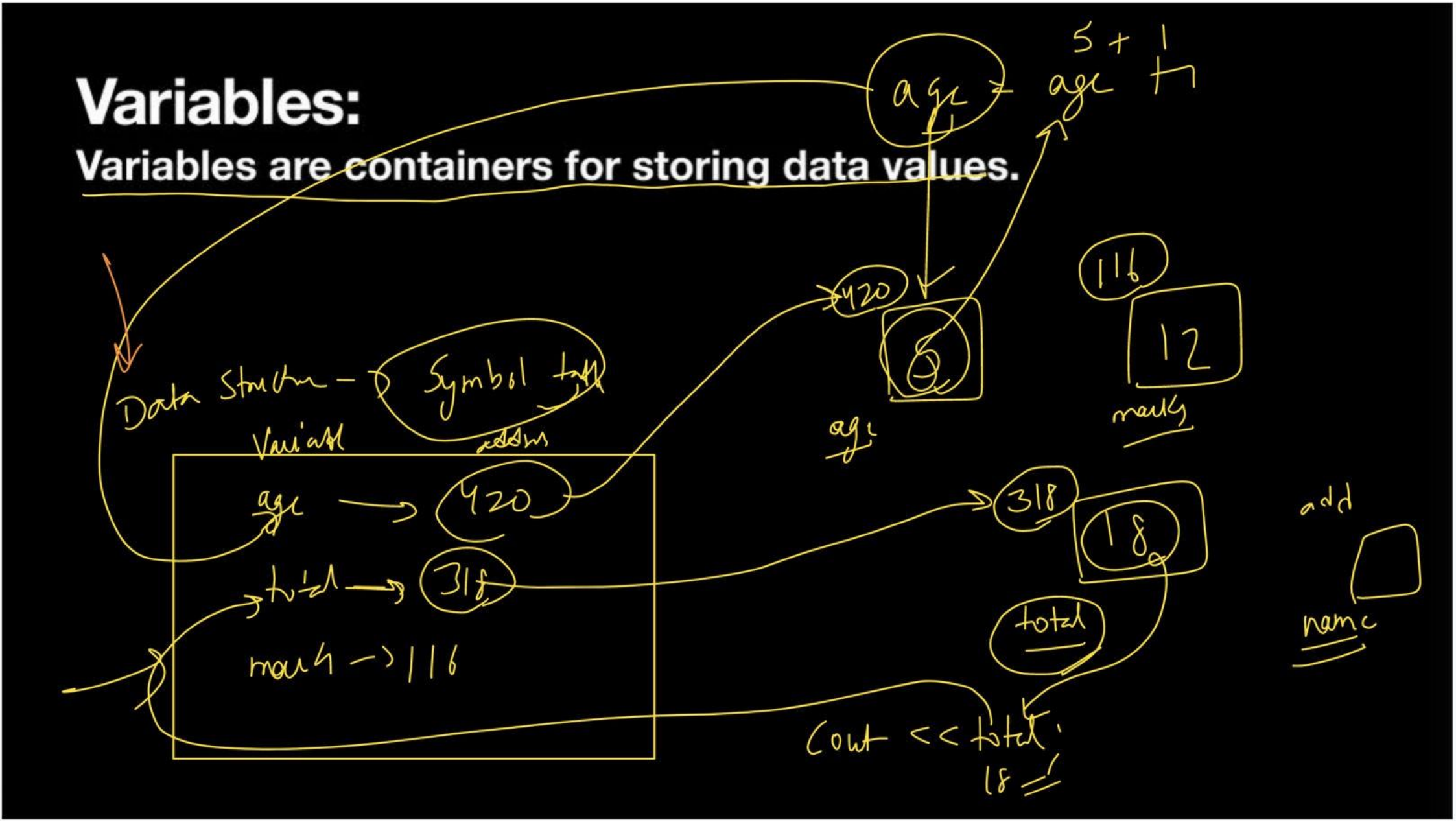


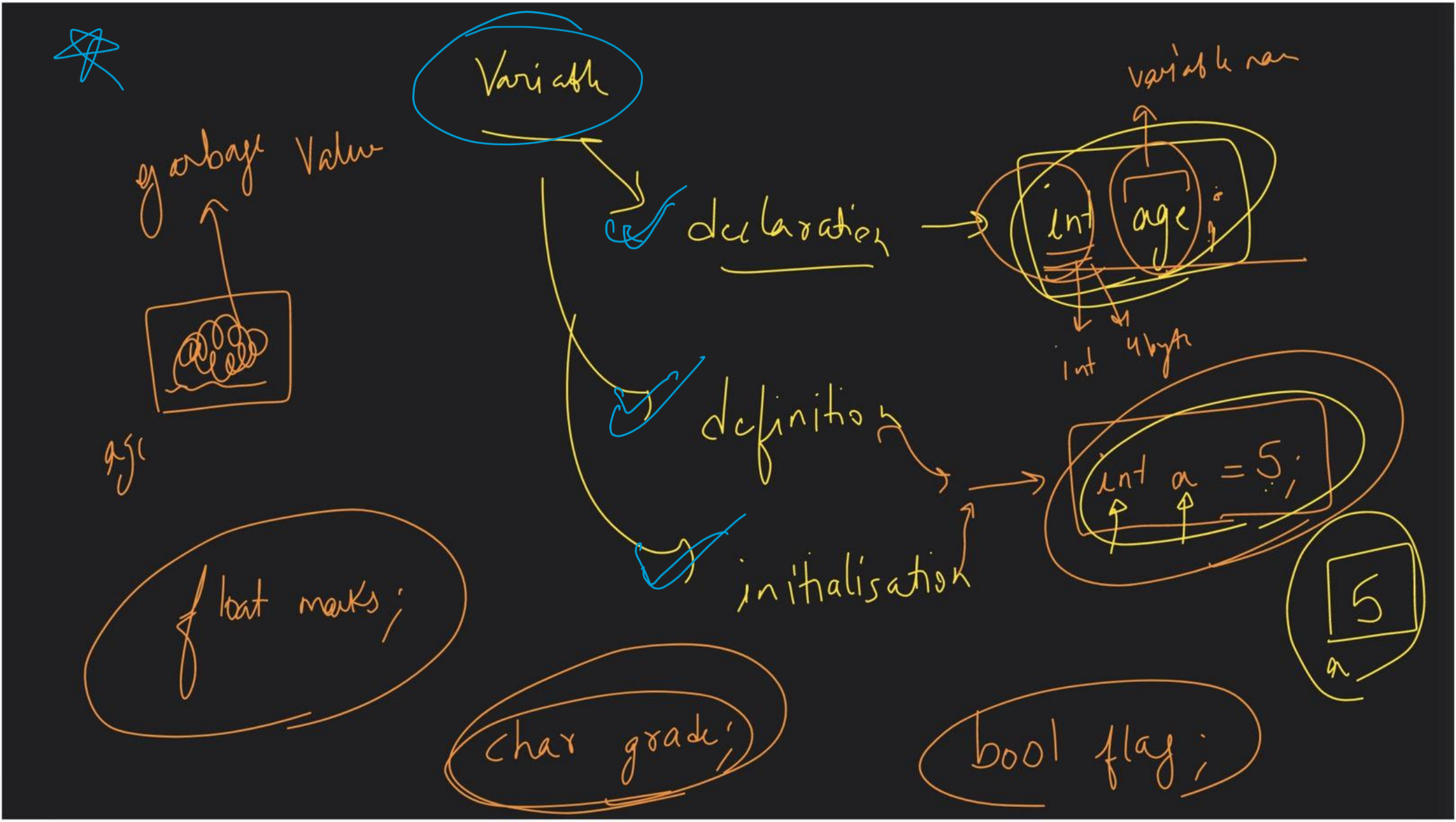
endl → end of line cout

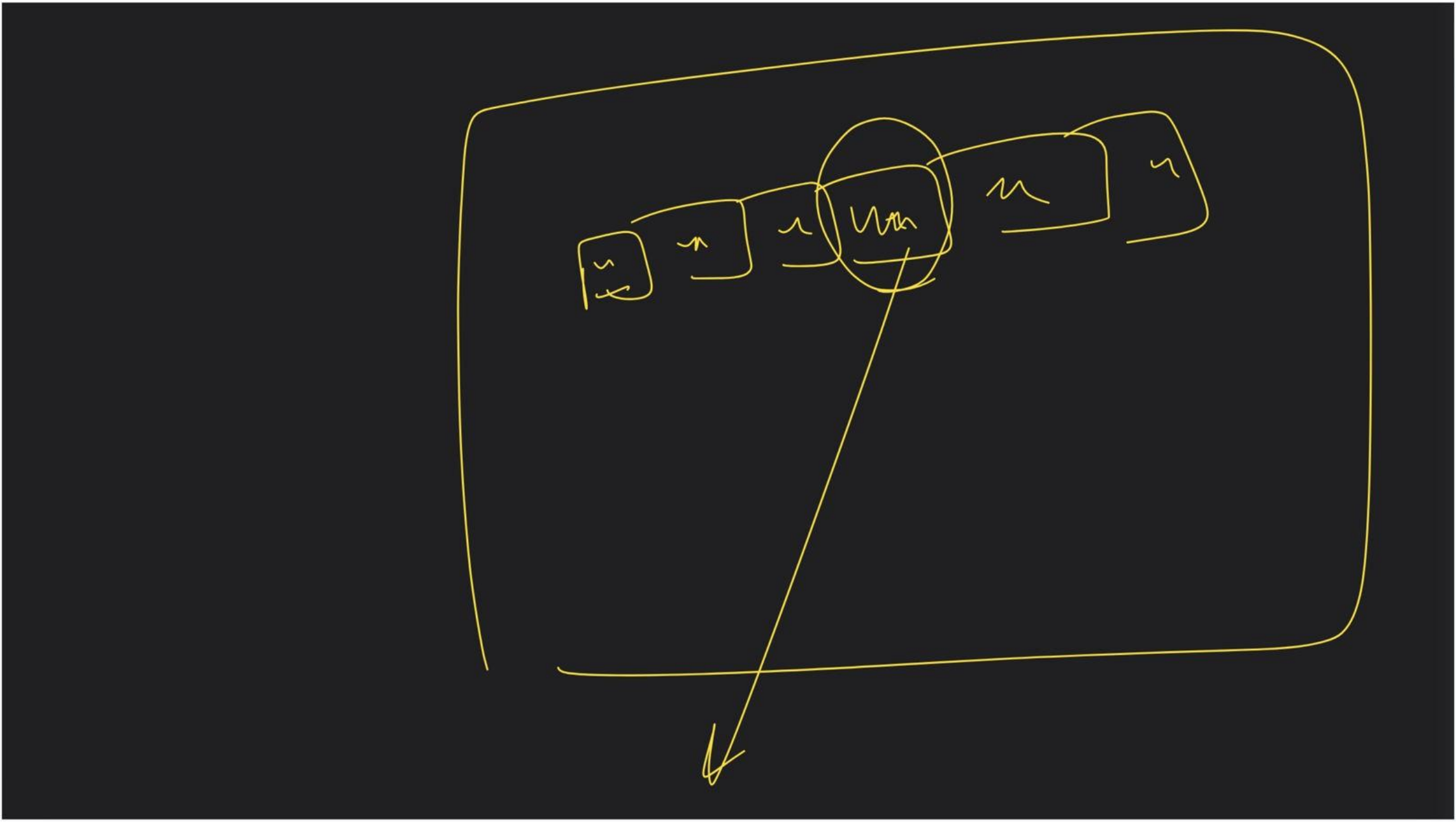

cin
input
cin >> ranch
bit → 0 , 1
byte → 8 bit
0 0 0 0 1 0 1 0
address
container 7
5
ranch
cin >> ranch
cin >> ranch
207
8
Variable name
ranch

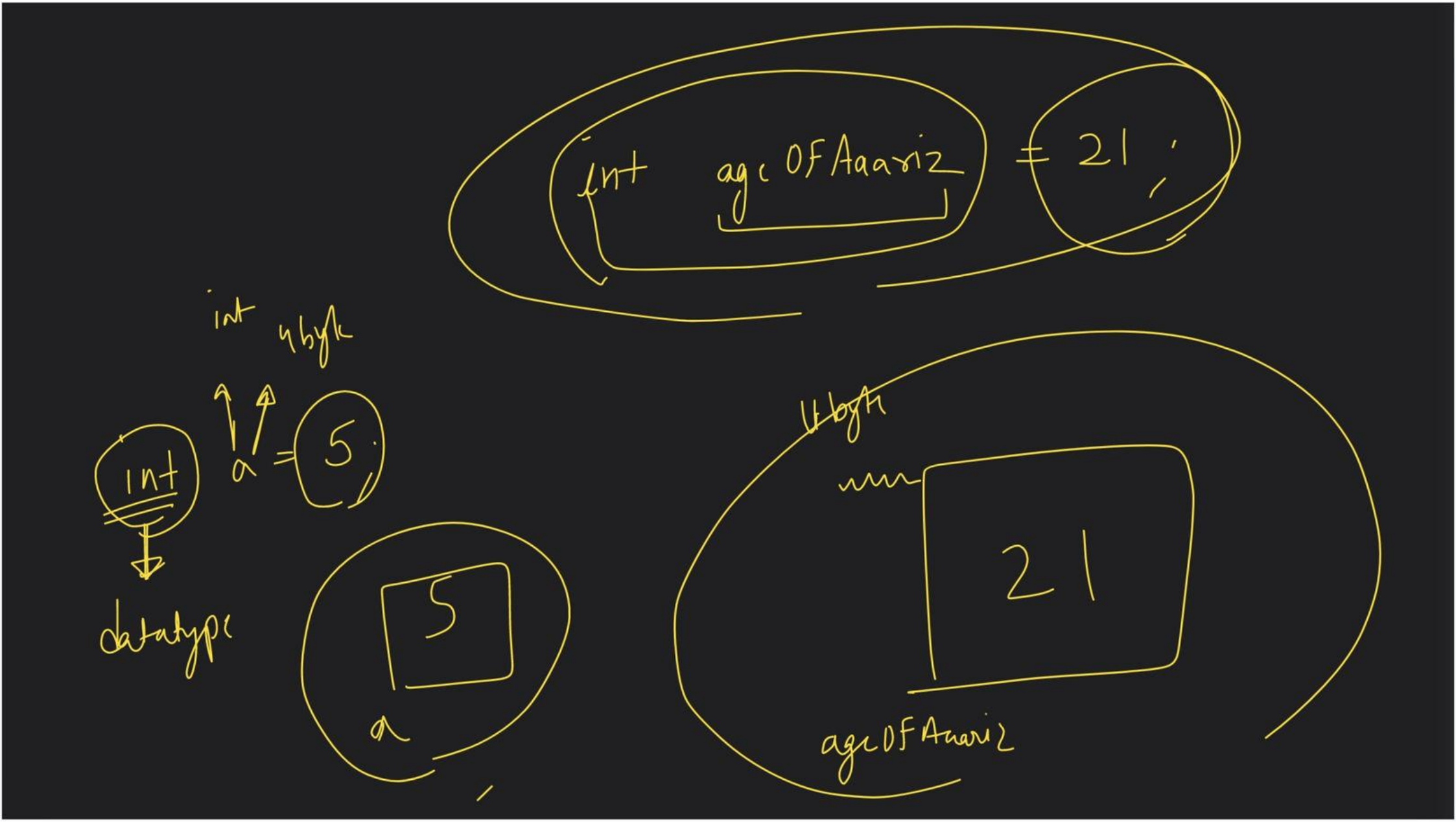
Variables:

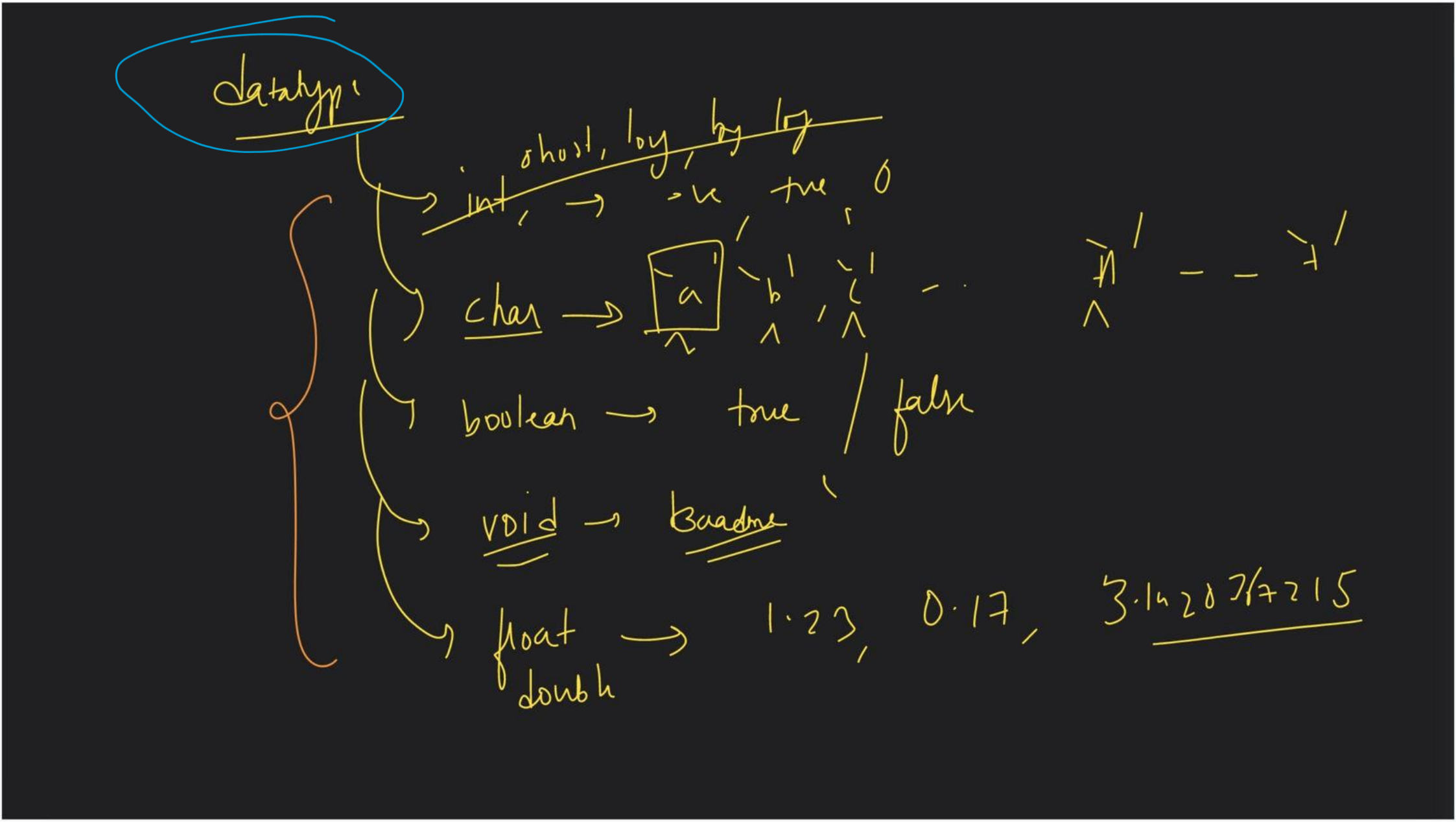
Variables are containers for storing data values.

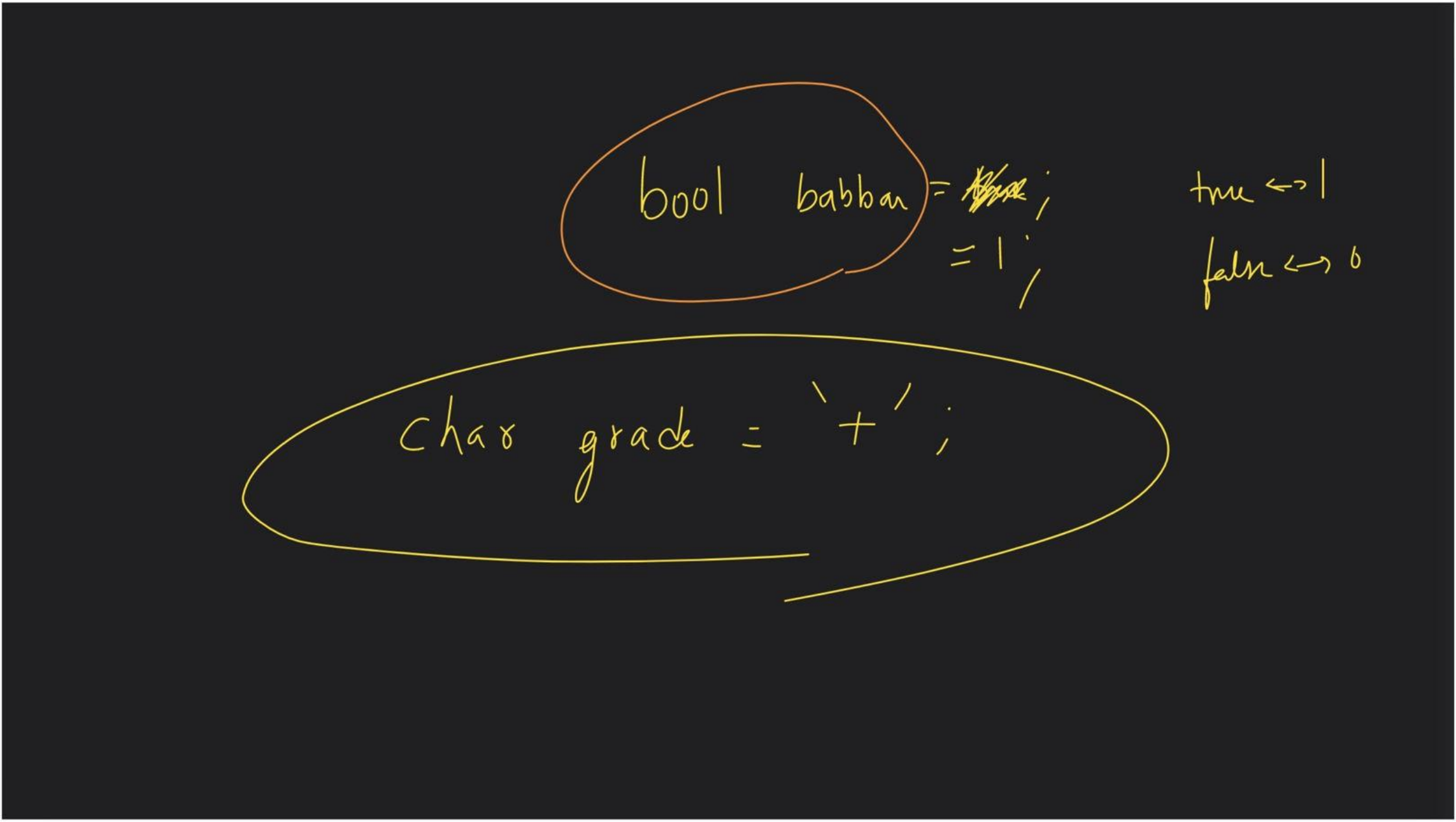



Variable
garbage Value
age
declaration
definition
initialisation
variable name
int age;
int
4 byte
int a = 5;
5
a
float marks;
char grade;
bool flag;




int ageOfAaariz = 21;
int
4byte
int
a = 5;
datatype
5
a
4byte
21
ageOfAaariz

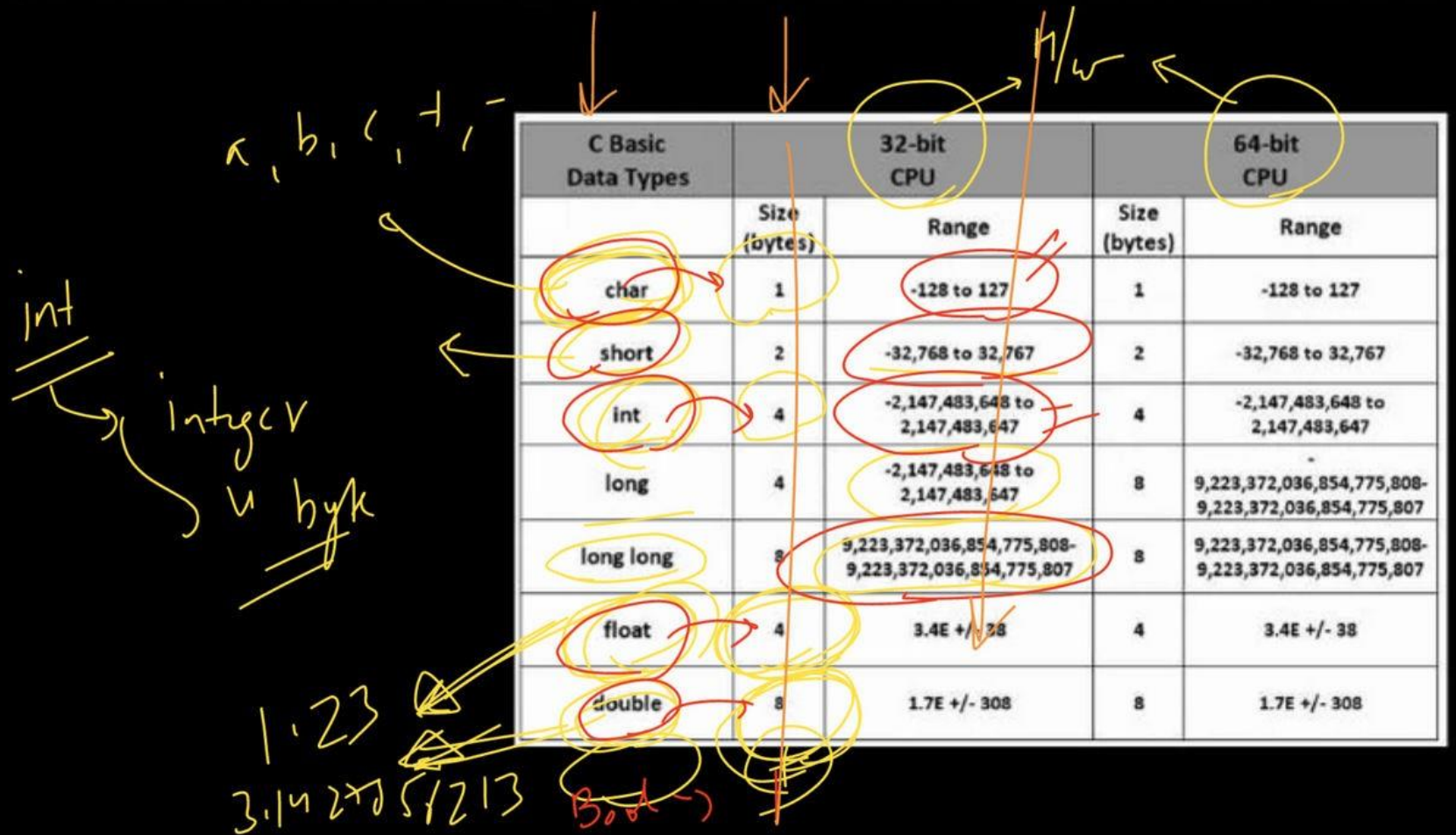

Datatype:
int → -ve true 0
char → a b
boolean → true / false
void → Baadme
float → 1.23, 0.17, 3.14207/7215
double


bool babban = ;
= 1;
true ↔ 1
false ↔ 0
char grade = '+';

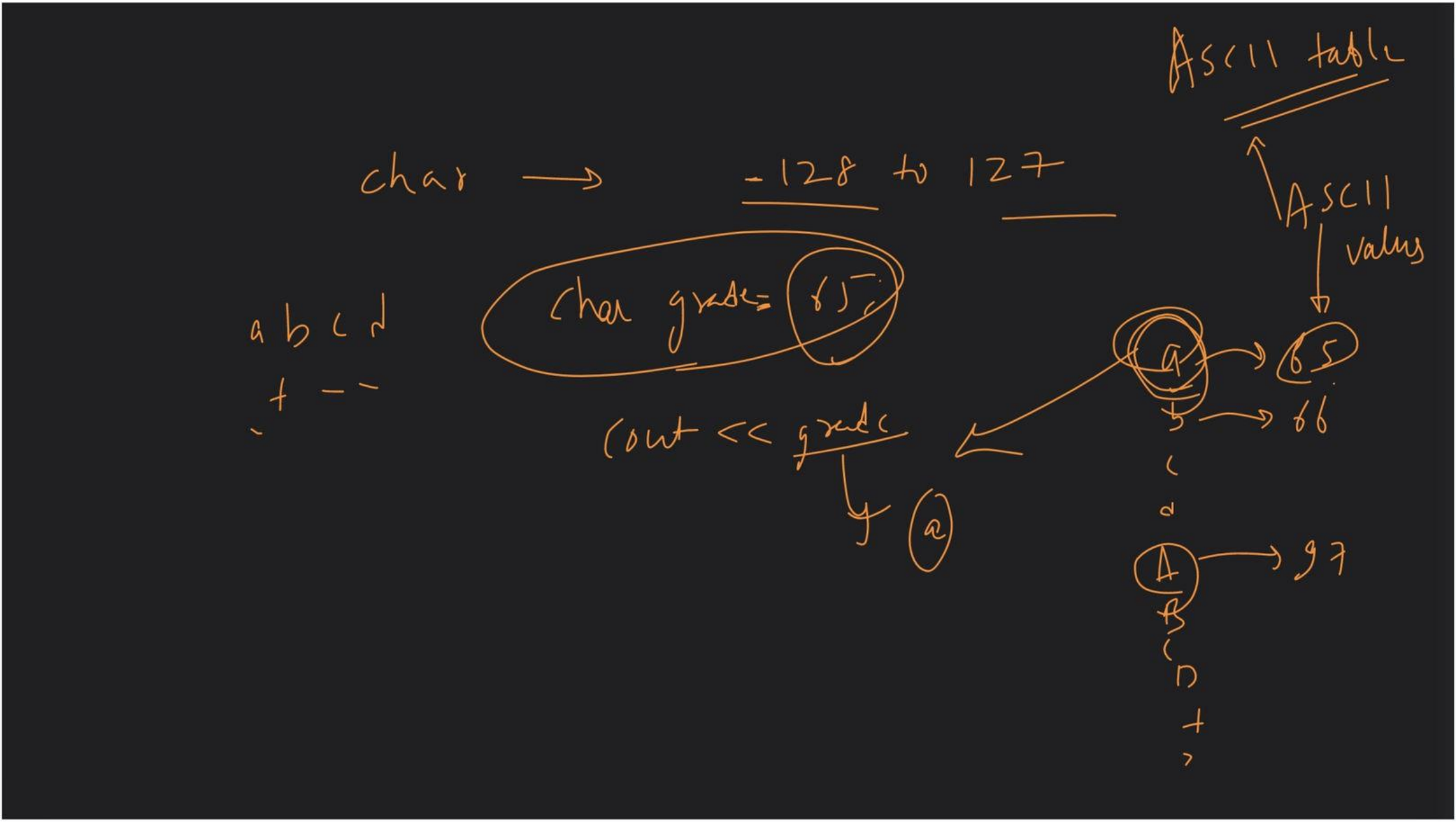
Datatypes:

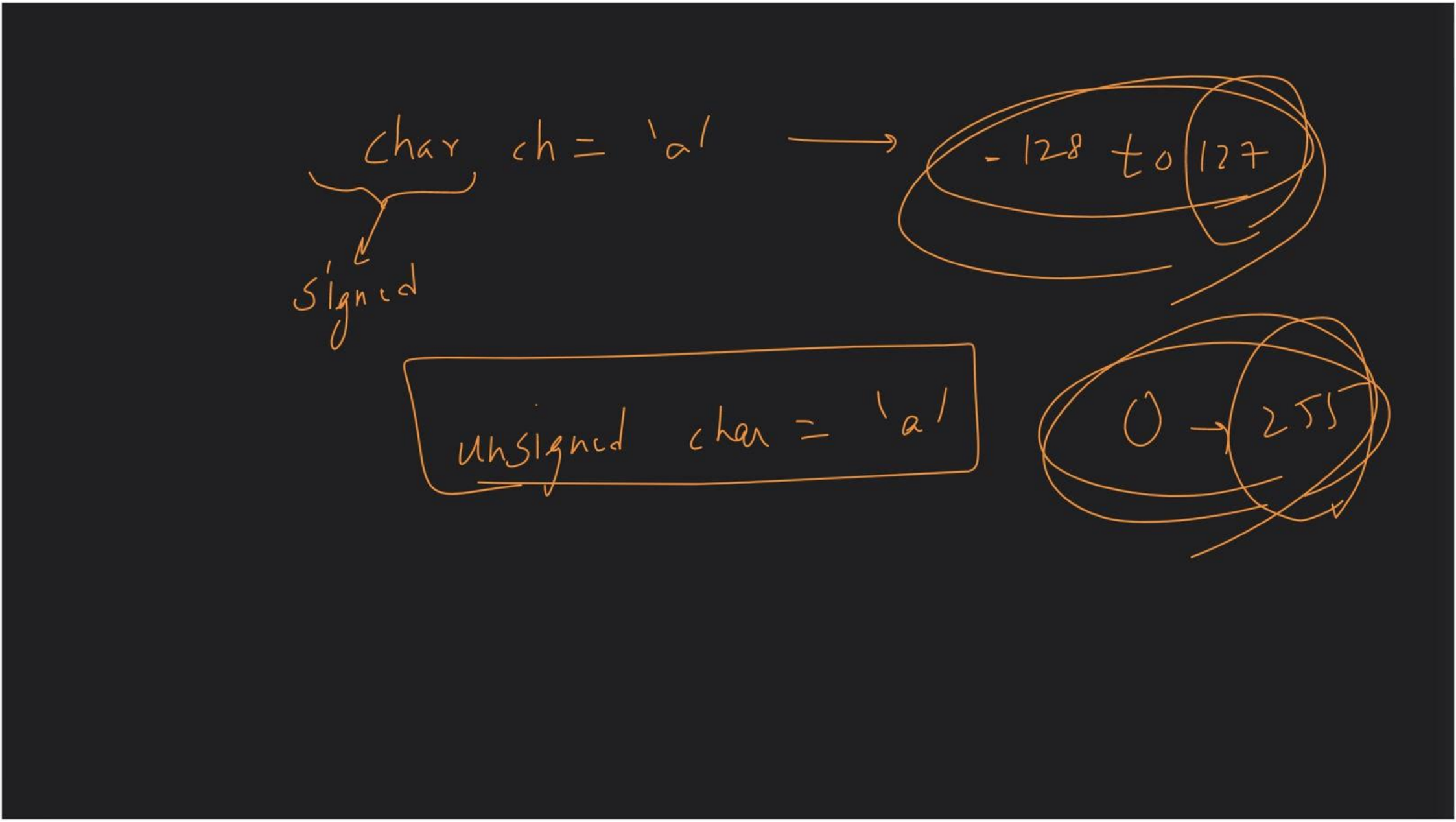
The data type specifies the size and type of information the variable will store

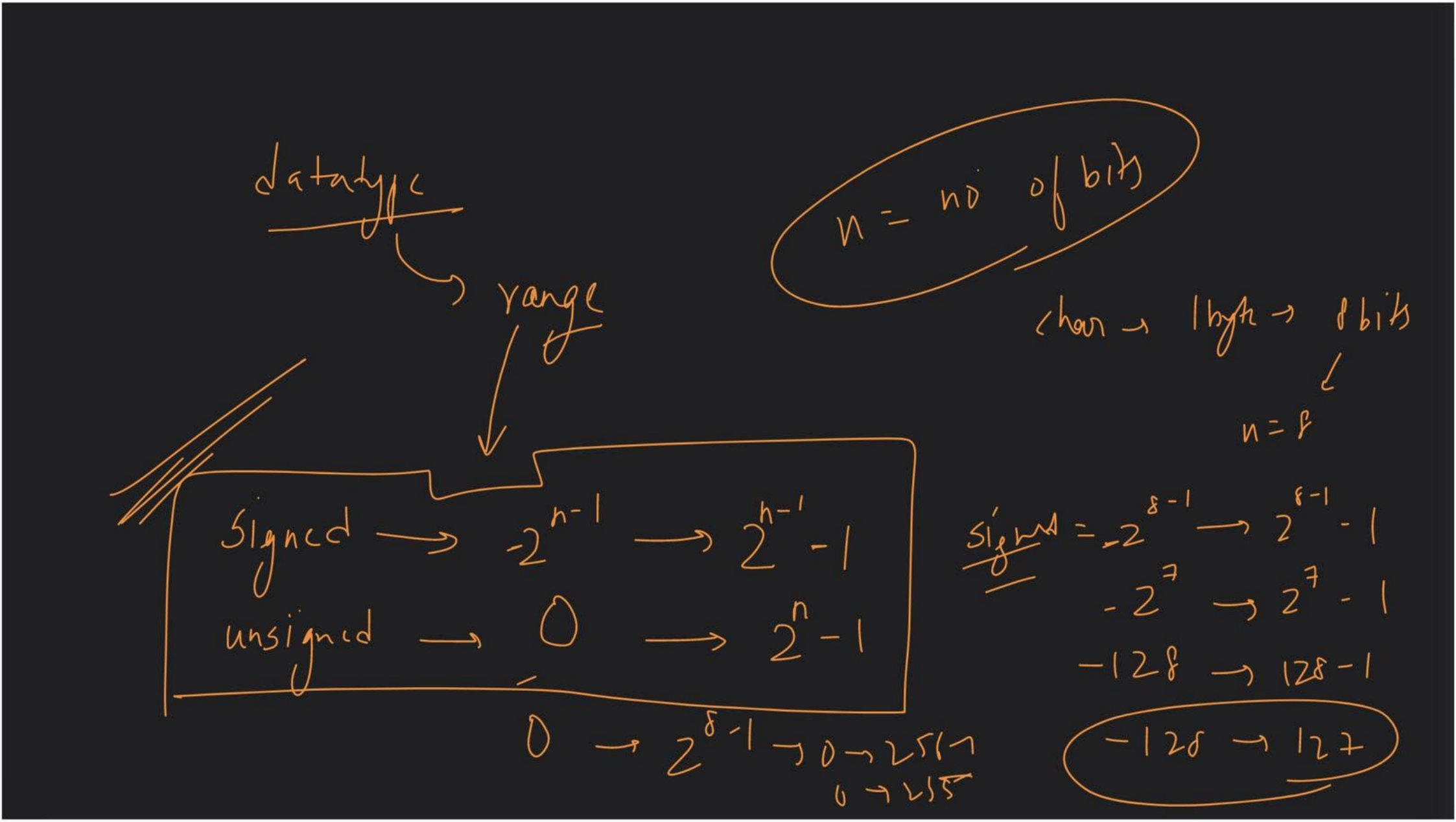
- Diagram



C Basic 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	-32,768 to 32,767
int	4	-2,147,483,648 to 2,147,483,647	4	-2,147,483,648 to 2,147,483,647
long	4	-2,147,483,648 to 2,147,483,647	8	-9,223,372,036,854,775,808-9,223,372,036,854,775,807
long long	8	9,223,372,036,854,775,808-9,223,372,036,854,775,807	8	9,223,372,036,854,775,808-9,223,372,036,854,775,807
float	4	3.4E +/- 38	4	3.4E +/- 38
double	8	1.7E +/- 308	8	1.7E +/- 308


Ascii table
char → -128 to 127
Ascii values
char grade = 85;
a b c d
cout << grade;
65
66
97
a
@


char ch = 'a'
signed
-128 to 127
unsigned char = 'a'
0 → 255

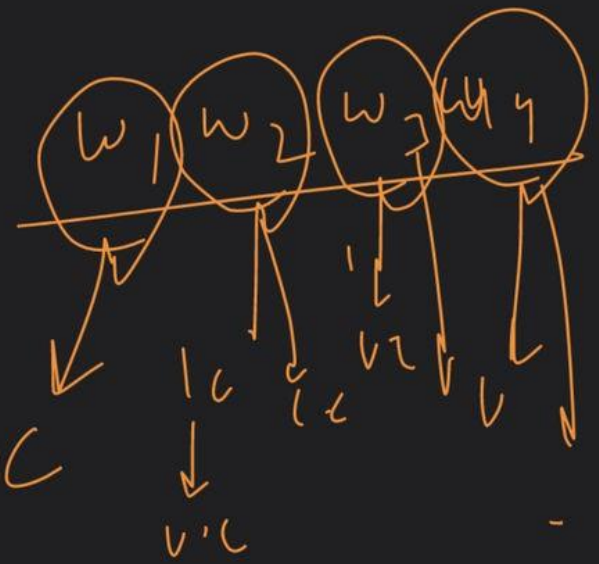

datatype
range
n = no of bits
char → 1byte → 8bits
n = 8
Signed → -2^{n-1} → $2^{n-1}-1$
unsigned → 0 → 2^n-1
Signed = -2^{8-1} → $2^{8-1}-1$
-2^7 → 2^7-1
-128 → $128-1$
-128 → 127
0 → 2^{8-1} → 0 → 255
0 → 255

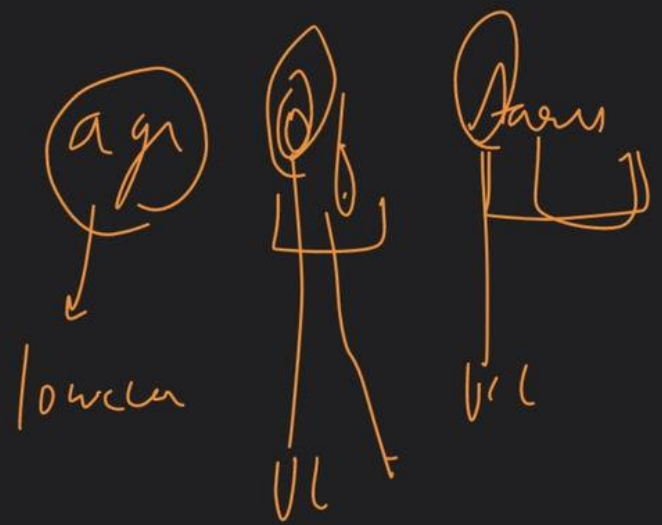
Variable Naming Conventions:

Naming Conventions rules for Variables are:

- It should begin with an alphabet.
- There may be more than one alphabet, but without any spaces between them.
- Digits may be used but only after alphabet.
- No special symbol can be used except the underscore (_) symbol. When multiple words are needed, an underscore should separate them.
- No keywords or command can be used as a variable name.
- All statements in C++ language are case sensitive. Thus a variable A (in uppercase) is considered different from a variable declared a (in lowercase).

```
int love=5;
int Love=5;
```





love_babbar

age Of Aaariz

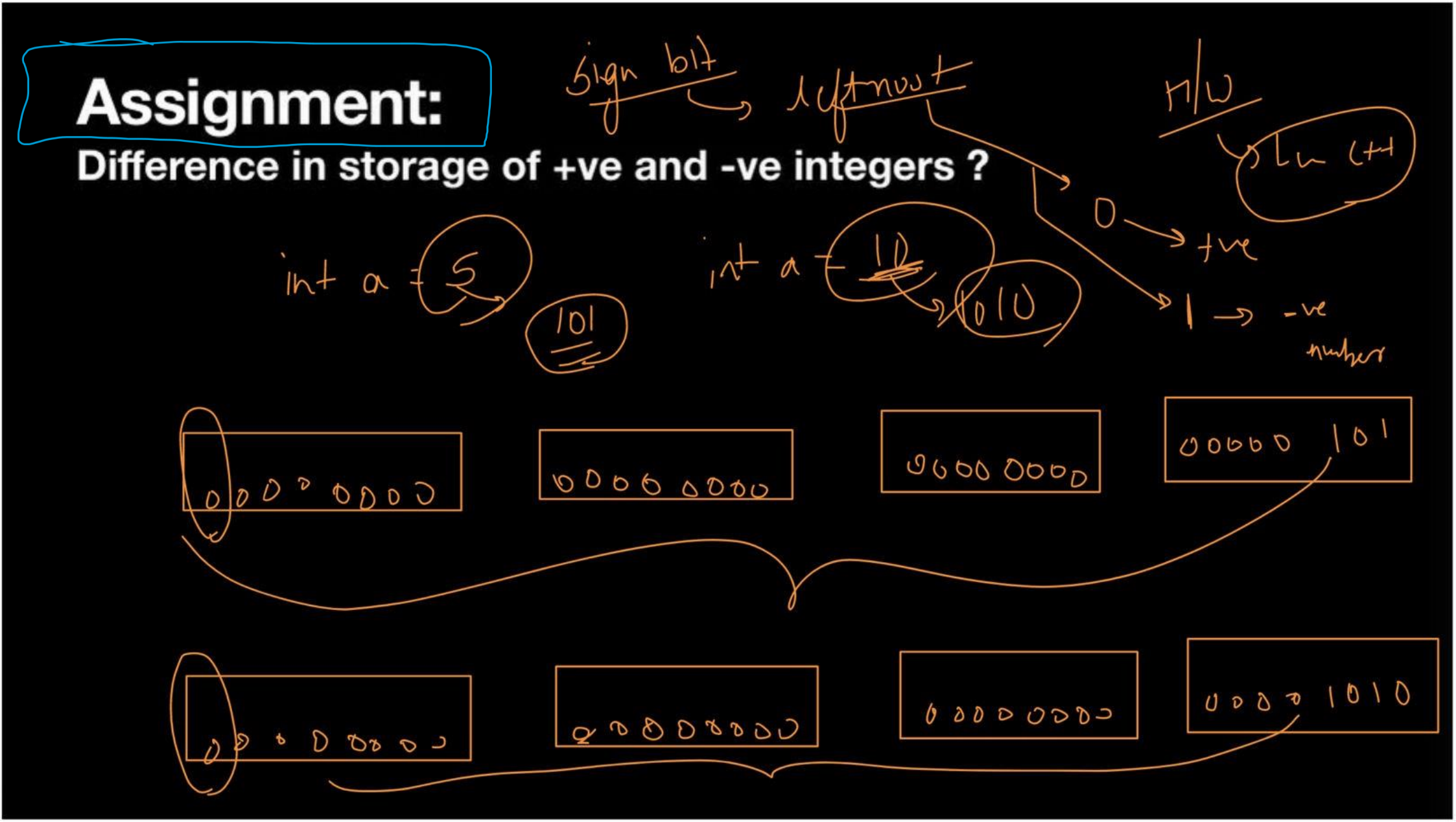
loveBabbar

harsh Kapoor

adityaKumar

Assignment:

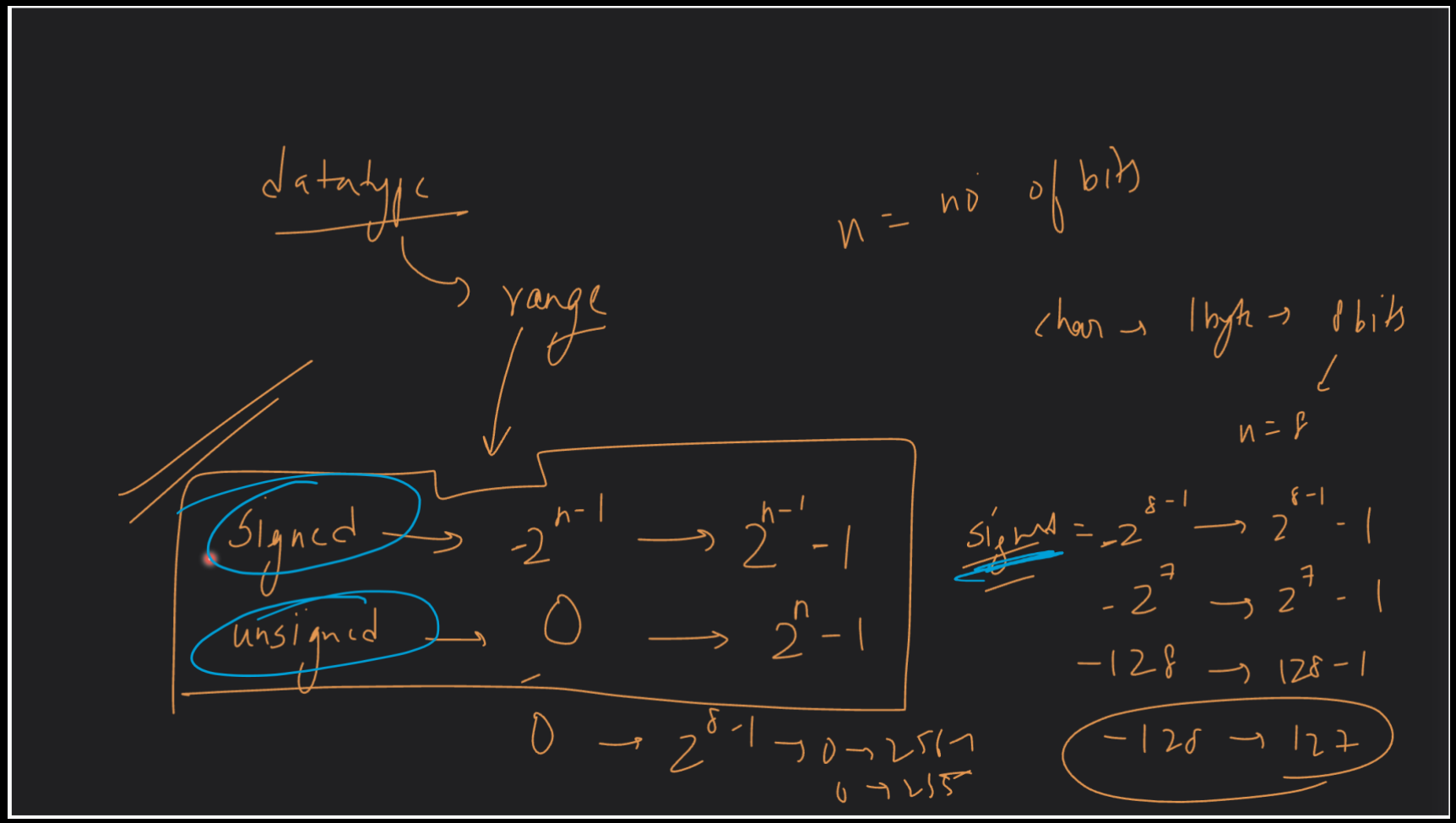
Difference in storage of +ve and -ve integers ?



1's complement

2's complement

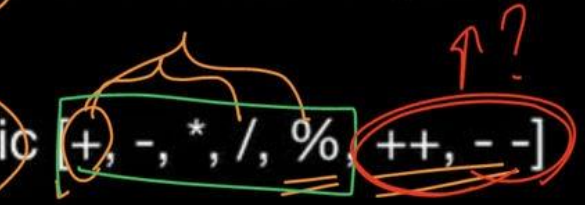
Signed vs Unsigned Integers:

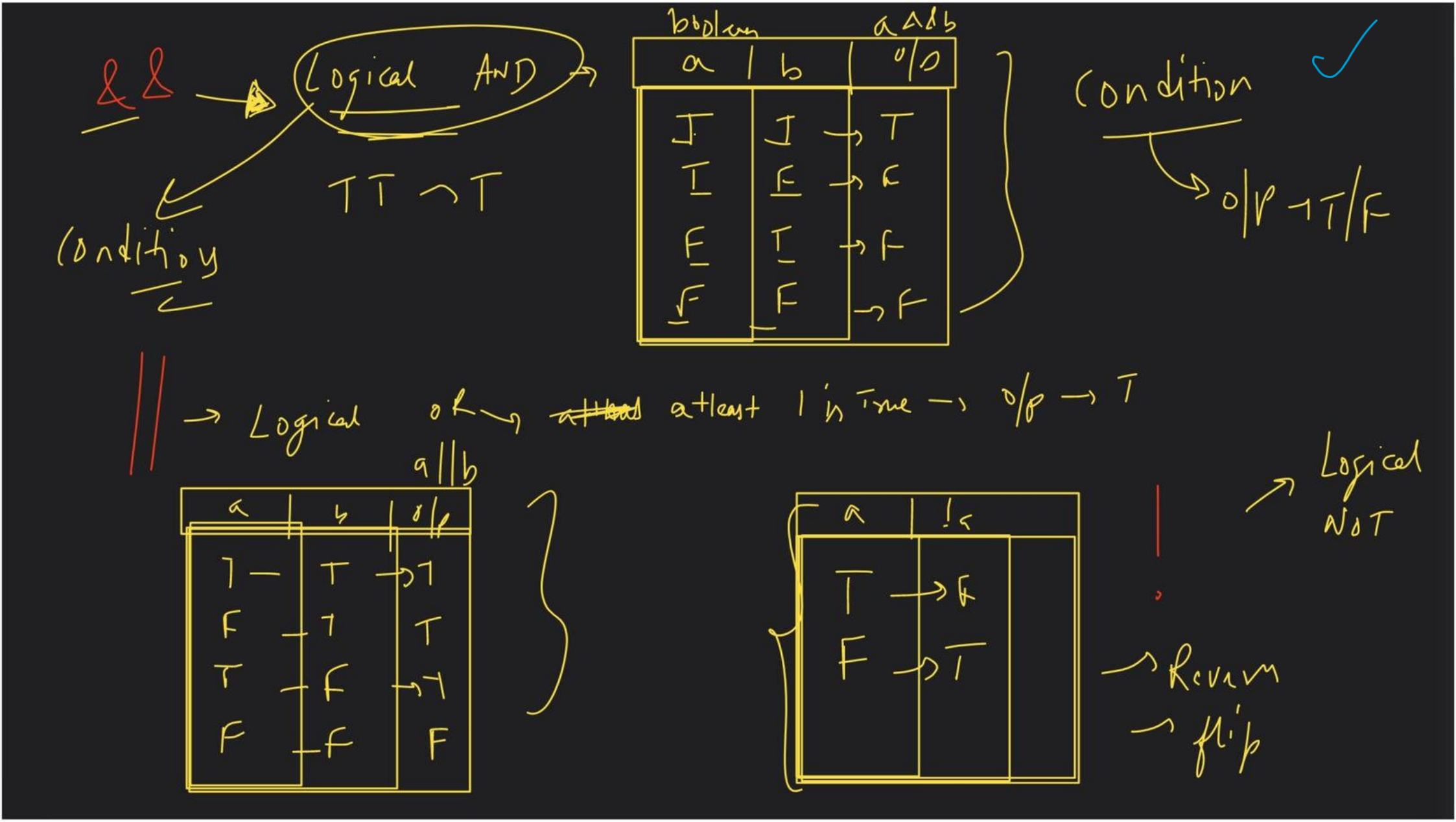


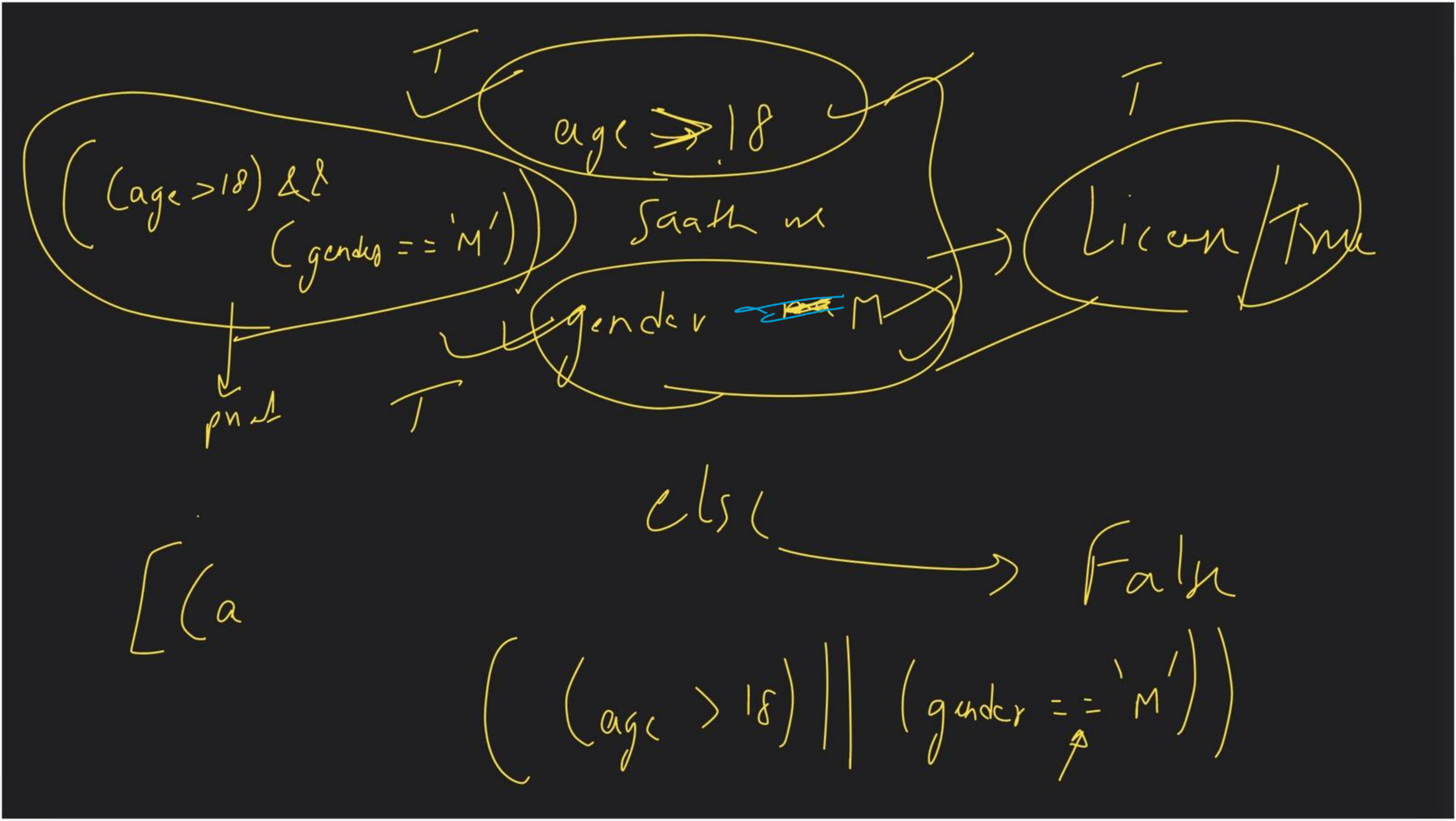
Operators:

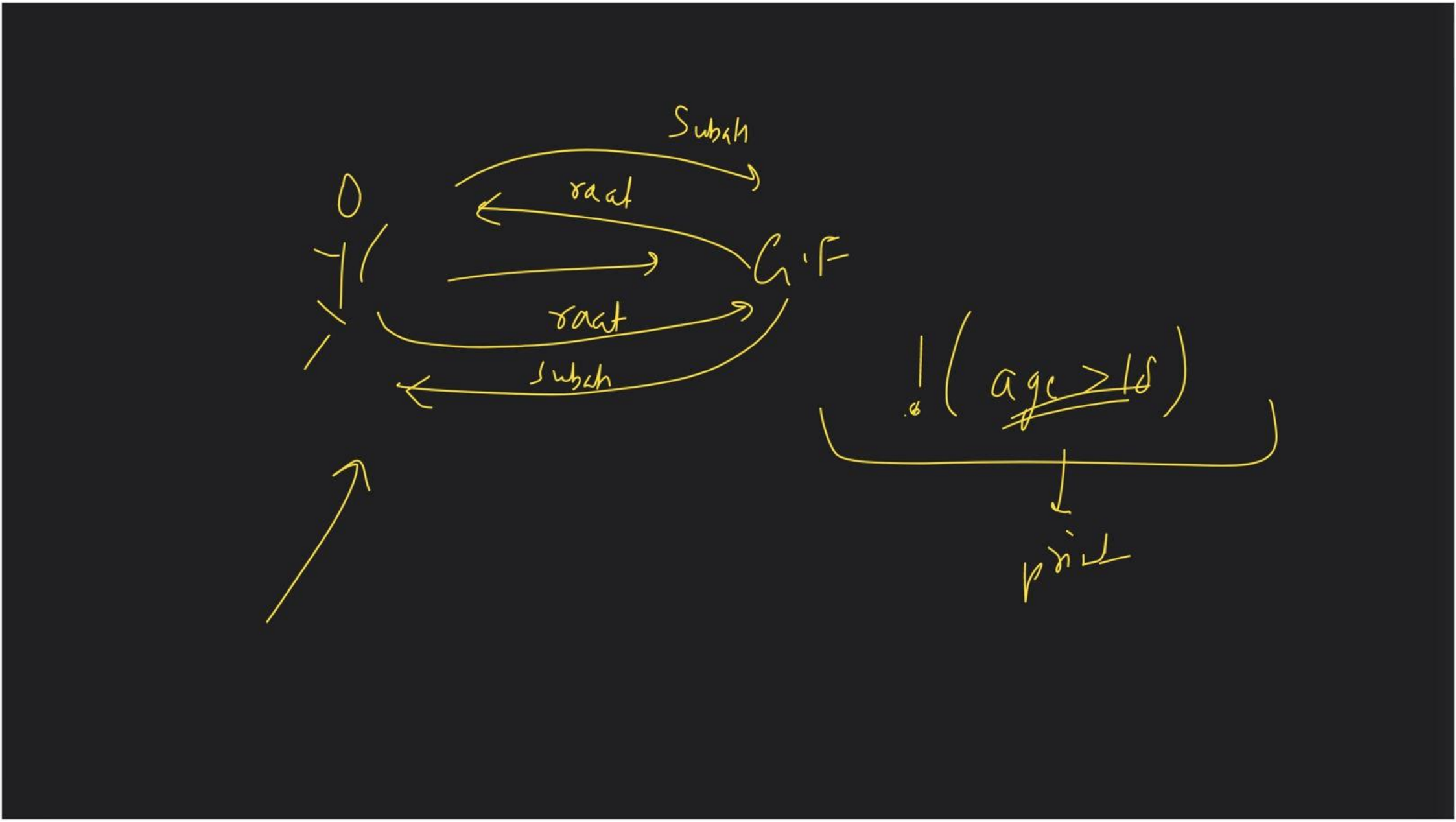
Operators are used to perform operations on variables and values.

- Arithmetic [+, -, *, /, %, ++, - -]
- Relational [==, !=, >, <, >=, <=]
- Assignment [=]
- Logical [&&, ||, !]
- Bitwise [&, |, ~, ^, >>, <<]




&&
Logical AND
TT → T
Condition
bool
a && b
a | b | o/p
T | T → T
T | F → F
F | T → F
F | F → F
Condition ✓
o/p → T/F
|| → Logical OR → atleast 1 is true → o/p → T
a || b
a | b | o/p
T | T → T
F | T → T
T | F → T
F | F → F
! → Logical NOT
a | !a
T → F
F → T
→ Reverse
→ flip


((age>18) &&
(gender=='M'))
print
T
age >= 18
saath me
gender == M
T
T
T
Licen / True
else
False
[(a
((age > 18) || (gender == 'M'))


Subash
Ravi
Subash
!(age>18)
print

Assignment:

Complete week-“Learn C++” recorded videos.

- 32 bit vs 64 bit Architecture
- Typecasting: Implicit and Explicit
- Binary and Decimal Number System