asics

++

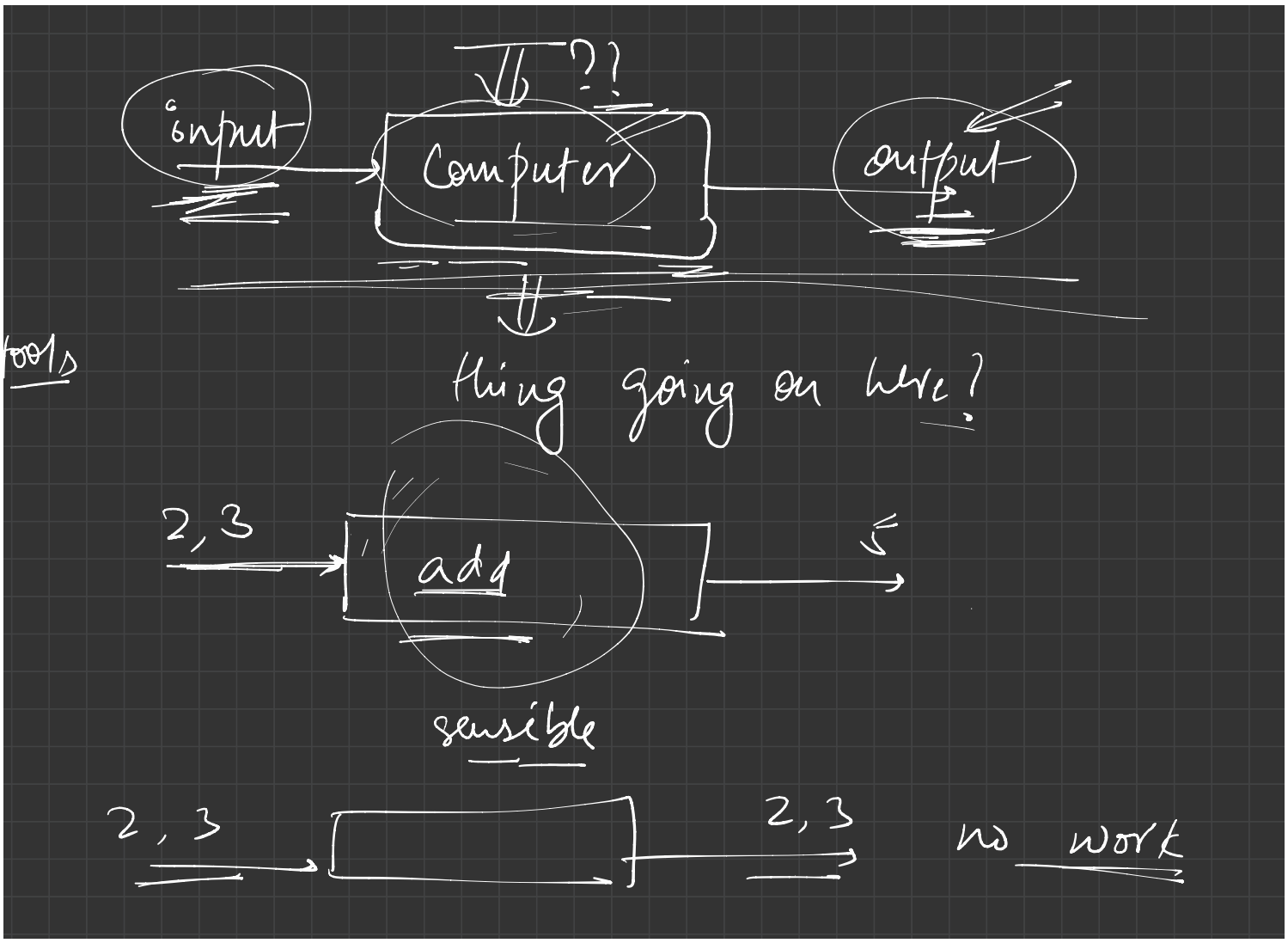
{ Basics
Fundamentals

C++

another language?

C++?
Algorithms?
Programs?

{ we are trying to communicate
with the computer? }


input
Computer
output
??
tools
thing going on here?
2, 3
add
sensible
2, 3
2, 3
no work

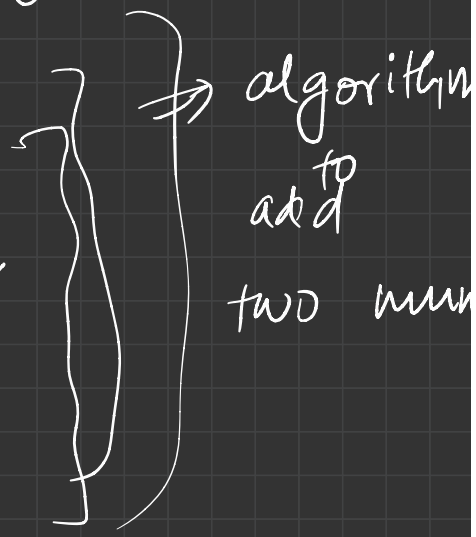
to do something and to get some output we give some instructions

↓

Programs

How to add two numbers?

we take the first number
we take the second number
we perform addition (maths)
and we tell the result.



e need to provide the computer with some
structions for it to work / give output.
input)
the set of instructions to get the output
for some input ⇒ Algorithm

- must be clear
- must be precise
- must be correct

Execution / Run

Computer → we are telling how to work on input and get output

{giving a set of instructio

Computer only understands binary

0 or 1

→ It is impossible to communicate/instru
with the computer in 0s and 1s.
binary

→ Computer understands Binary 0s and 1s

→ we understand → English | Hindi, etc.

How do we communicate?

Problem → Computer → Solu

Problem ⇒ give instruction in human readable | understandable

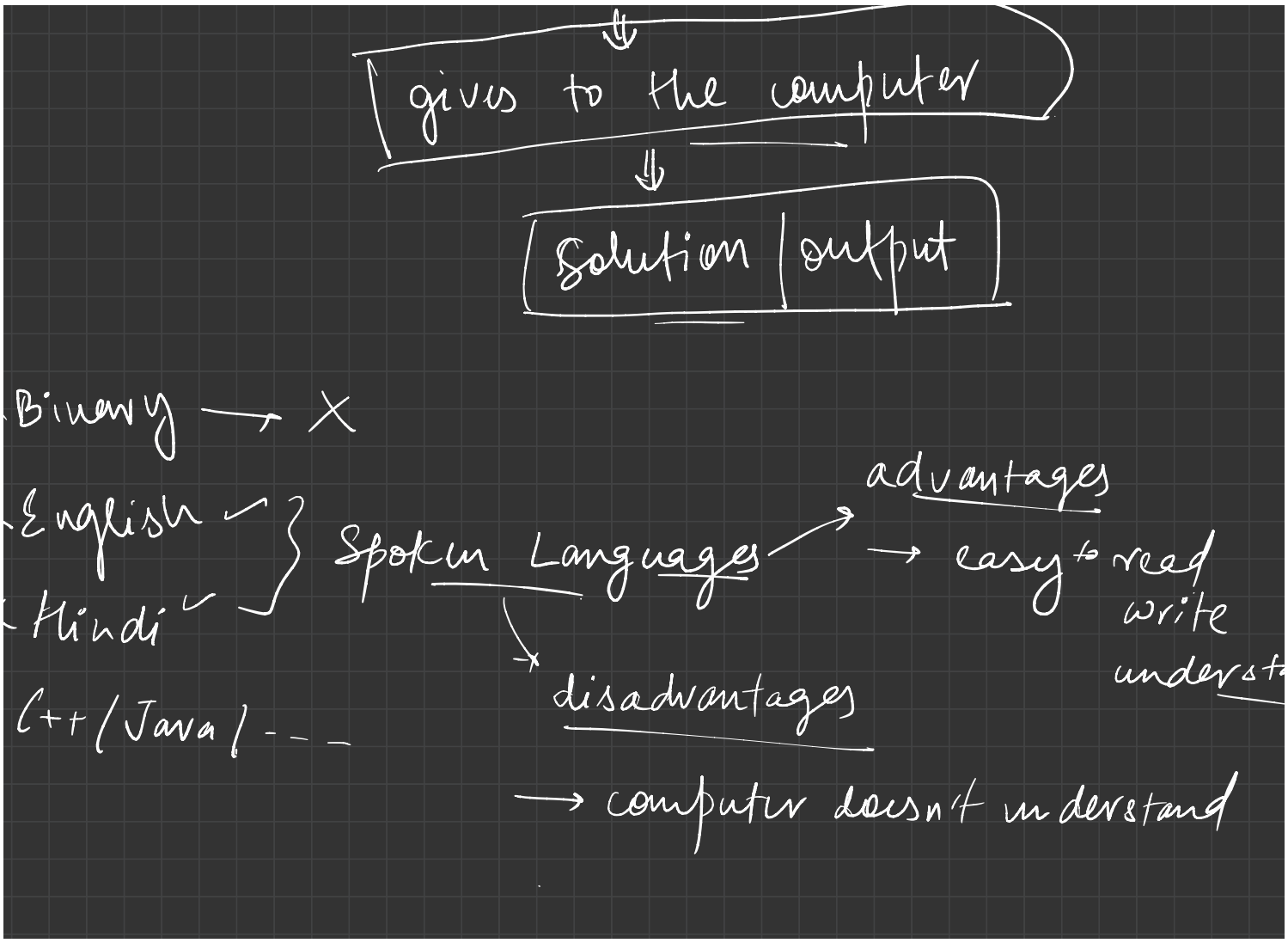
input

language → many

which one to u

translator

translates to Binary (0s and 1s)


gives to the computer
Solution | output
Binary → X
English
Hindi
Spoken Languages
advantages
→ easy to read write
disadvantages
→ computer doesn't understand
C++ / Java / ---

→ there are many

↓

We need as many translators

→ no proper set of rules

↓

different interpretations.

imprecise

It is not always possible to instruct the computer properly in a spoken language.

↓

too many ?

Why
ok em Language

We need some standard language to communicate with the computer.

↓

Programming Languages

(C, C++, Java, Python, ——)

↓

C++

A programming language is a set of rules that gives us a way to communicate with the computer.

↓

provide instructions

tell what to do.

- understandable by the humans
- should be able to properly translated to binary.

Syntax: grammar/set of rules of the programming language.

Programming Languages

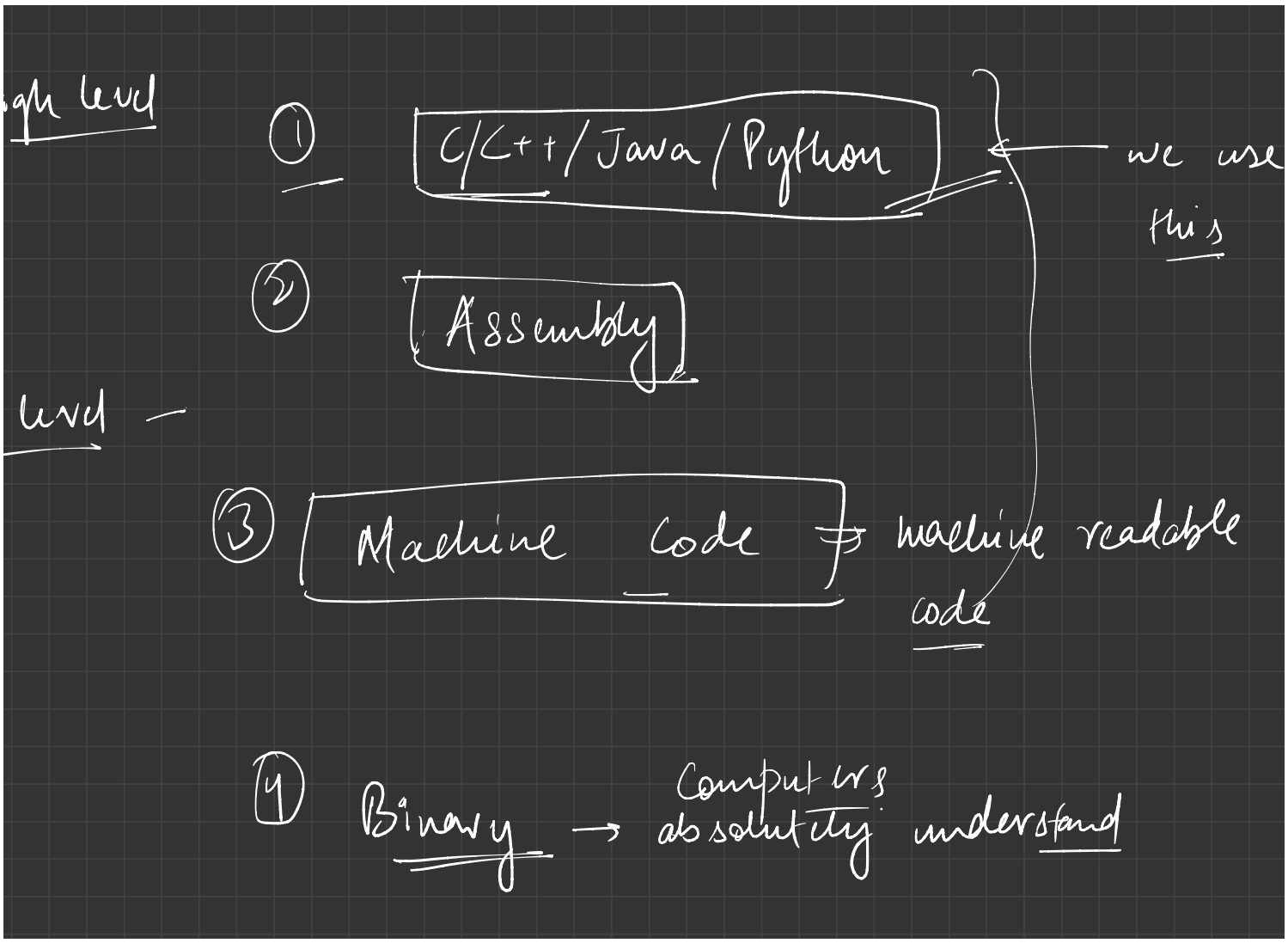
more understandable by humans — **High level**

more understandable by computers — **Low level**

Python, Java, C++, C, Assembly, Machine Code

Python → almost like english

print ("Hello")

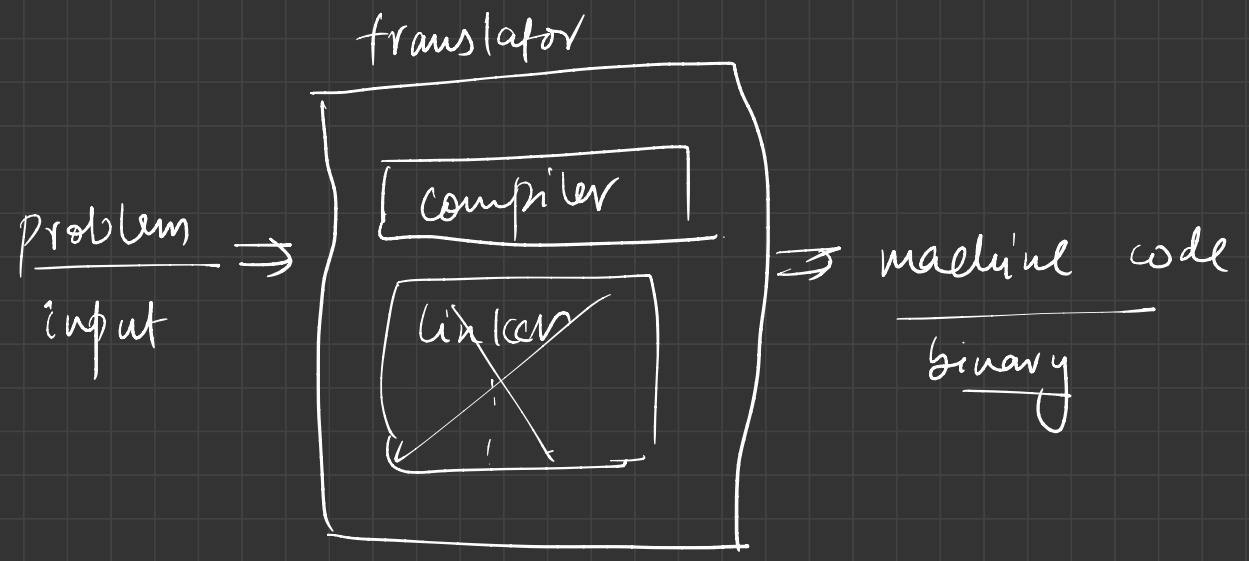

gh level
1
C/C++/Java/Python
we use this
2
Assembly
level –
3
Machine code
machine readable code
4
Binary → Computers absolutely understand

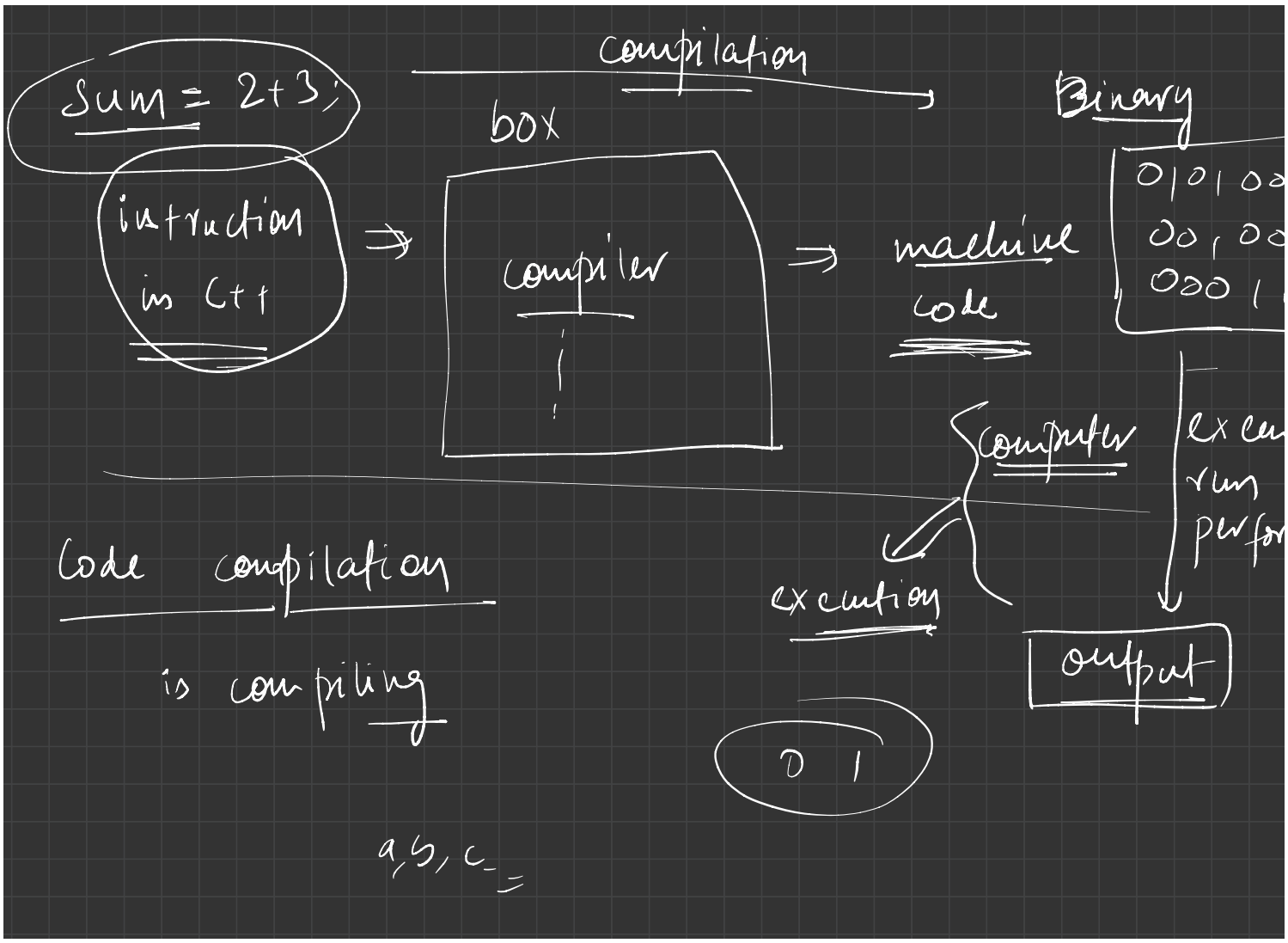
e only use High-level languages

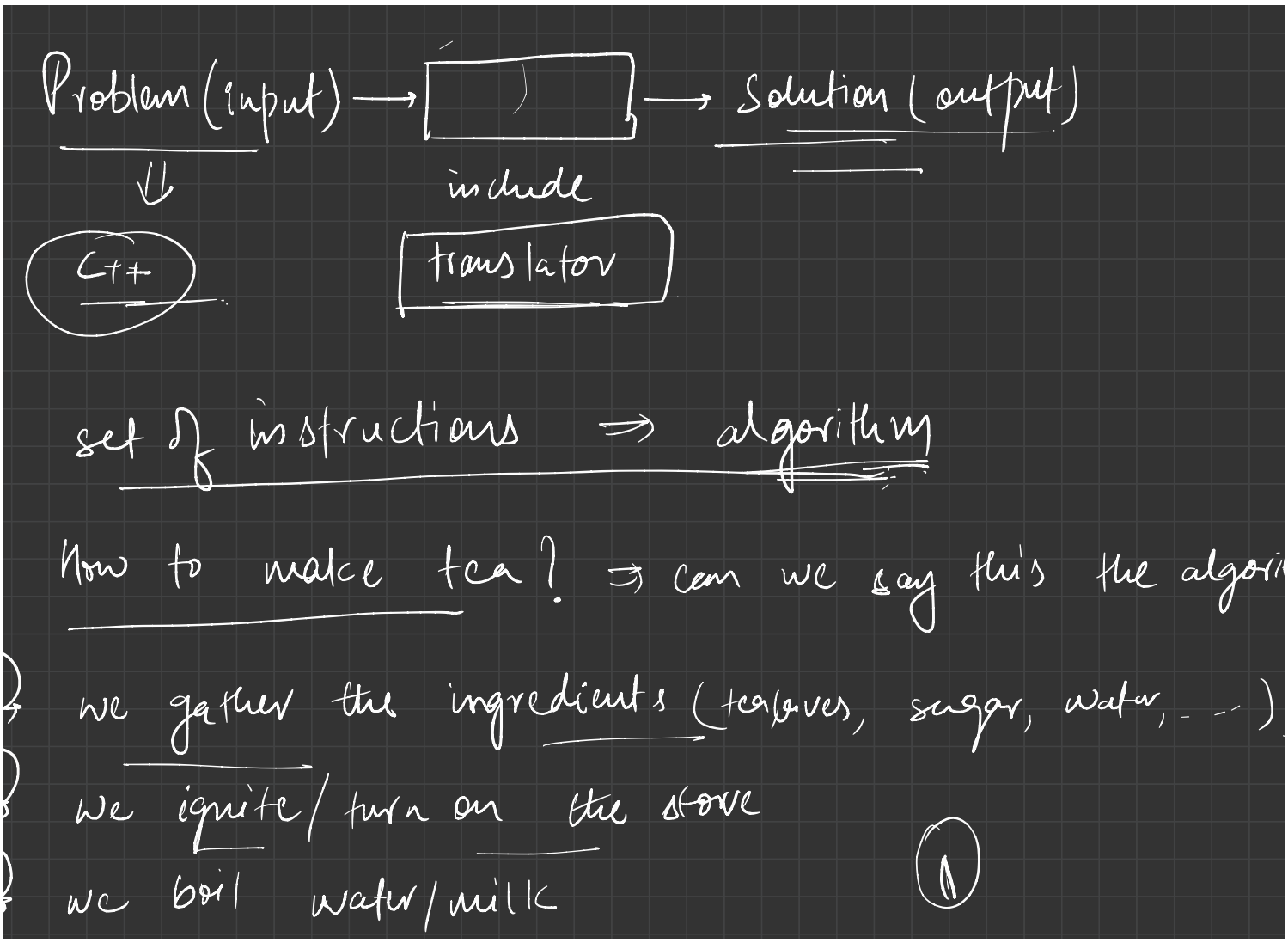
⇓

C++ ← we will use

how does it work?

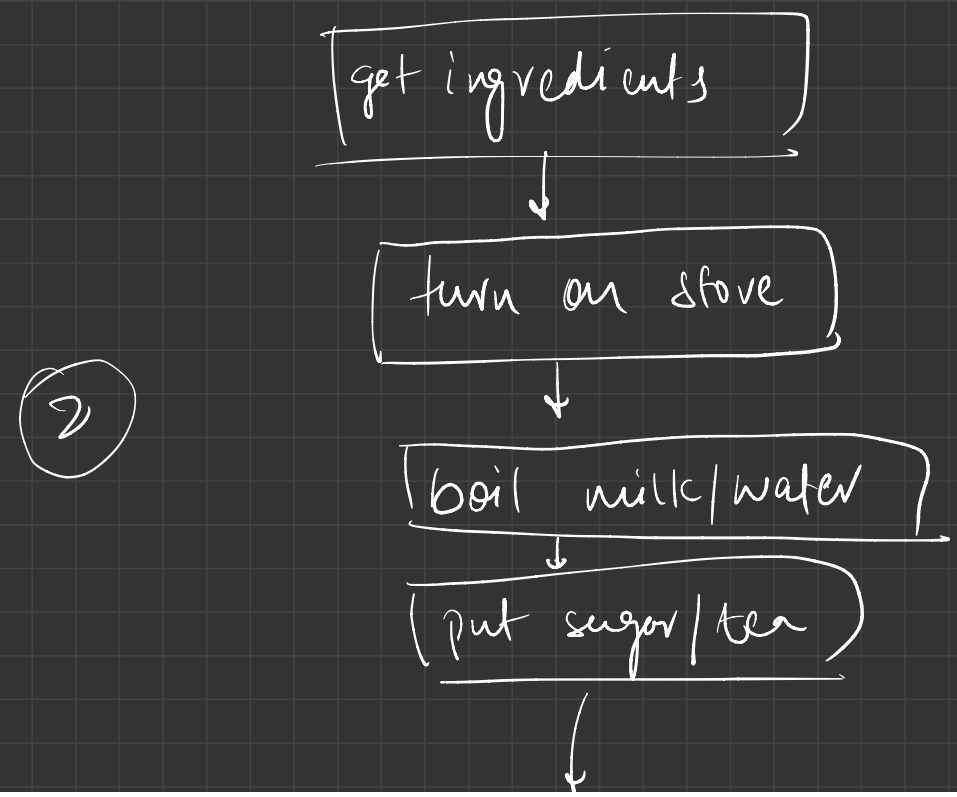



Sum = 2+3;
instruction in C++
compilation
box
compiler
machine code
Binary
0 1 0 1 0 0
0 0 , 0 0
0 0 0 1
computer
execution
output
Code compilation
is compiling
0 1
a, b, c =


Problem (input)
Solution (output)
include
translator
C++
set of instructions ⇒ algorithm
How to make tea? ⇒ can we say this is the algori
we gather the ingredients (tealeaves, sugar, water, ...)
we ignite / turn on the stove
we boil water / milk
1

we put sugar/tea leaves, etc.

we wait for 5 mins



flowchart

(2)

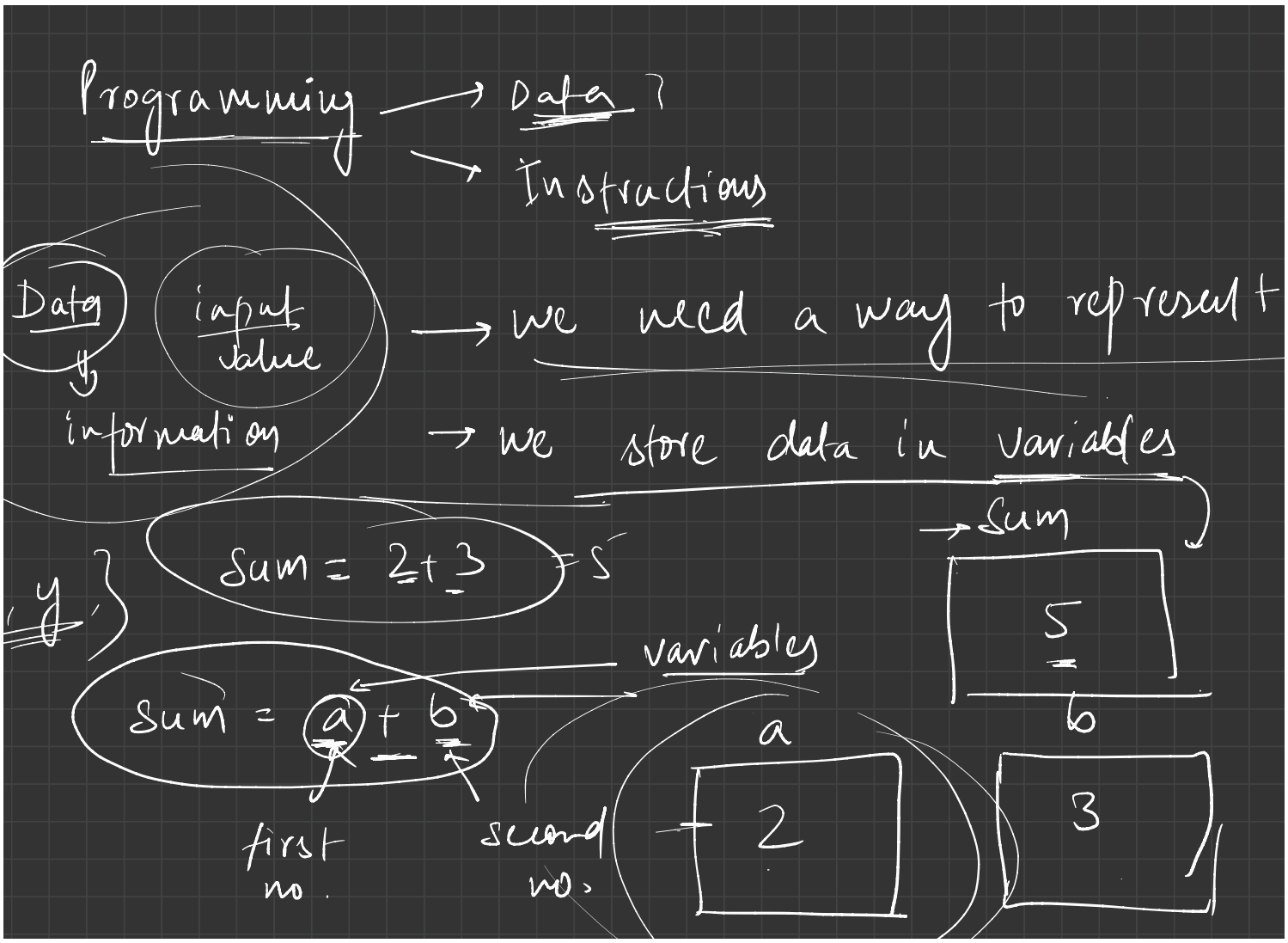
wait for 5 mins

The same algorithm can be expressed as

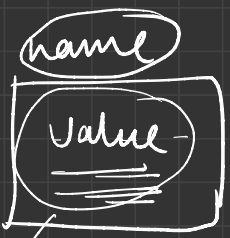
spoken language, flow chart, programming language code, pseudo code.

spoken language: complex & difficult to translate

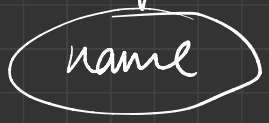
code: best for computers


Programming
Data
Instructions
Data
input value
information
we need a way to represent
we store data in variables
Sum = 2+3
Sum
5
y
variables
Sum = a + b
a
b
2
3
first no.
second no.

variables ⇒



bucket



Sum = a + b

2 + 3 = 5

→ getting the value

→ setting the value

→ creating a variable

naming rules (specific to C++)

① it must start with a letter or _ (underscore)

② and must only contain letters, numbers, or _

a ✓ → can be a variable name

Which of the following is not a proper variable name?

1. sum ✓
2. _sum ✓
3. _0abc ✓
4. 1vbc ✗
5. Pat246 ?? ✓
6. rat_1245_abc ✓
7. pqr%abc ✗

23@abc ✗

{a-z, A-Z} ✓

Instruction? → an order

Instructing a computer { spoken lang. / progm. lang. / binary (0100 11) }

Instruction: consisting of only 0s and 1s (low level)

→ some commands (high level)

cout << "Hello World"; in C++

↓

→ output → Hello World

Instructions

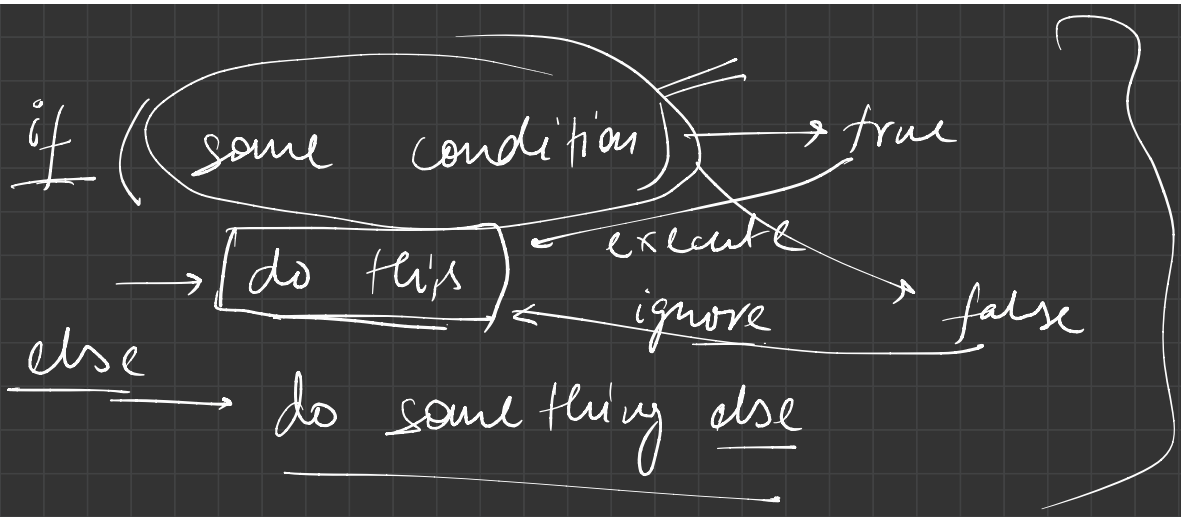
1. read / receive from a variable / taking an input
2. write / output / print
3. perform some operation (arithmetic ops).
   ↳ add, sub, multiply, di
4. assign a value to a variable (set)
5. do something based on some condition ⇒ Condition
6. repeat something ⇒ Loops

assign 5 to a

Conditionals (If-Else)

if (tomorrow is Sunday) then it is a Holiday,
else it is not a Holiday.

if (tomorrow is Sunday)
→ go to class
else
→ do not go to class



Doing something based on whether the given condition is true / false.

a [5]

if (a is divisible by 2)

→ output (a is even) & ignore

else

→ output (a is odd)

Loops

→ do something many times

→ output (Hello World)

Output "Hello World" 5 times ?

→ output (Hello World)
→ output (Hello World)
→ output (HelloWorld)
→ output (HelloWorld)
→ output (Hello World)

Solve !!

1000

- instead use a loop → true

while (some condition)

→ do something

for-loop
do-while
do-until

repetitionsLeft = 5

while (repetitionsLeft > 0) {

output ("Hello World")

decrease (repetitionsLeft)

}

3

Dry Run:

repetitionsLeft = 3

repetitions left > 0 → tru

3 > 0 ✓

→ Write "I will not mis behave" 100 times.

I will not misbehave

I will not -

⋮

⇒ Computer

output ("I will not --

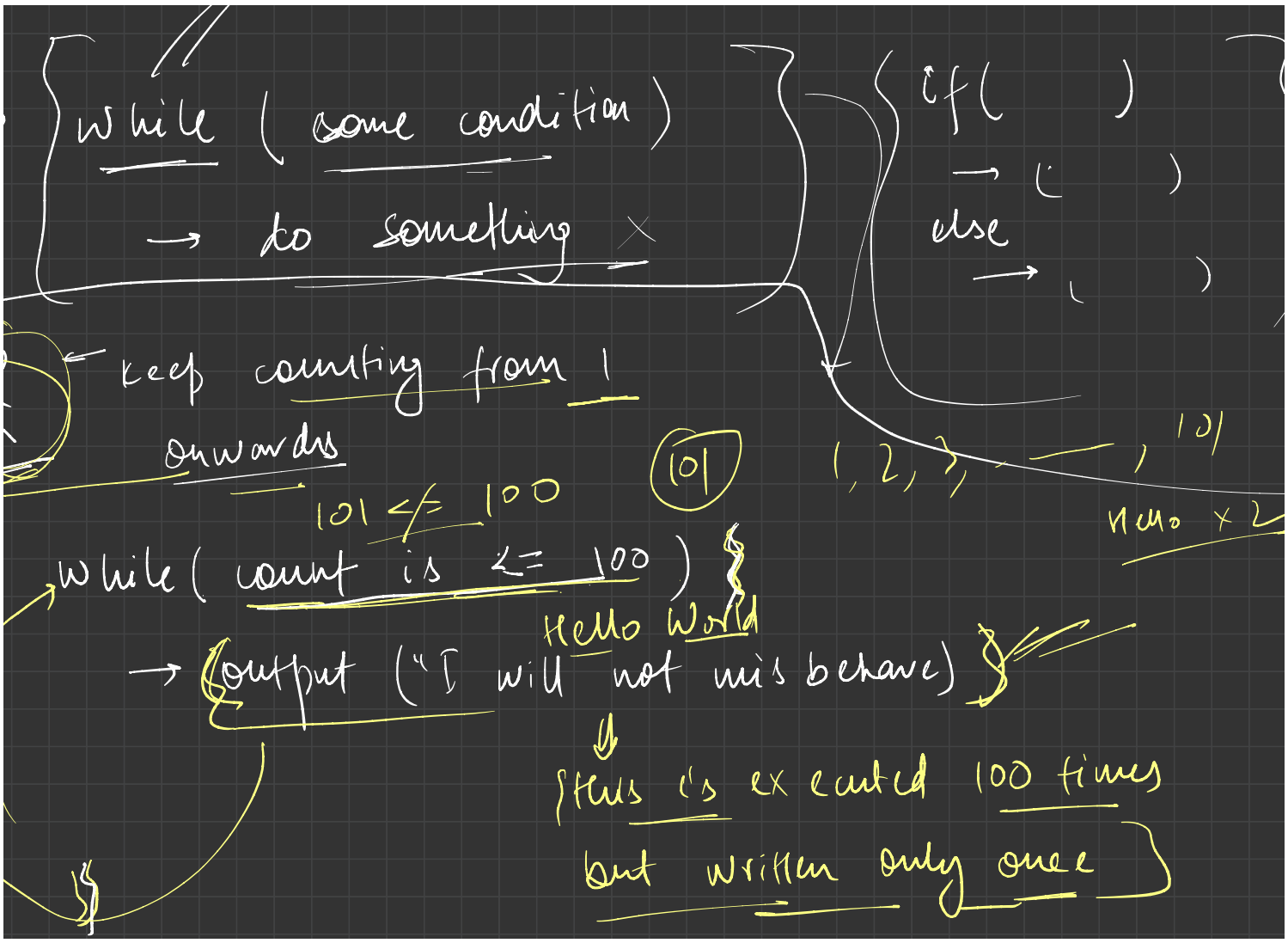
↳ out);

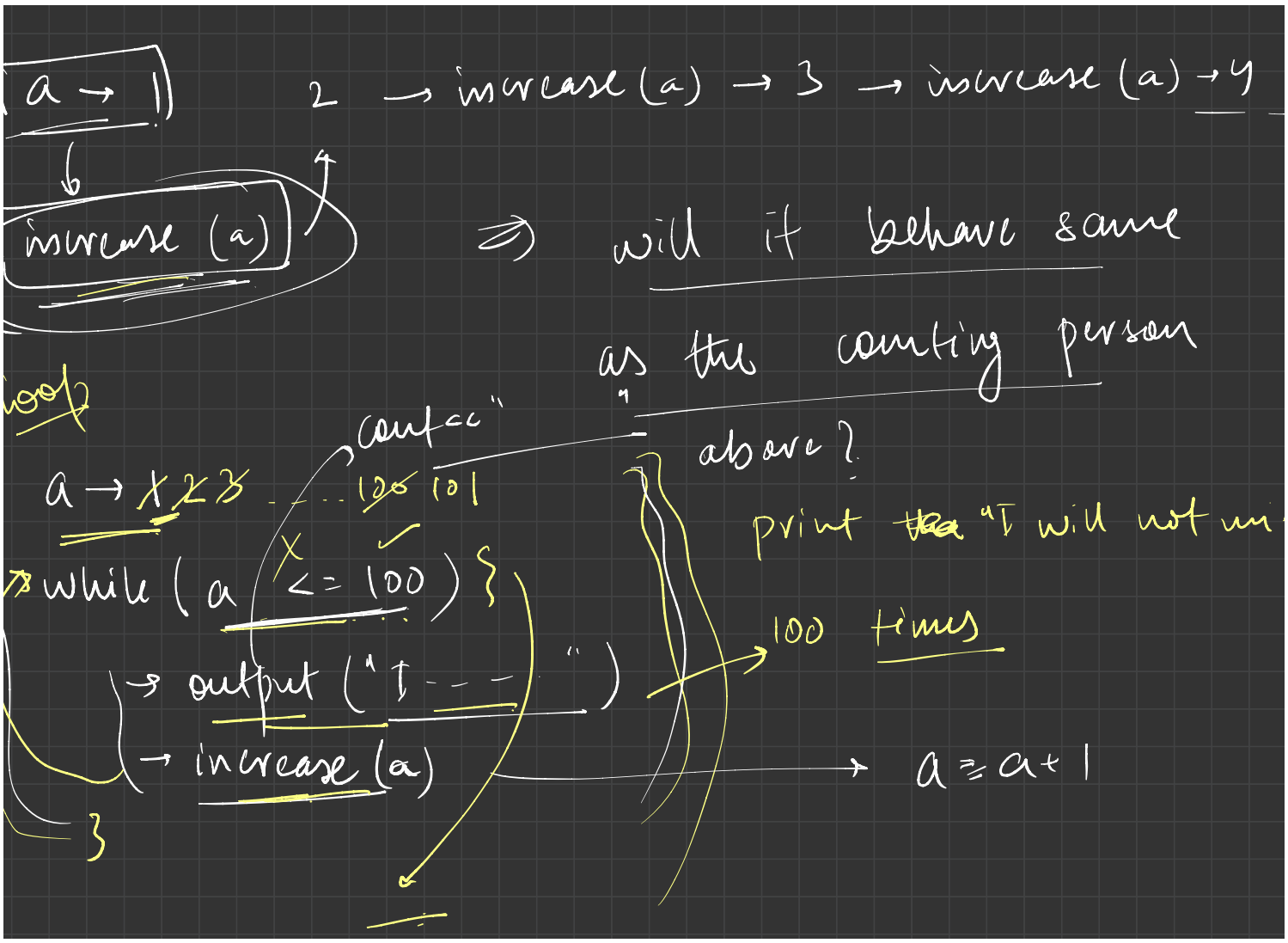
out

⋮

100 times

↗ While - loop

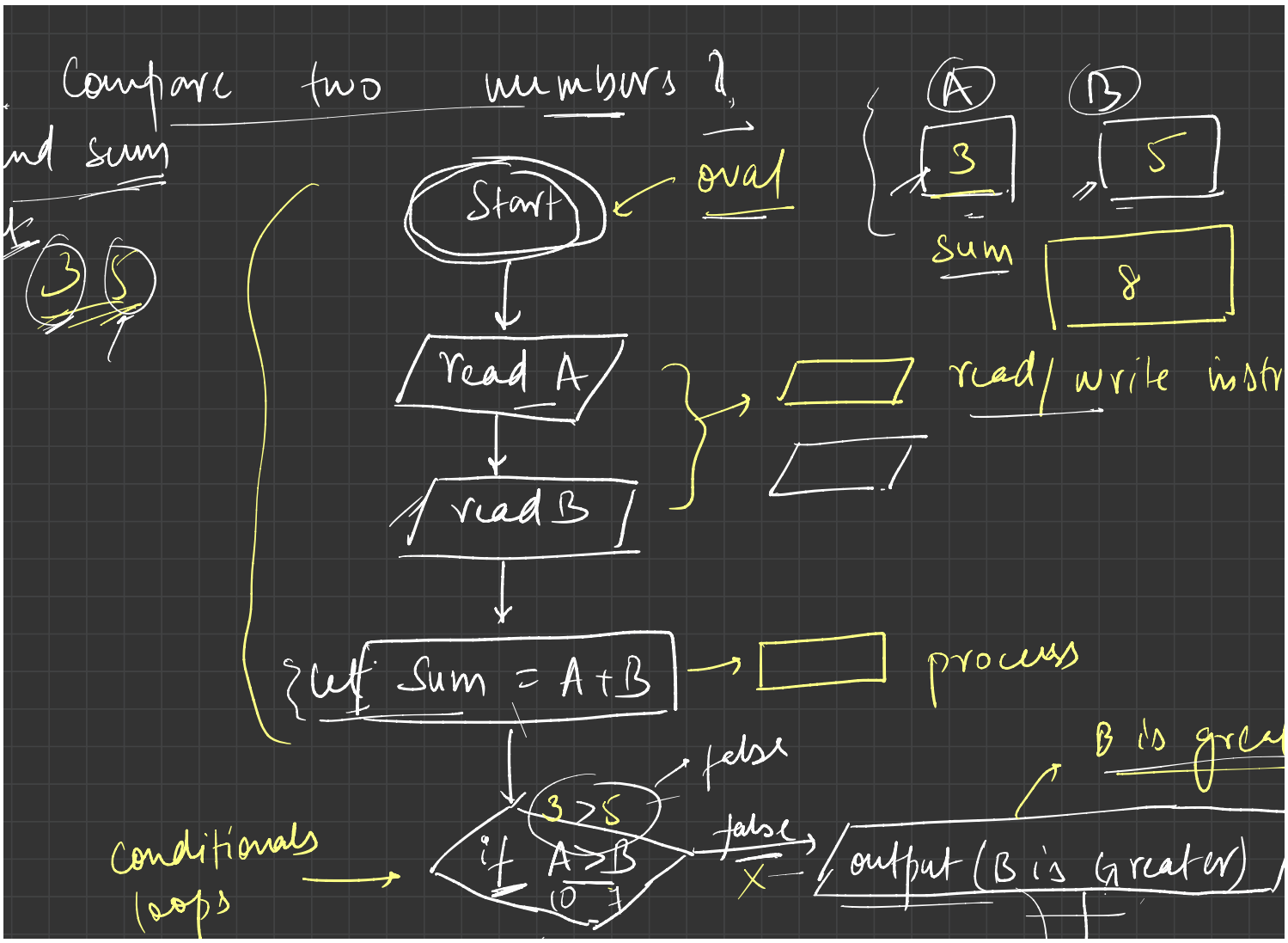

while (some condition)
→ do something
if( )
→ ( )
else
→ ( )
keep counting from 1 onwards
101 <= 100
101
1, 2, 3, ..., 101
Hello World x 2
while (count is <= 100)
→ {output ("I will not misbehave)}
Hello World
this is executed 100 times
but written only once

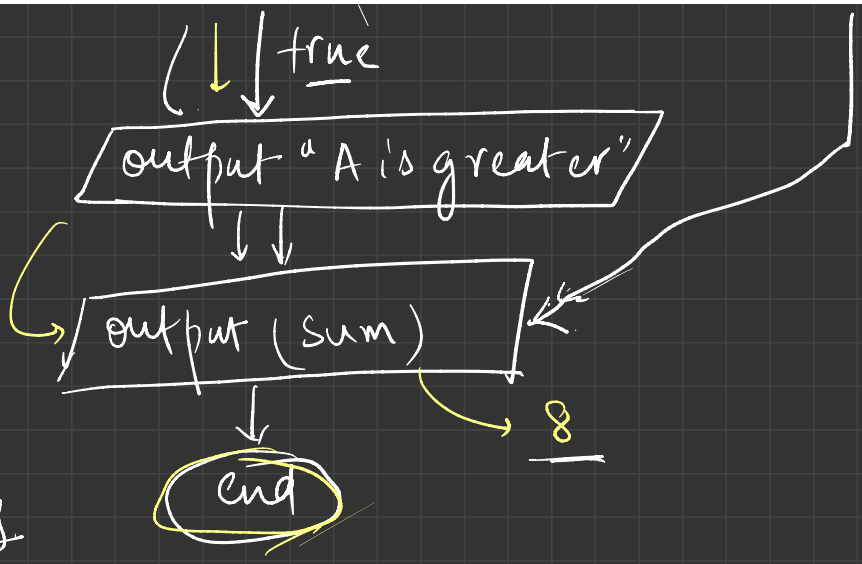

a → 1
2 → increase(a) → 3 → increase(a) → 4
increase(a)
⇒ will it behave same as the counting person above?
loop
a → 1 2 3 ... 100 101
while (a <= 100) {
output ("I ---")
increase(a)
}
a = a + 1
100 times
print "I will not

flow chart → diagrammatic representation of

Pseudocode → set of instructions

- read
- write
- assign
- perform operation
- conditionals
- loops


Compare two numbers ?
nd sum
3
5
Start
oval
A
B
3
5
sum
8
read A
read B
read/write instr
Let Sum = A+B
process
3>5
false
if A>B
(0 1)
false
X
output (B is Greater)
B is grea
conditionals
loops



Programming Lang

pseudo Code ⇒ an abstract way to represent instructions / flowch

① start

② read A

③ read B

④ sum ← A+B

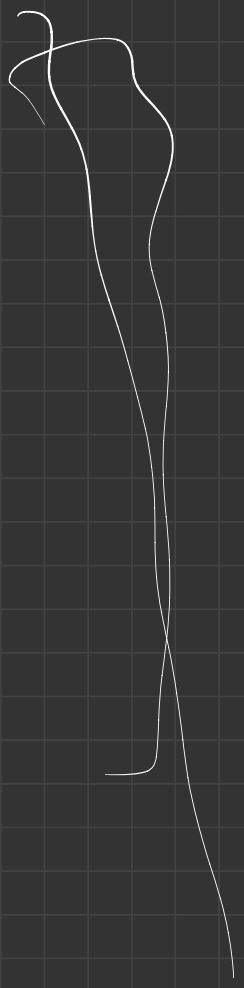
⑤ if (A>B)

output ('A is greater')

else

output ('B is greater')

⑥ output (sum)



(7) end

how does it gets the solution / output

⇓

following inst: (algorithm)