

The alternative
billions o

**... and it is with great difficulty that
I selected the student Charley**

Ricky, you ever think...
what if it's too much?

Programming, algorithms...
What if we can't keep up?

One for Harvard
and one for Oxford

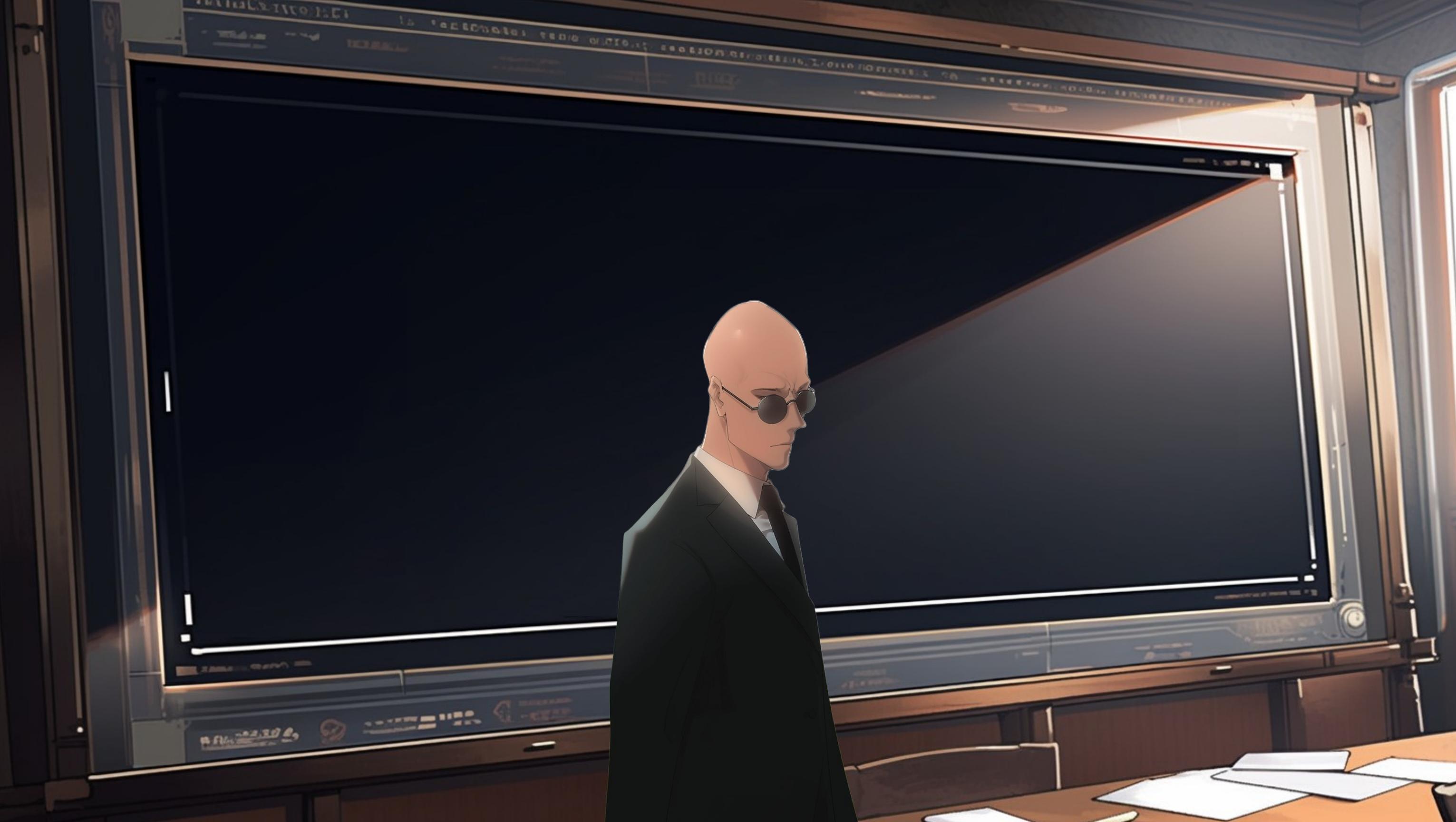
Dany, my man,
when have you ever
not aced a test?

You worry about
everything!

And so, with evident reluctance, the UN approved and established UPU,
Ultimate Programming University, granting it exceptional permissions.



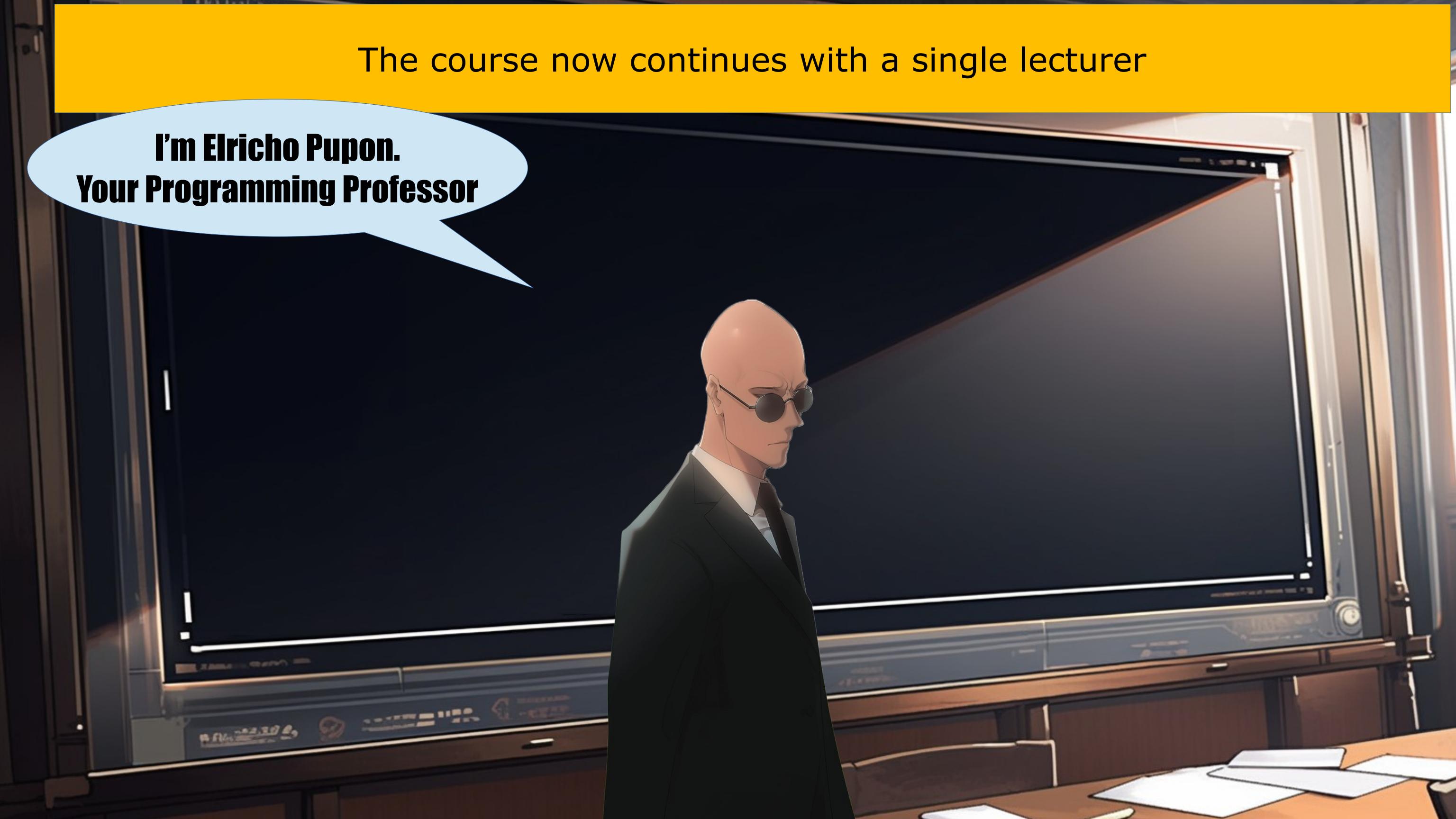
**I'm Elricho Pupon.
Your Programming Professor**



The course now continues with a single lecturer

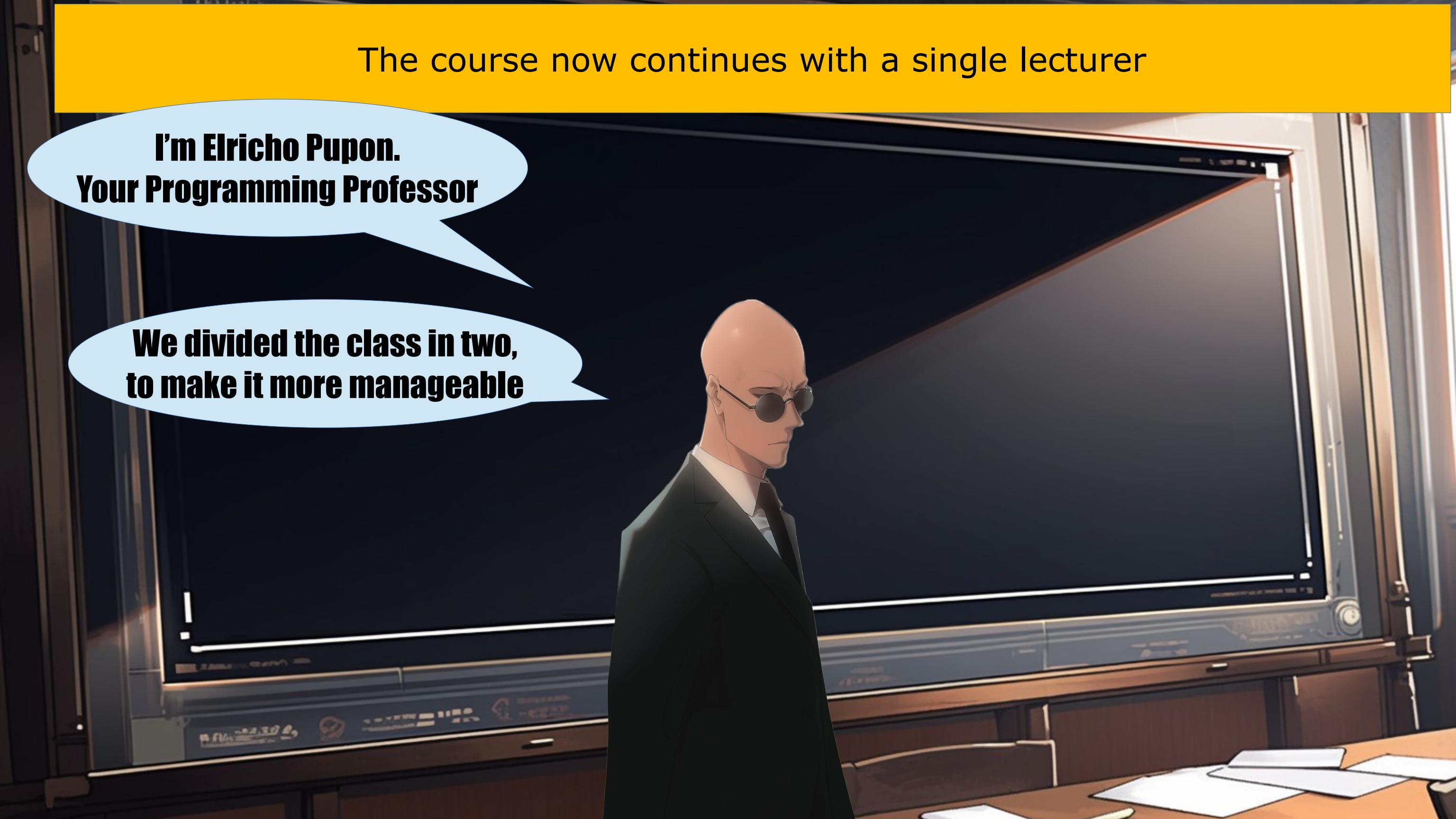


The course now continues with a single lecturer



I'm Elricho Pupon.
Your Programming Professor

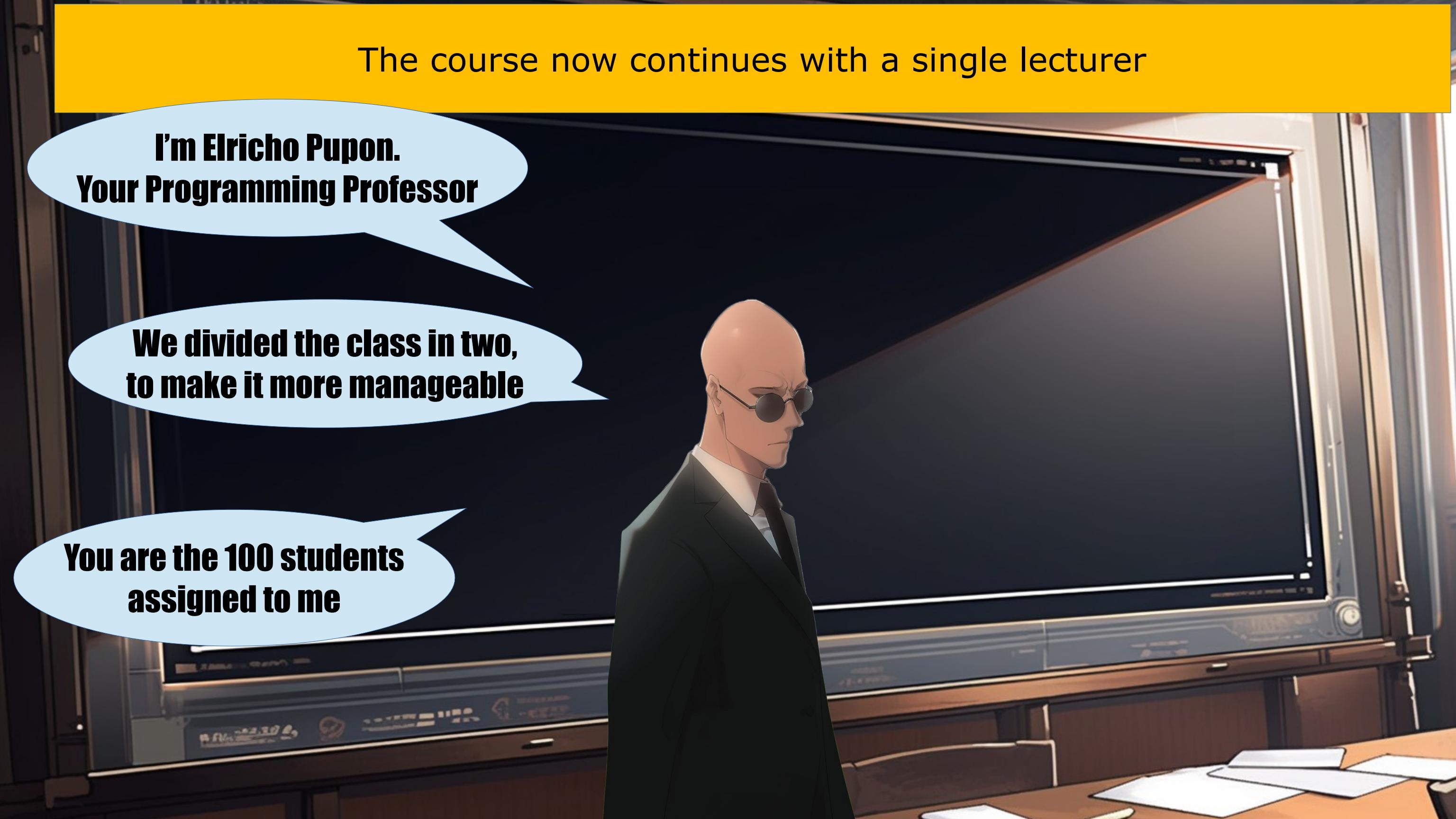
The course now continues with a single lecturer



I'm Elricho Pupon.
Your Programming Professor

We divided the class in two,
to make it more manageable

The course now continues with a single lecturer

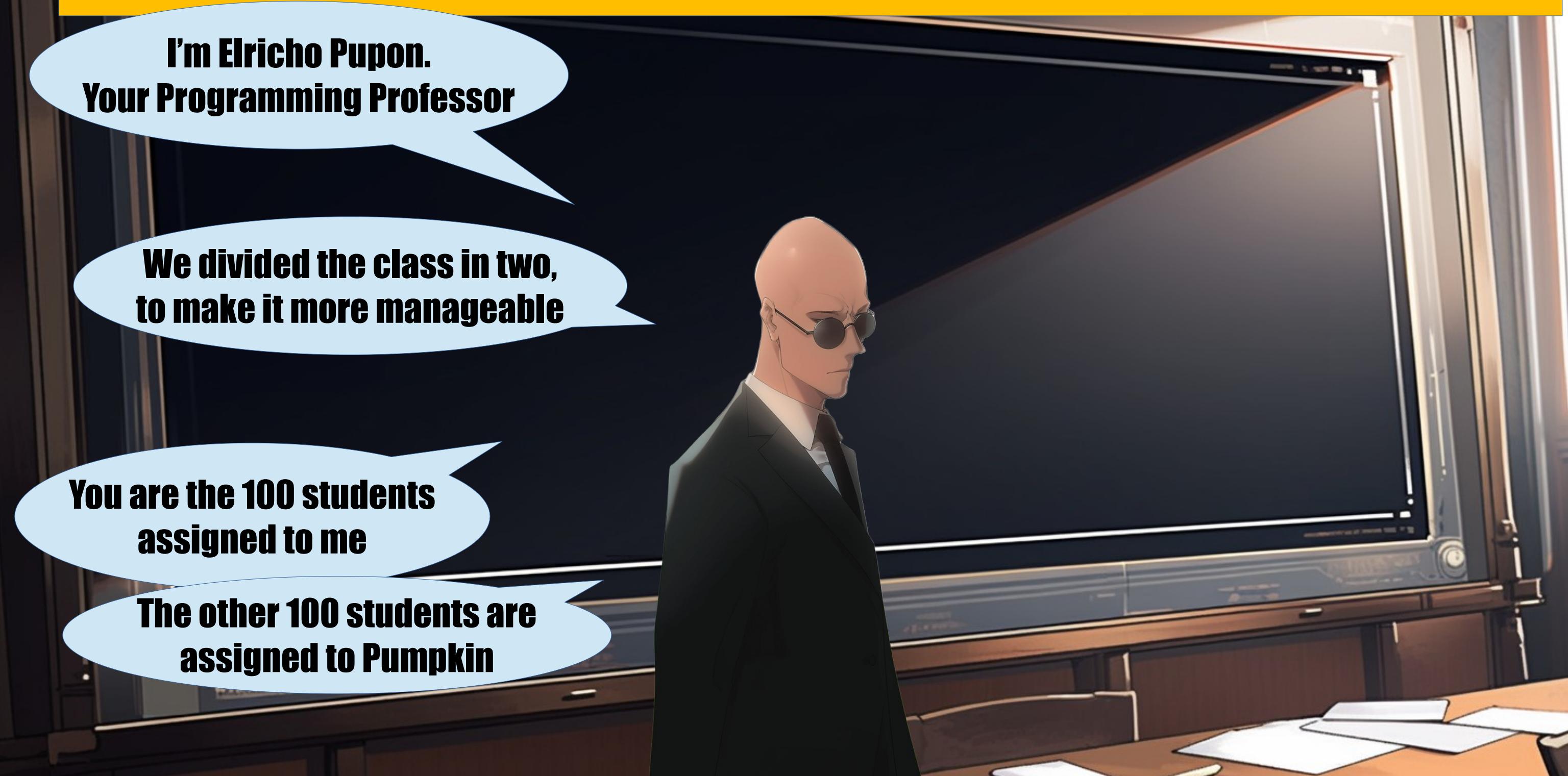


I'm Elricho Pupon.
Your Programming Professor

We divided the class in two,
to make it more manageable

You are the 100 students
assigned to me

The course now continues with a single lecturer



I'm Elricho Pupon.
Your Programming Professor

We divided the class in two,
to make it more manageable

You are the 100 students
assigned to me

The other 100 students are
assigned to Pumpkin



Variables:



Variables:

Places where the computer stores information.

They have names, called identifiers.

We can access them by their names, and we can update the data stored in them by using the 'equal' sign.

For example, variables can contains numbers, and we can use conventional operations on those numbers, like addition, multiplication or subtraction



Variables:

Places where the computer stores information.

They have names, called identifiers.

We can access them by their names, and we can update the data stored in them by using the 'equal' sign.

For example, variables can contain numbers, and we can use conventional operations on those numbers, like addition, multiplication or subtraction



Variables:

Places where the computer stores information.

They have names, called identifiers.

We can access them by their names, and we can update the data stored in them by using the 'equal' sign.

For example, variables can contains numbers, and we can use conventional operations on those numbers, like addition, multiplication or subtraction



Variables:

Places where the computer stores information.

They have names, called identifiers.

We can access them by their names, and we can update the data stored in them by using the 'equal' sign.

For example, variables can contains numbers, and we can use conventional operations on those numbers, like addition, multiplication or subtraction





**We can start
with some
example code**



**We can start
with some
example code**

**We are going to use Java, but
this would work the same in
many other languages**



```
int a = 24;
```



```
int a = 24;  
int b = 50;
```



```
int a = 24;  
int b = 50;
```



I'm declaring two variables,
“a” and “b”

```
int a = 24;  
int b = 50;
```

**I'm declaring two variables,
“a” and “b”**

They are both of type “int”



```
int a = 24;  
int b = 50;
```



I'm declaring two variables,
“a” and “b”

They are both of type “int”

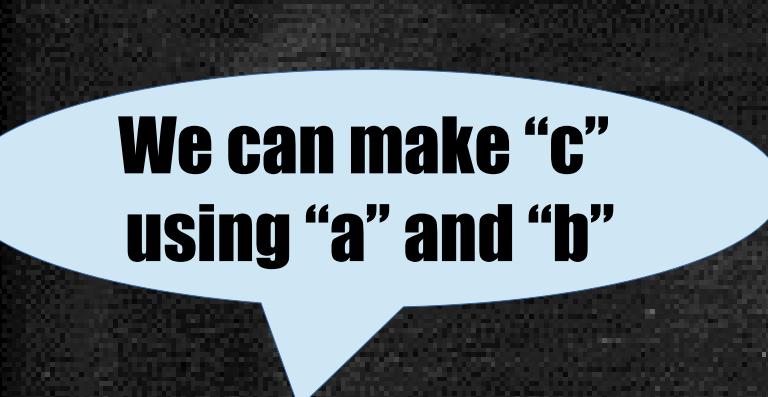
They are initialized with
values “24” and “50”

```
int a = 24;  
int b = 50;
```



```
int a = 24;  
int b = 50;  
int c = a + b + 10;
```





**We can make “c”
using “a” and “b”**

```
int a = 24;  
int b = 50;  
int c = a + b + 10;
```



**We can make “c”
using “a” and “b”**

```
int a = 24;  
int b = 50;  
int c = a + b + 10;
```

Here “c” is “a” plus “b” plus “10”



We can make “c”
using “a” and “b”

```
int a = 24;  
int b = 50;  
int c = a + b + 10;
```

Here “c” is “a” plus “b” plus “10”

As you can see, we can use
variable names or numbers

```
int a = 24;  
int b = 50;  
int c = a + b + 10;
```



```
int a = 24;  
int b = 50;  
int c = a + b + 10;  
assert c == ??;
```





What is the value of “c”?

```
int a = 24;  
int b = 50;  
int c = a + b + 10;  
assert c == ??;
```



What is the value of “c”?

```
int a = 24;  
int b = 50;  
int c = a + b + 10;  
assert c == ??;
```

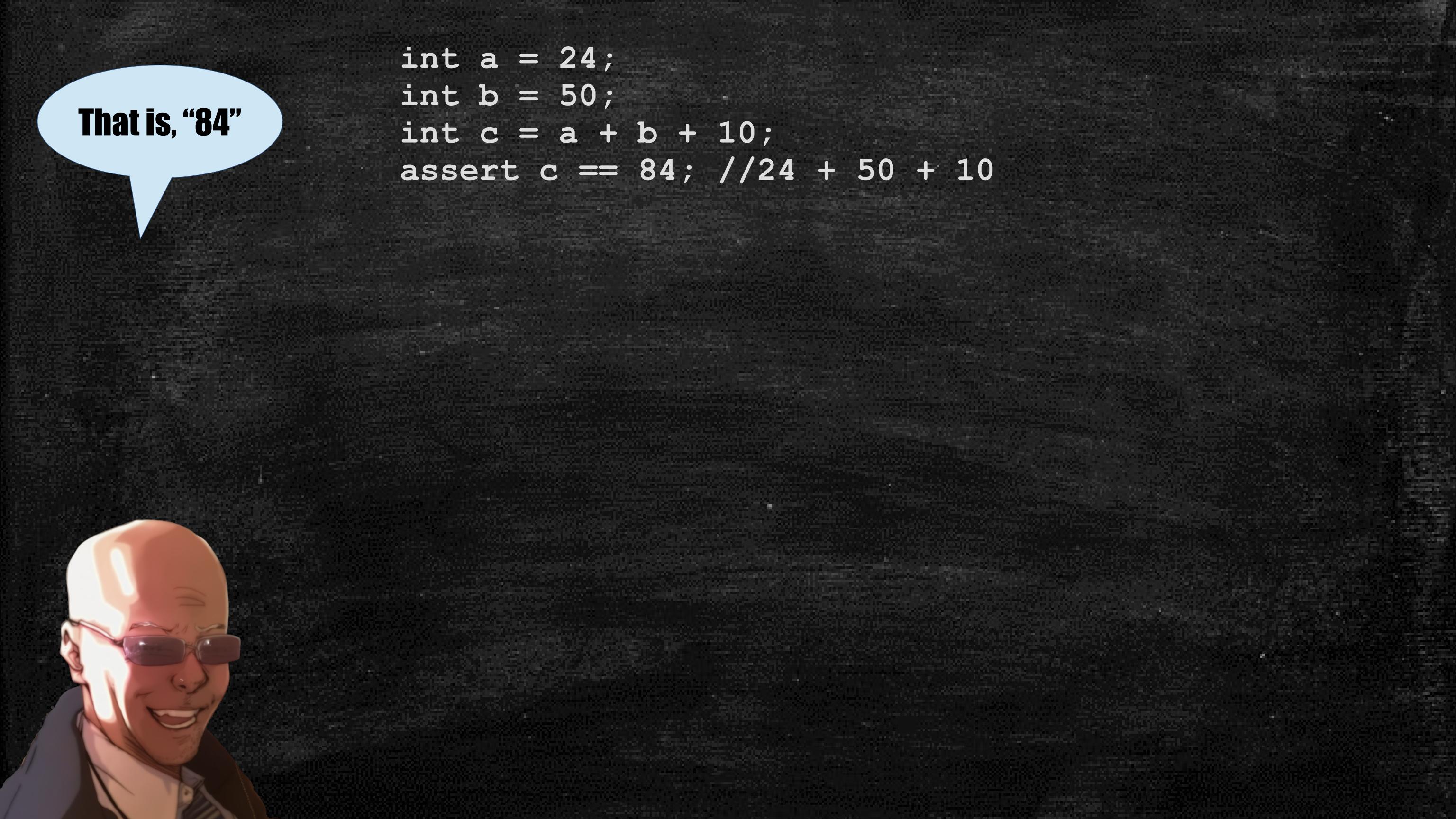
**It is the value of “a” plus
the value of “b” plus “10”**

```
int a = 24;  
int b = 50;  
int c = a + b + 10;  
assert c == ??; // + +
```



```
int a = 24;  
int b = 50;  
int c = a + b + 10;  
assert c == ??; //24 + 50 + 10
```





That is, “84”

```
int a = 24;  
int b = 50;  
int c = a + b + 10;  
assert c == 84; //24 + 50 + 10
```



That is, “84”

```
int a = 24;  
int b = 50;  
int c = a + b + 10;  
assert c == 84; //24 + 50 + 10
```

**We write “assert” to mark expressions
that should hold true**



That is, “84”

```
int a = 24;  
int b = 50;  
int c = a + b + 10;  
assert c == 84; //24 + 50 + 10
```

**We write “assert” to mark expressions
that should hold true**

**“equals equals” evaluates to “true” if the
left and the right value are the same, and “false”
otherwise**





We can reuse a variable
instead of making a new one

```
int a = 24;  
int b = 50;
```

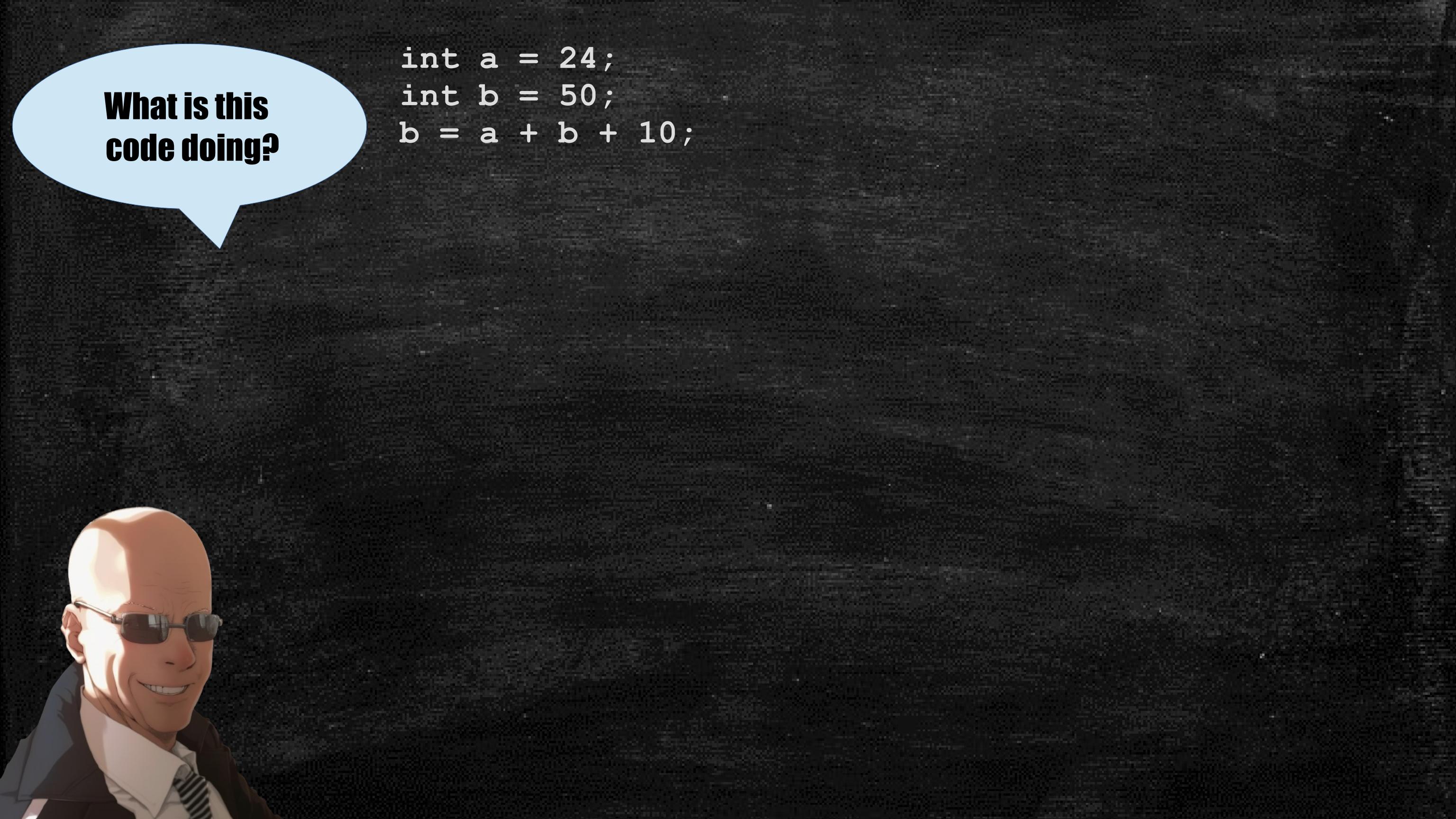


```
int a = 24;  
int b = 50;  
b =
```



```
int a = 24;  
int b = 50;  
b = a + b + 10;
```



A cartoon illustration of a bald man with a mustache, wearing dark sunglasses and a black suit jacket over a white shirt. He is looking towards the right. A light blue speech bubble originates from his mouth, containing the text "What is this code doing?".

**What is this
code doing?**

```
int a = 24;  
int b = 50;  
b = a + b + 10;
```



**What is this
code doing?**

```
int a = 24;  
int b = 50;  
b = a + b + 10;
```

**It is updating the value of “b”
using the result of the expression
on the right**



**What is this
code doing?**

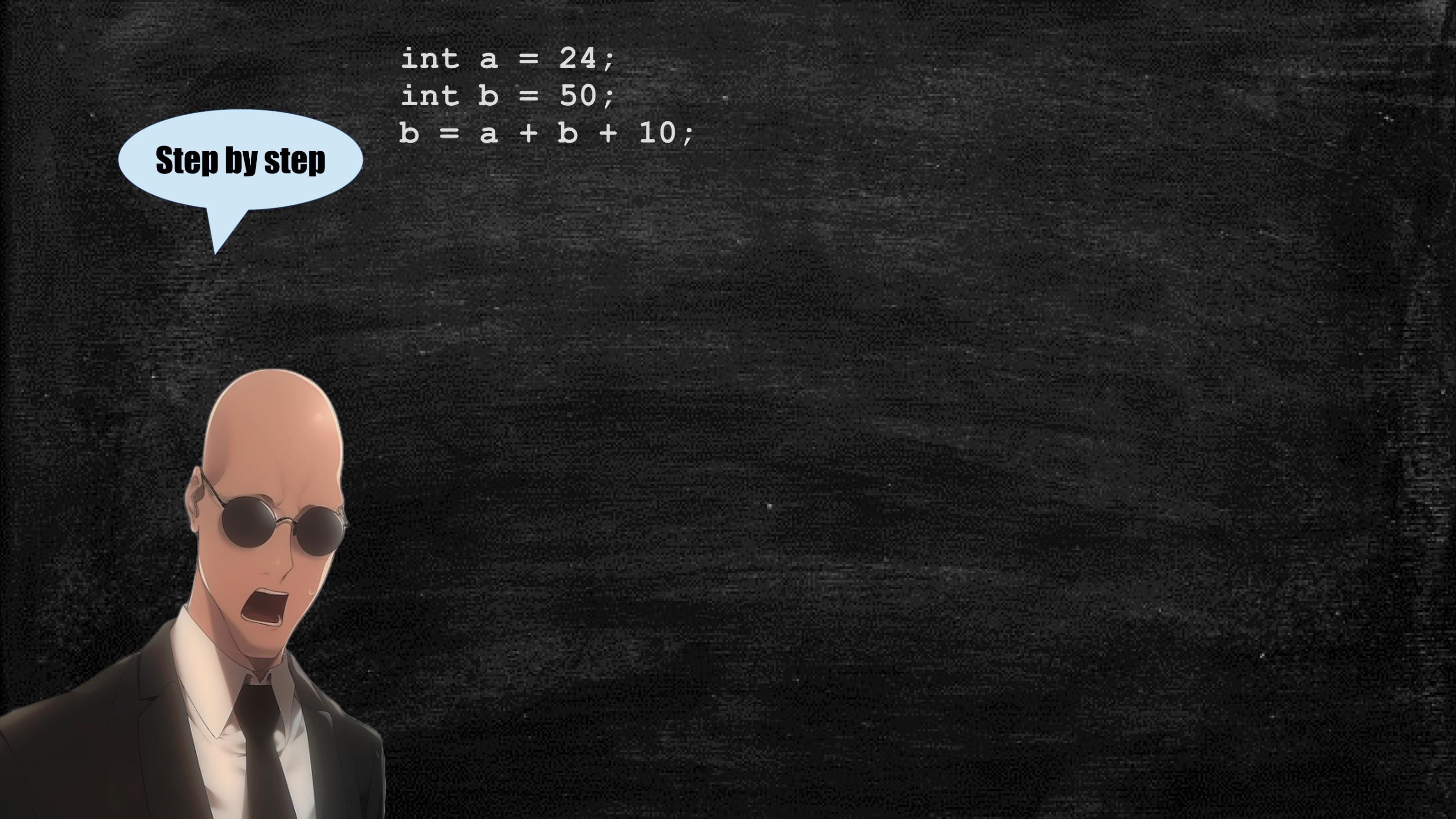
```
int a = 24;  
int b = 50;  
b = a + b + 10;
```

**It is updating the value of “b”
using the result of the expression
on the right**

**That is, the “equals” is not like the
math equals, where we state that two terms
have the same value**

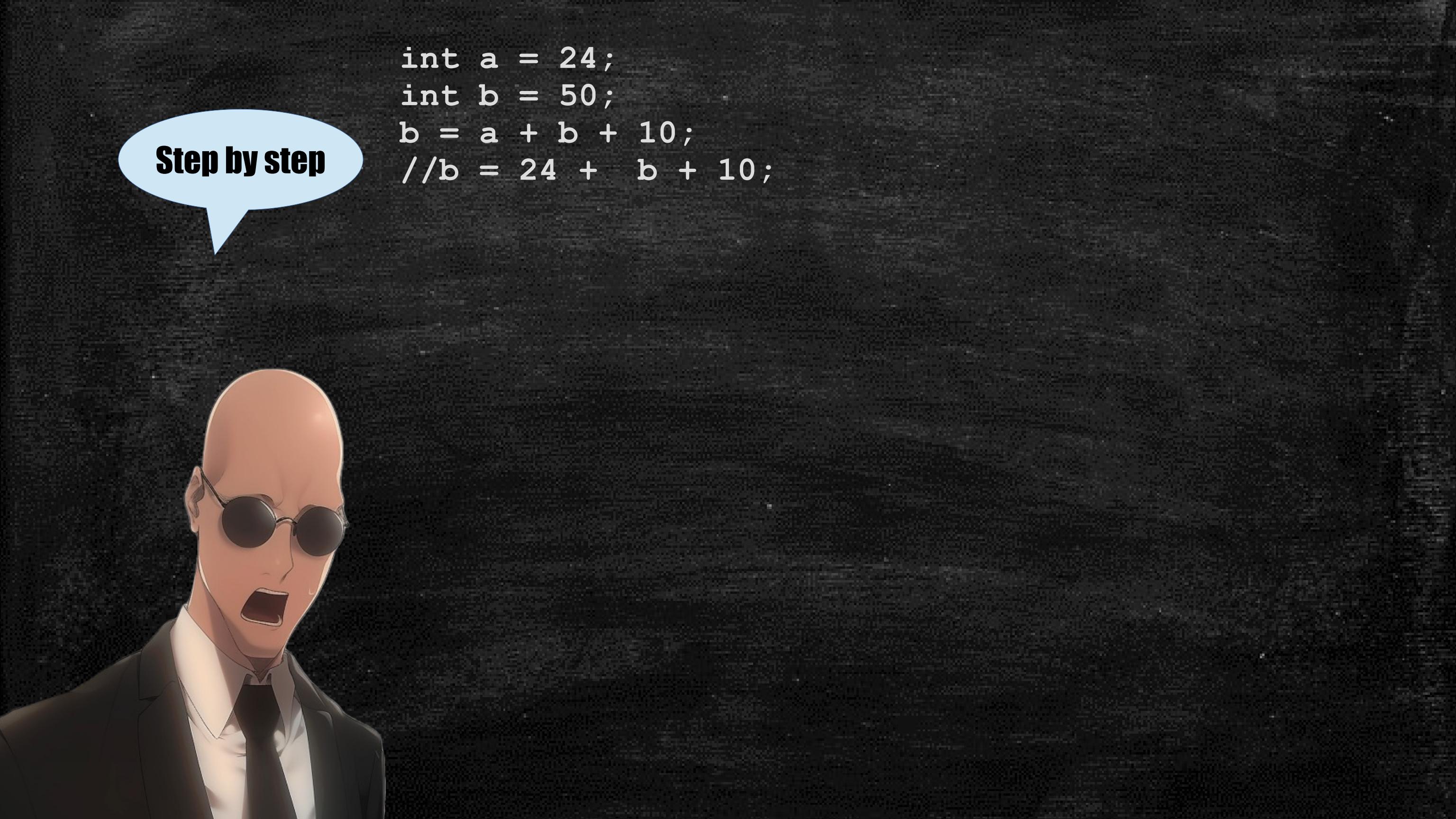
```
int a = 24;  
int b = 50;  
b = a + b + 10;
```





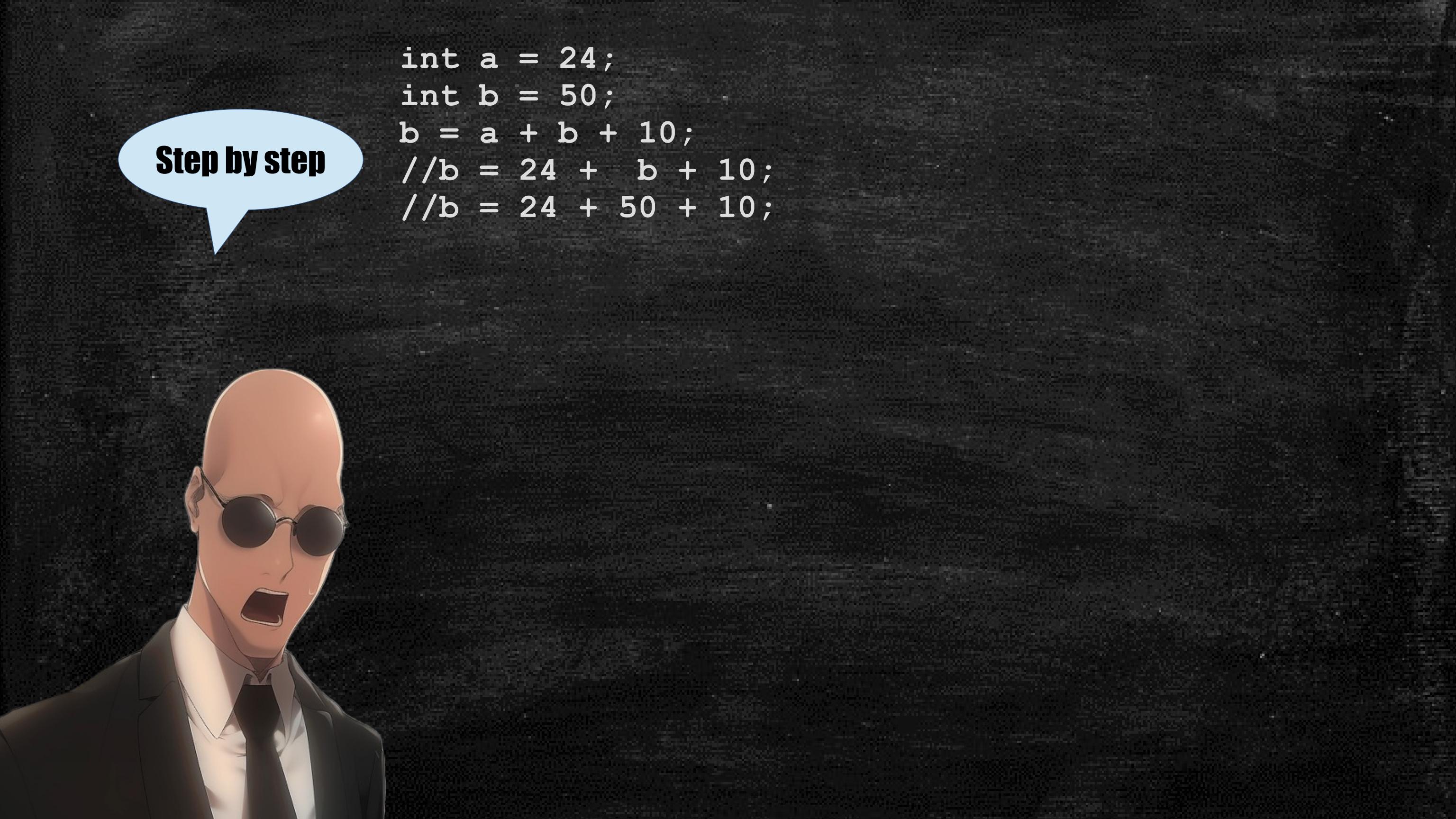
Step by step

```
int a = 24;  
int b = 50;  
b = a + b + 10;
```



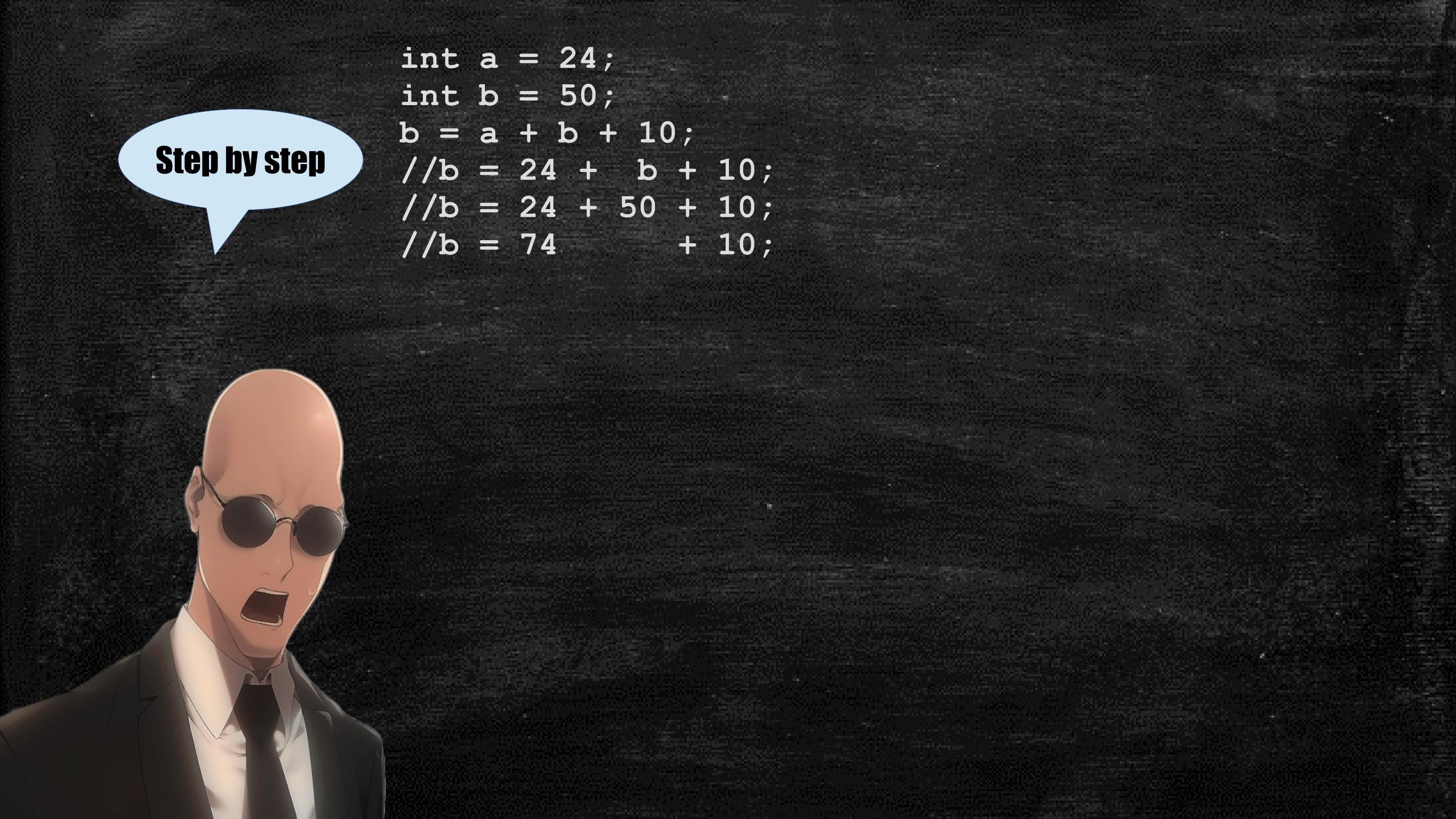
Step by step

```
int a = 24;  
int b = 50;  
b = a + b + 10;  
//b = 24 + b + 10;
```

A cartoon illustration of a bald man wearing dark sunglasses and a suit. He has a shocked or surprised expression with his mouth open. A light blue speech bubble originates from his head, containing the text "Step by step".

Step by step

```
int a = 24;  
int b = 50;  
b = a + b + 10;  
//b = 24 + b + 10;  
//b = 24 + 50 + 10;
```



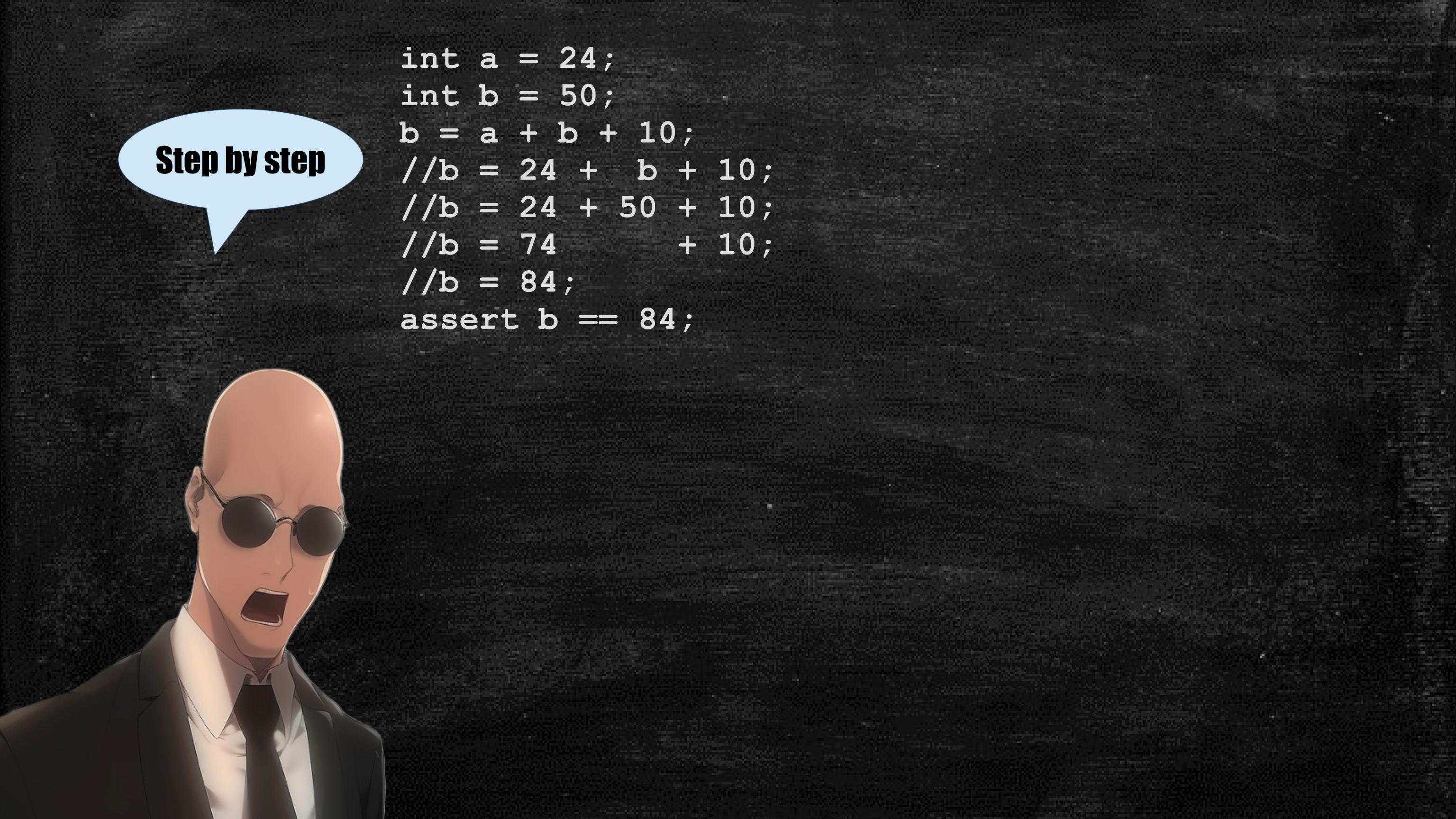
Step by step

```
int a = 24;  
int b = 50;  
b = a + b + 10;  
//b = 24 + b + 10;  
//b = 24 + 50 + 10;  
//b = 74 + 10;
```



Step by step

```
int a = 24;  
int b = 50;  
b = a + b + 10;  
//b = 24 + b + 10;  
//b = 24 + 50 + 10;  
//b = 74 + 10;  
//b = 84;
```

A cartoon illustration of a bald man wearing dark sunglasses and a suit. He has a shocked or surprised expression with his mouth open. A light blue speech bubble originates from his mouth, containing the text "Step by step".

Step by step

```
int a = 24;  
int b = 50;  
b = a + b + 10;  
//b = 24 + b + 10;  
//b = 24 + 50 + 10;  
//b = 74 + 10;  
//b = 84;  
assert b == 84;
```



Step by step

```
int a = 24;  
int b = 50;  
b = a + b + 10;  
//b = 24 + b + 10;  
//b = 24 + 50 + 10;  
//b = 74 + 10;  
//b = 84;  
assert b == 84;
```

**We can see that first “a” is read,
then “b” is read,
then the sum is computed,
finally “b” is updated**





**What if we wanted to swap
the content of two variables?**





**We can see our two variables
as our two hands,
holding one ball each**



**We can see our two variables
as our two hands,
holding one ball each**

**A hand only
holds one ball**



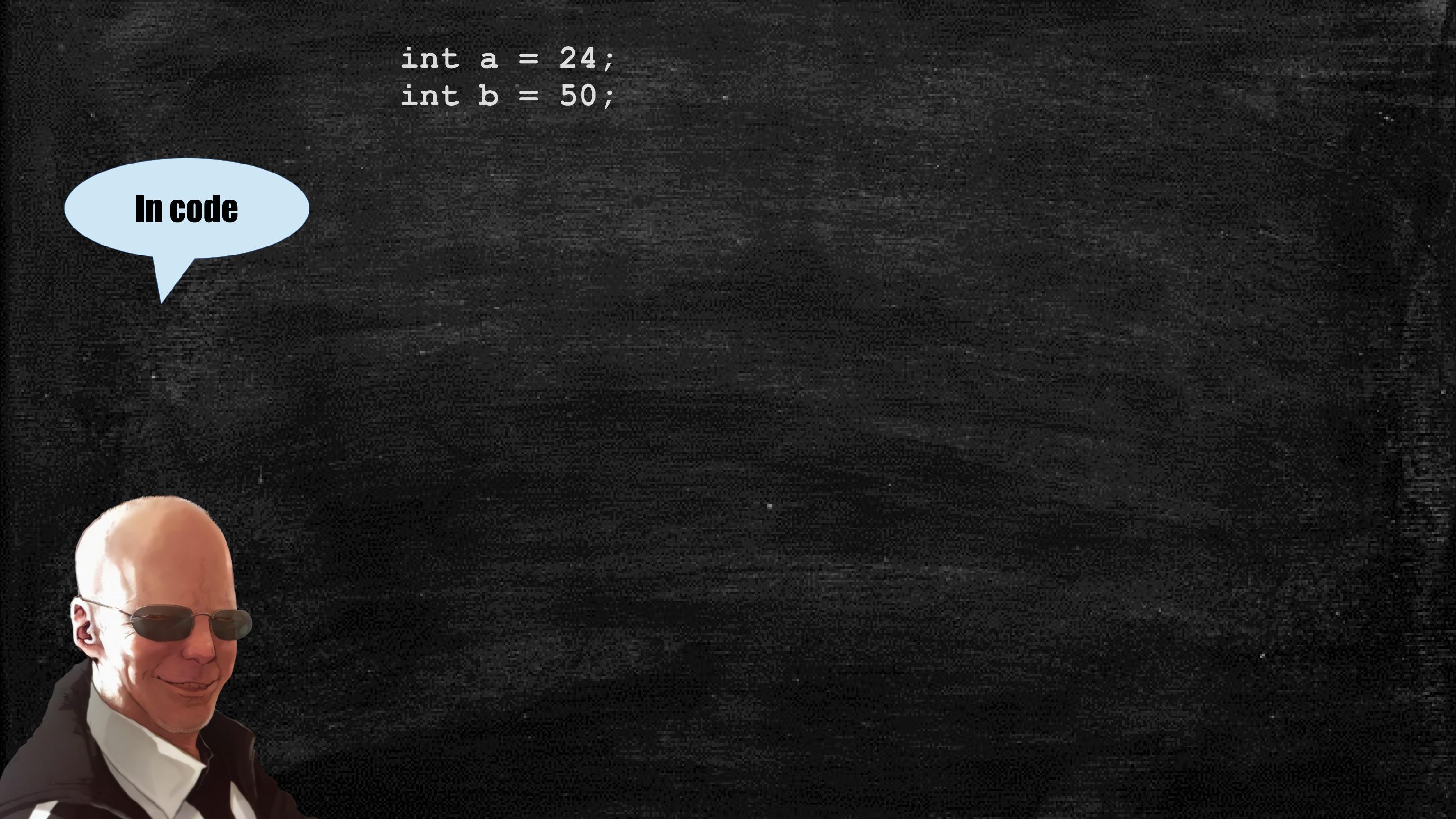
**We can see our two variables
as our two hands,
holding one ball each**

**A hand only
holds one ball**

**Taking the other ball, we have to
abandon the current ball, which
would fall down, out of reach**

```
int a = 24;  
int b = 50;
```





```
int a = 24;  
int b = 50;
```

In code



```
int a = 24;  
int b = 50;
```

In code

it looks like this!



```
int a = 24;  
int b = 50;  
a = b;
```

In code

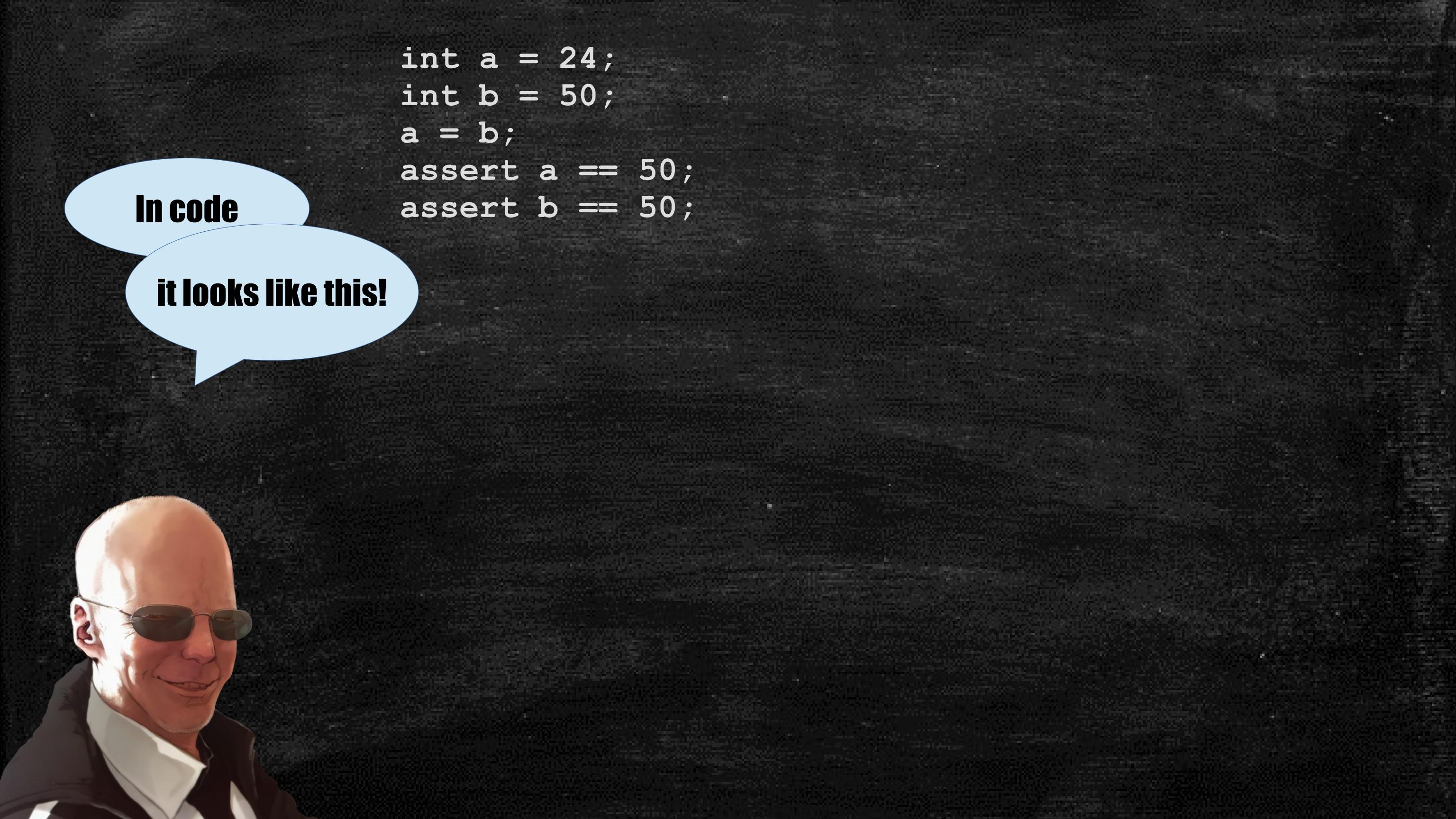
it looks like this!



```
int a = 24;  
int b = 50;  
a = b;  
assert a == 50;
```

In code

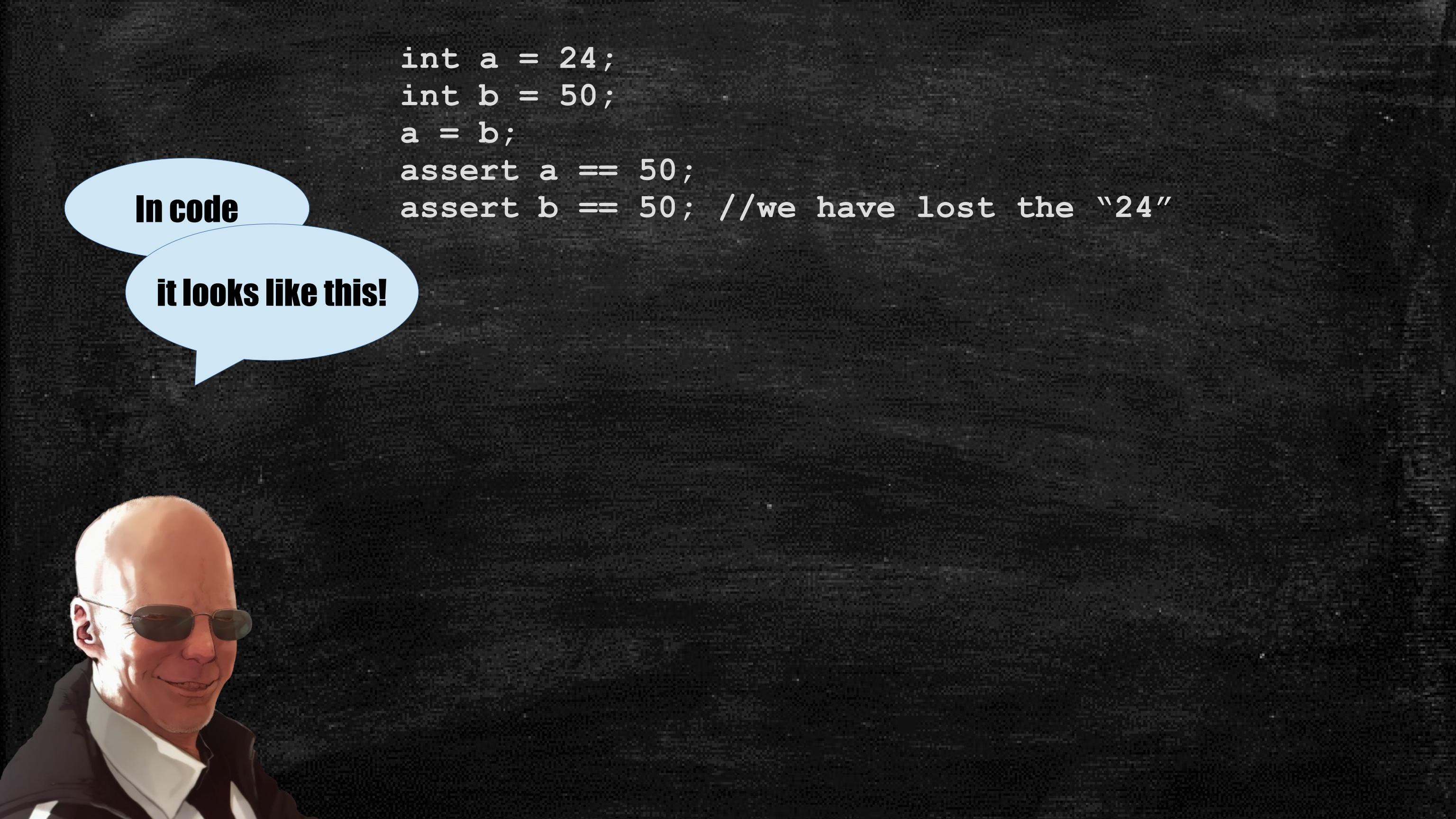
it looks like this!



```
int a = 24;  
int b = 50;  
a = b;  
assert a == 50;  
assert b == 50;
```

In code

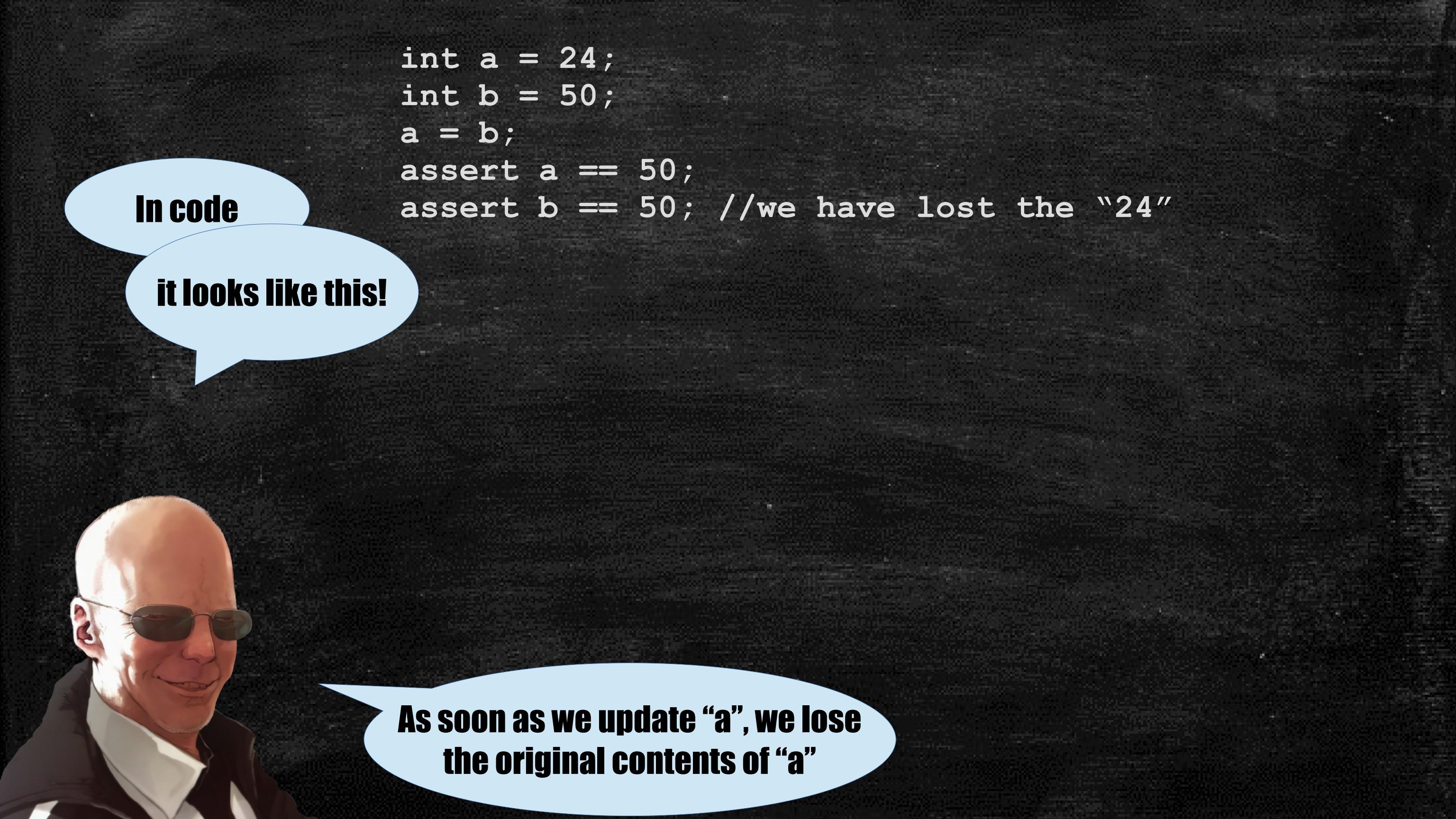
it looks like this!



```
int a = 24;  
int b = 50;  
a = b;  
assert a == 50;  
assert b == 50; //we have lost the "24"
```

In code

it looks like this!



```
int a = 24;  
int b = 50;  
a = b;  
assert a == 50;  
assert b == 50; //we have lost the "24"
```

In code

it looks like this!

As soon as we update “a”, we lose
the original contents of “a”



A cartoon character with a bald head, wearing sunglasses and a dark suit, stands in front of a chalkboard. He is holding a soccer ball in his left hand and a basketball in his right hand. A speech bubble originates from him.

**As you can see,
we can't do it!**



**As you can see,
we can't do it!**

**Not with only
two variables!**



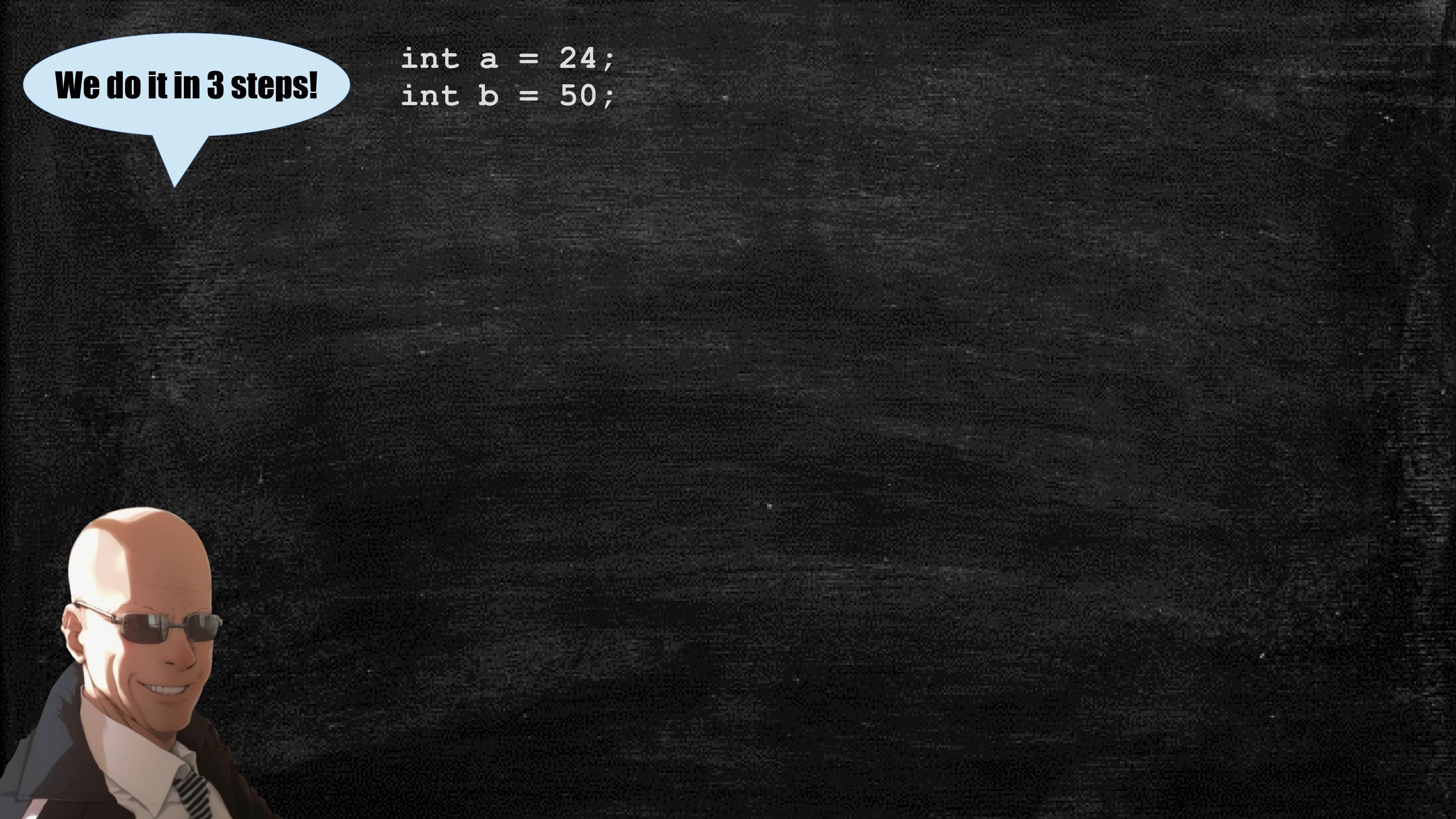
**As you can see,
we can't do it!**

**Not with only
two variables!**

**We need a third
variable to help us!**

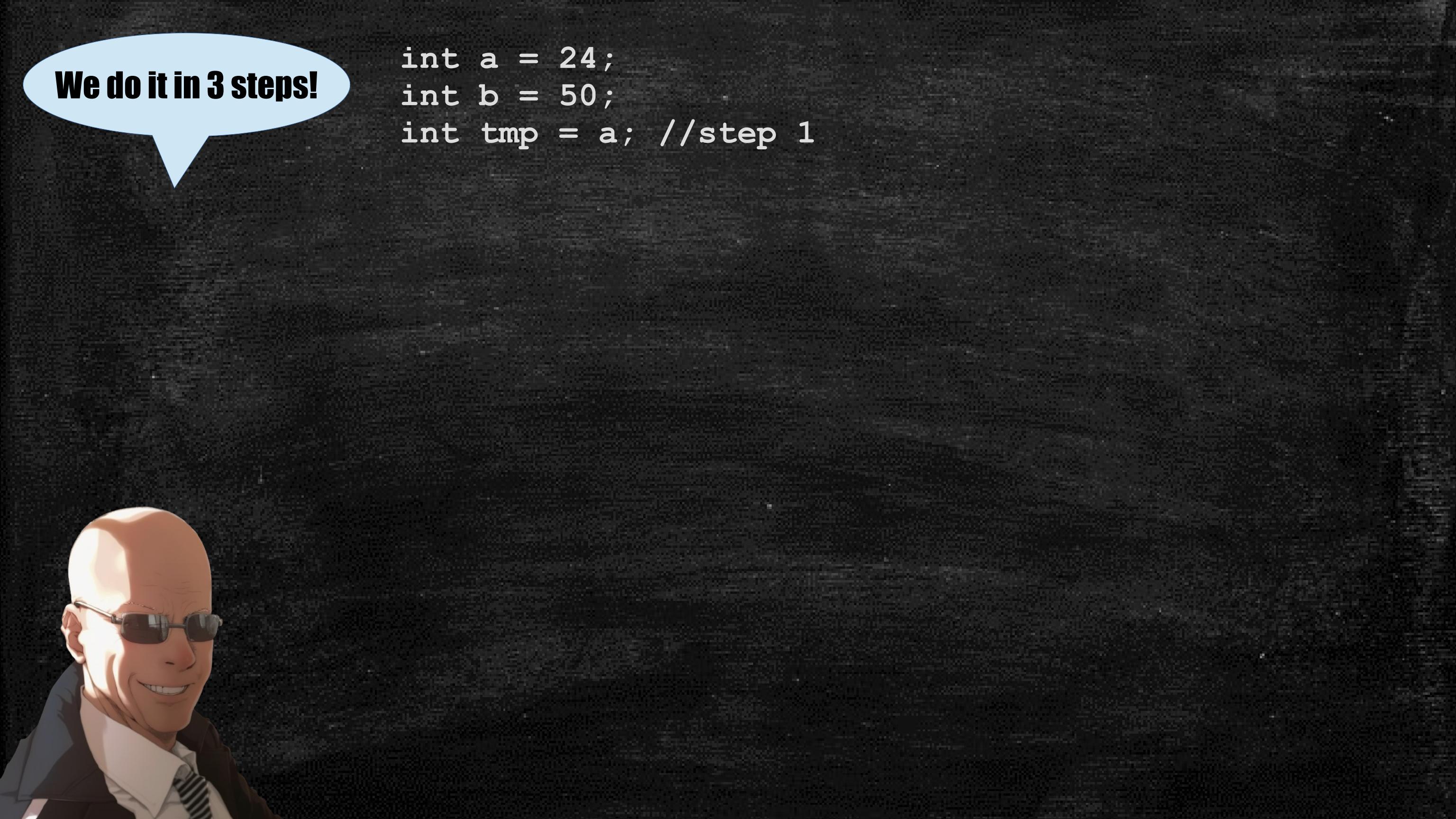
```
int a = 24;  
int b = 50;
```



A cartoon illustration of a bald man wearing dark sunglasses and a suit jacket over a striped shirt. He is smiling and looking towards the right. A light blue speech bubble originates from his mouth and contains the text "We do it in 3 steps!"

We do it in 3 steps!

```
int a = 24;  
int b = 50;
```

A cartoon illustration of a bald man wearing dark sunglasses and a suit jacket over a striped shirt. He is smiling and looking towards the right. A light blue speech bubble originates from his mouth and extends upwards and to the left.

We do it in 3 steps!

```
int a = 24;  
int b = 50;  
int tmp = a; //step 1
```



We do it in 3 steps!

```
int a = 24;  
int b = 50;  
int tmp = a; //step 1
```

**First we save the original value of “a”
in a new temporary local variable**



We do it in 3 steps!

```
int a = 24;  
int b = 50;  
int tmp = a; //step 1  
a = b; //step 2
```

**First we save the original value of “a”
in a new temporary local variable**



We do it in 3 steps!

```
int a = 24;  
int b = 50;  
int tmp = a; //step 1  
a = b; //step 2
```

**First we save the original value of “a”
in a new temporary local variable**

Then we write the content of “b” into “a”



We do it in 3 steps!

```
int a = 24;  
int b = 50;  
int tmp = a; //step 1  
a = b; //step 2  
b = tmp; //step 3
```

**First we save the original value of “a”
in a new temporary local variable**

Then we write the content of “b” into “a”



We do it in 3 steps!

```
int a = 24;  
int b = 50;  
int tmp = a; //step 1  
a = b; //step 2  
b = tmp; //step 3
```

**First we save the original value of “a”
in a new temporary local variable**

Then we write the content of “b” into “a”

Finally we write “tmp” into “b”



We do it in 3 steps!

```
int a = 24;  
int b = 50;  
int tmp = a; //step 1  
a = b; //step 2  
b = tmp; //step 3  
assert a == 50;
```

**First we save the original value of “a”
in a new temporary local variable**

Then we write the content of “b” into “a”

Finally we write “tmp” into “b”



We do it in 3 steps!

```
int a = 24;  
int b = 50;  
int tmp = a; //step 1  
a = b; //step 2  
b = tmp; //step 3  
assert a == 50;  
assert b == 24;
```

**First we save the original value of “a”
in a new temporary local variable**

Then we write the content of “b” into “a”

Finally we write “tmp” into “b”





**As you can see, it can not be
done with only two variable**



**As you can see, it can not be
done with only two variable**

It is mathematically proven



As you can see, it can not be done with only two variable

It is mathematically proven

You need a third variable!





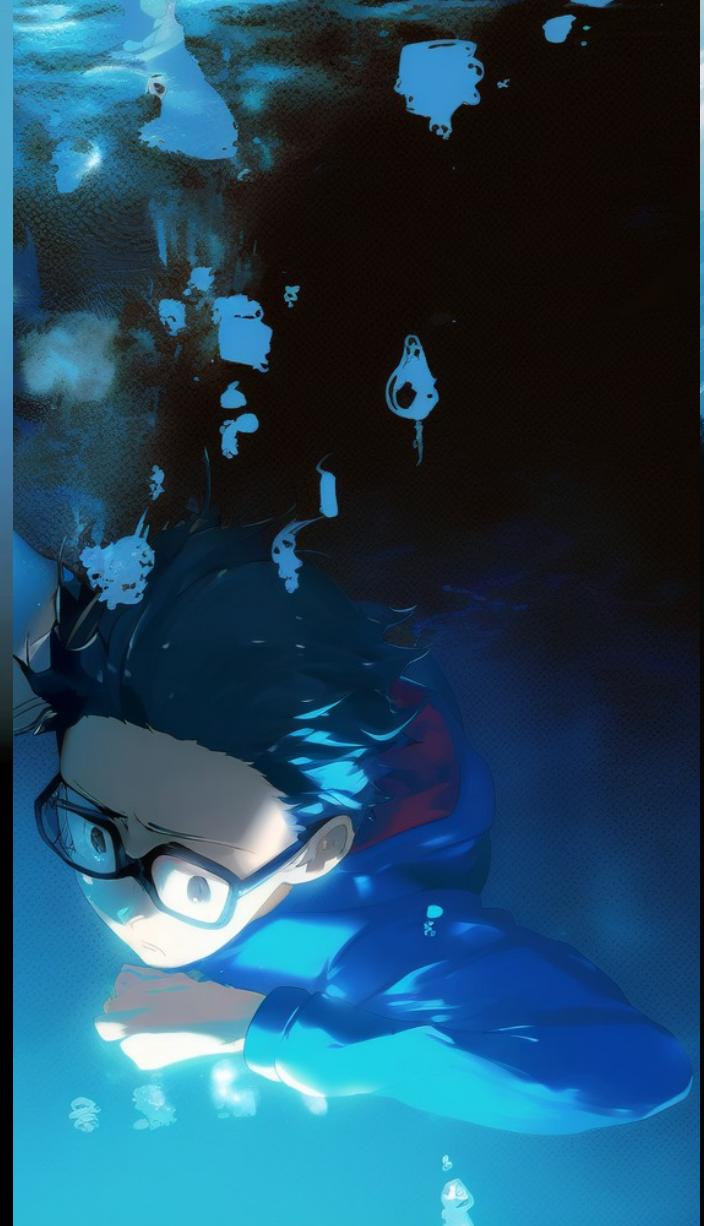
Something is wrong here

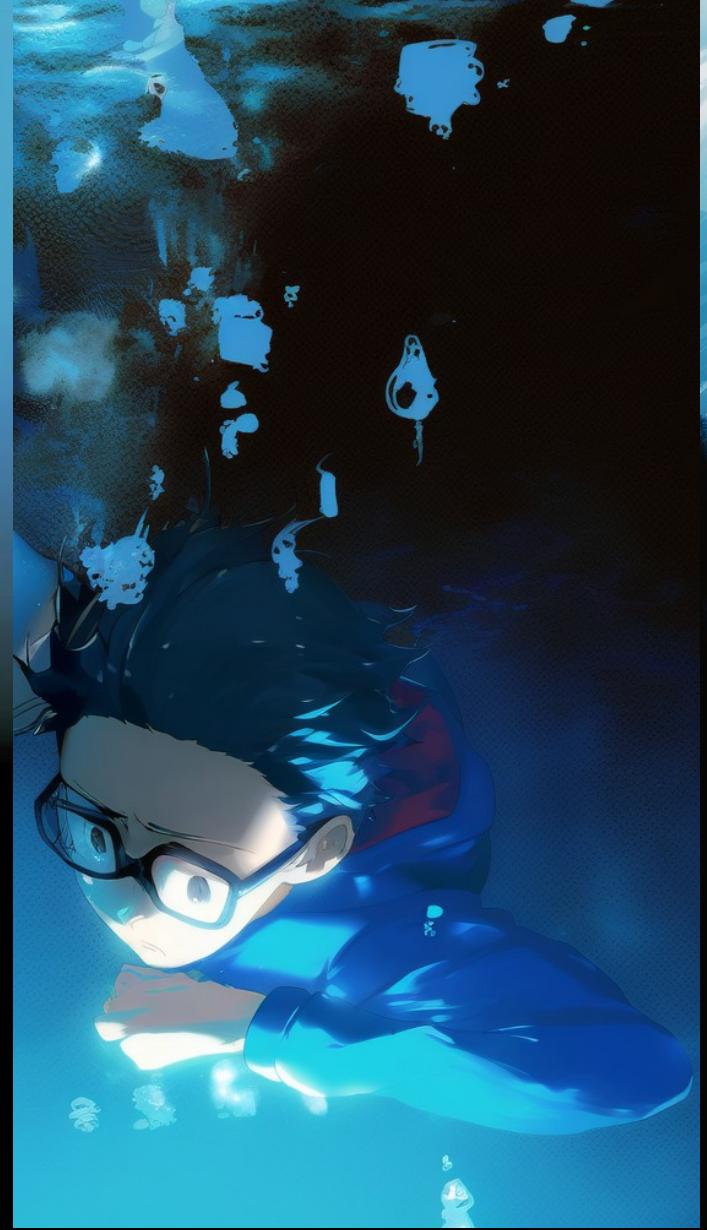


Something is wrong here



I'm seeing stuff

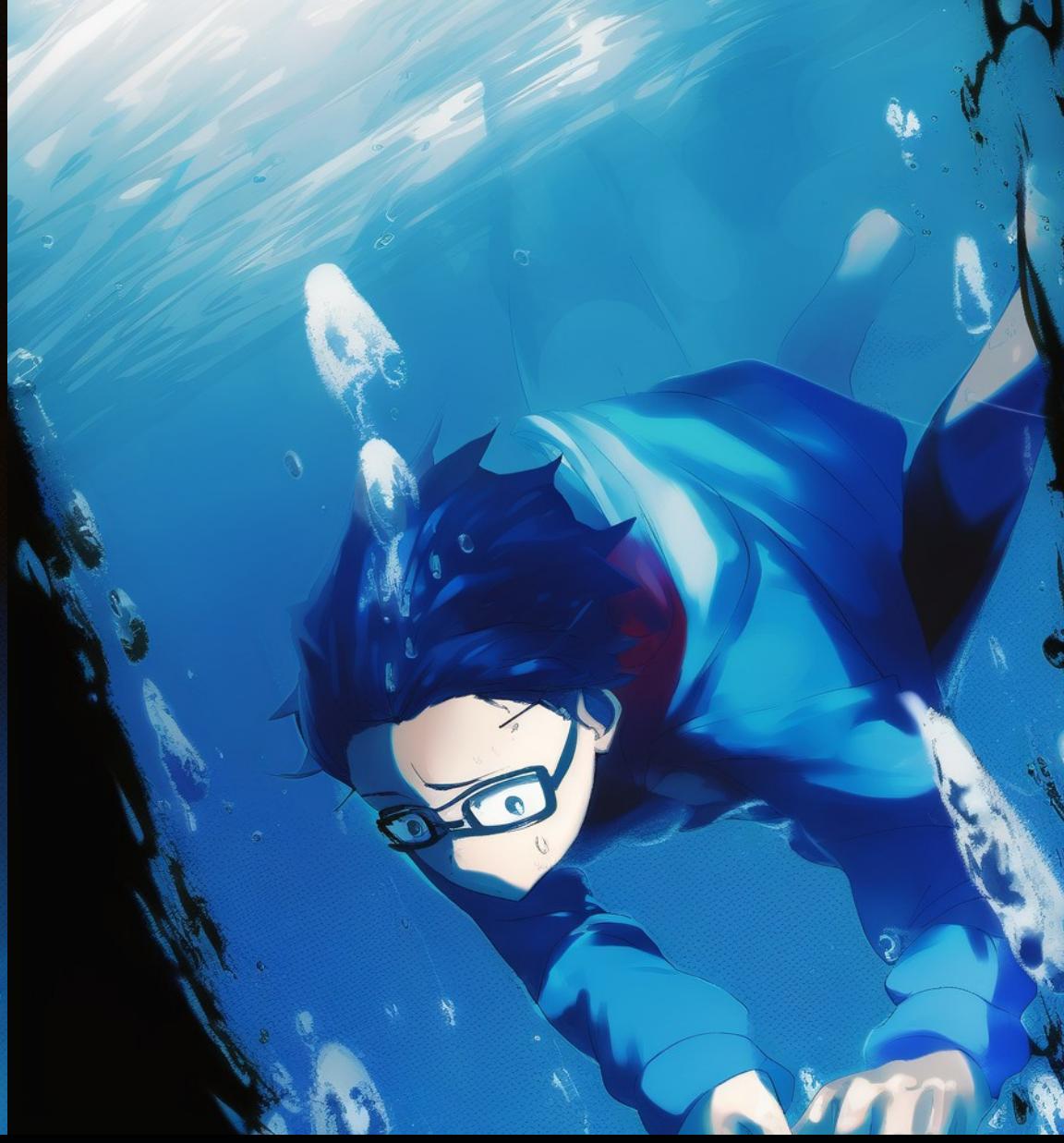
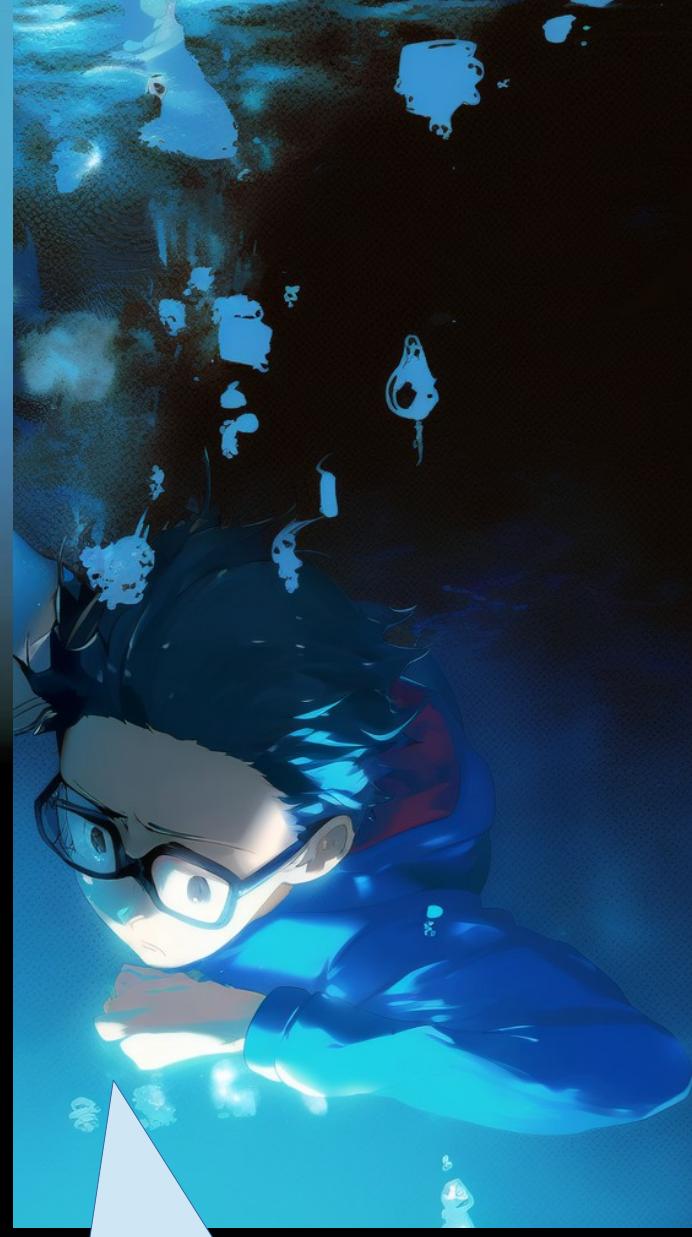




What?

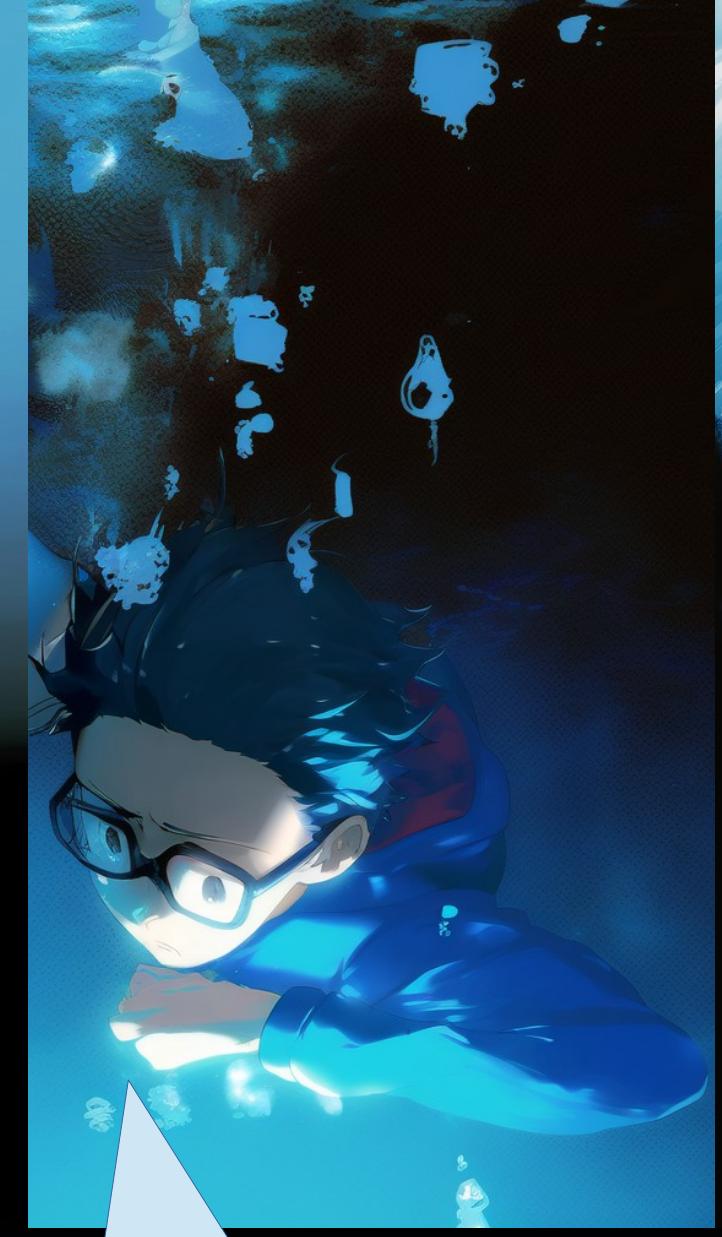


What?
I'm under water?





What?



I'm under water?



What is this?



Two glasses?
With water?



Two glasses?
With water?

But, I'm already
under water



Two glasses?
With water?

But, I'm already
under water

I must be
dreaming





Anyway



Anyway

Can I swap
the content

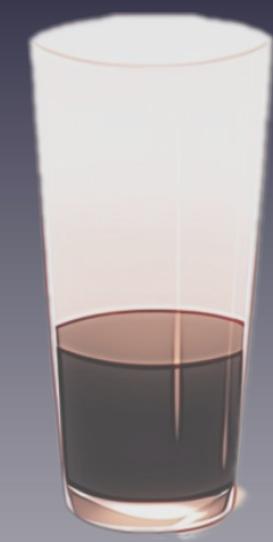


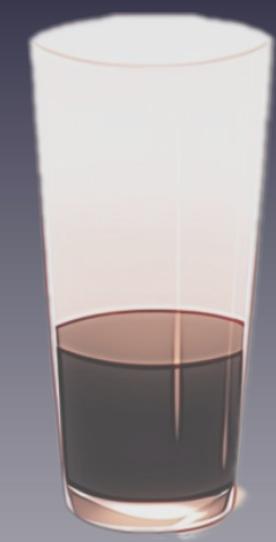
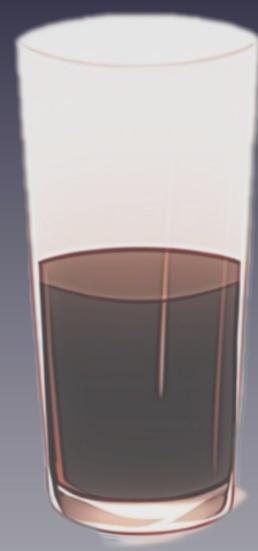
Anyway

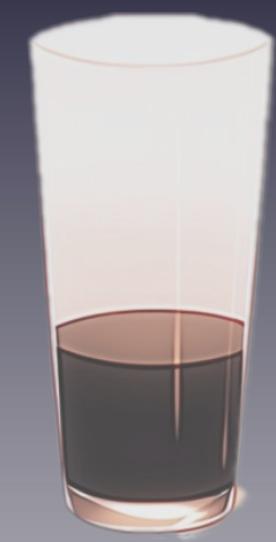
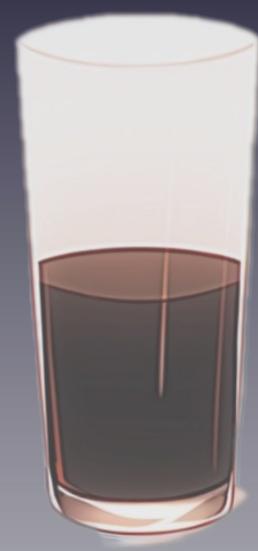
Can I swap
the content

without a
third glass?





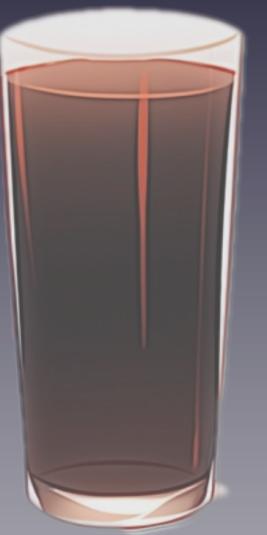
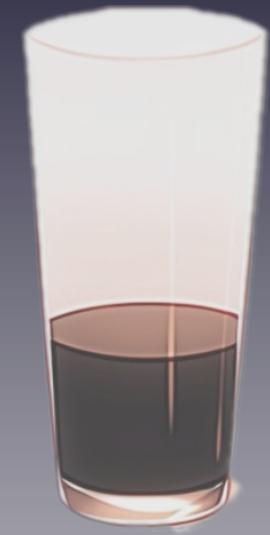
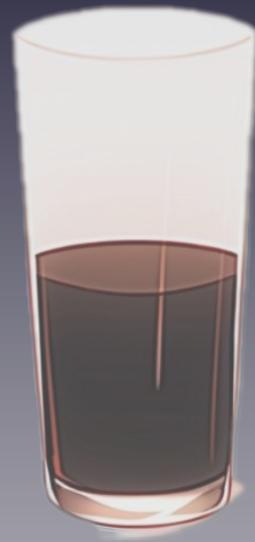






If I add the two glasses
in the first one





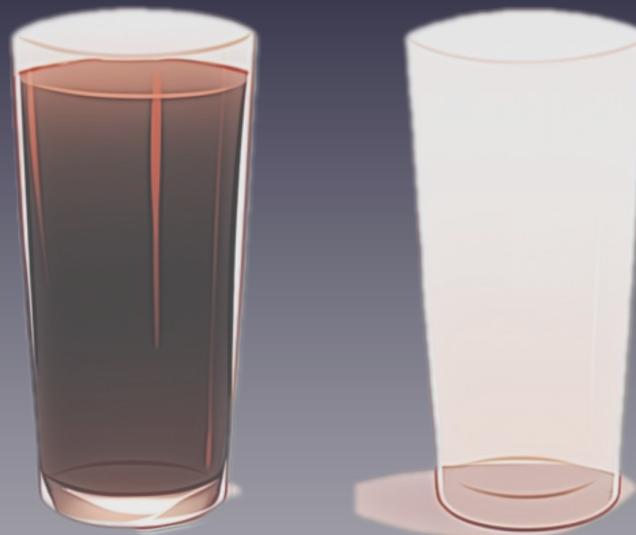
If I add the two glasses
in the first one



Would the second
glass get empty?

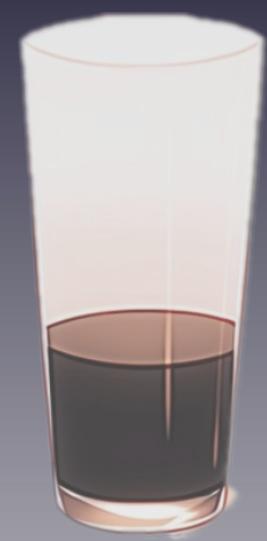


```
int a = 24;
```

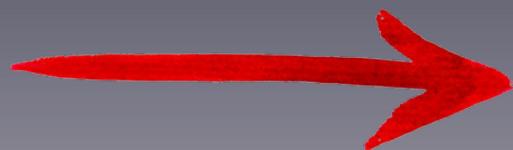


If I add the two glasses
in the first one

Would the second
glass get empty?



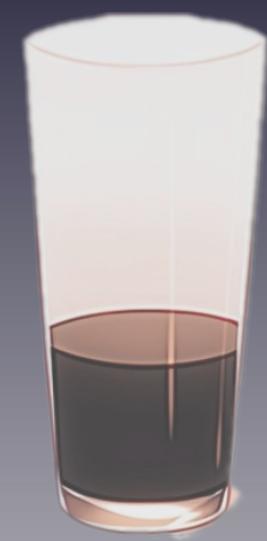
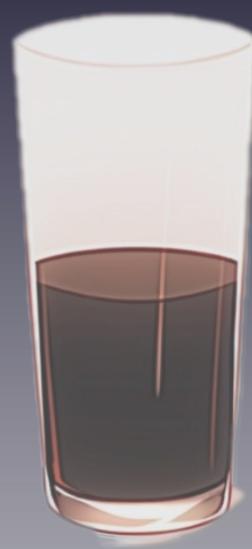
```
int a = 24;  
int b = 50;
```



If I add the two glasses
in the first one



Would the second
glass get empty?



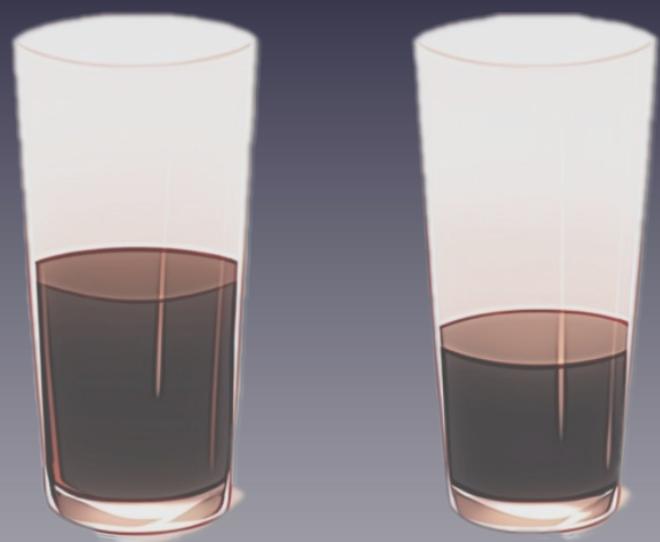
```
int a = 24;  
int b = 50;  
a = a + b;
```



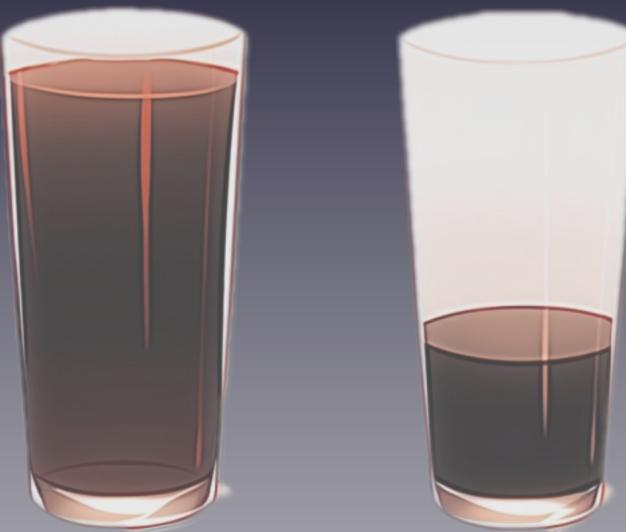
If I add the two glasses
in the first one

Would the second
glass get empty?





```
int a = 24;  
int b = 50;  
a = a + b;
```



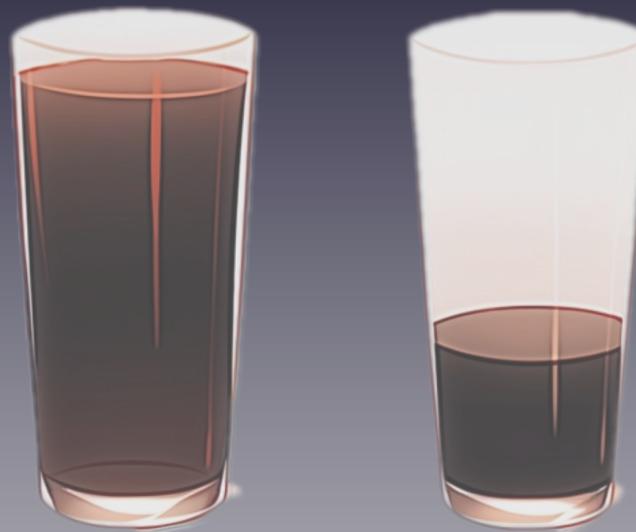
If I add the two glasses
in the first one



Would the second
glass get empty?



```
int a = 24;  
int b = 50;  
a = a + b;
```

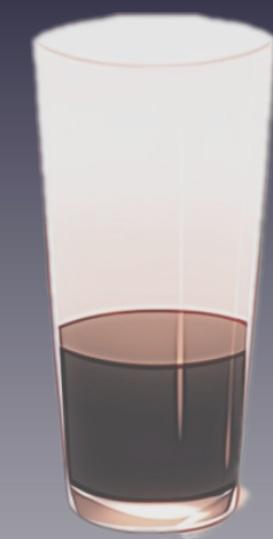
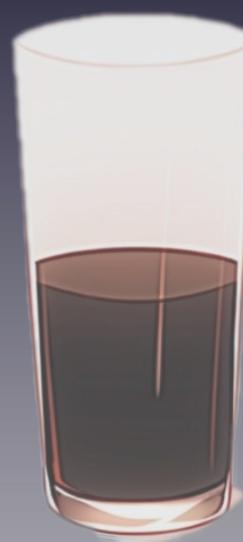


If I add the two glasses
in the first one



Would the second
glass get empty?

No, this would
not update b



```
int a = 24;  
int b = 50;  
a = a + b;
```



If I add the two glasses
in the first one

Would the second
glass get empty?

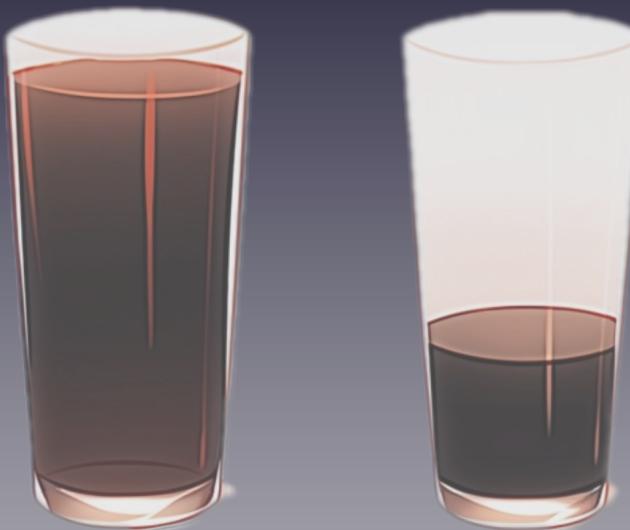
No, this would
not update b

Values are not shuffled
around but duplicated



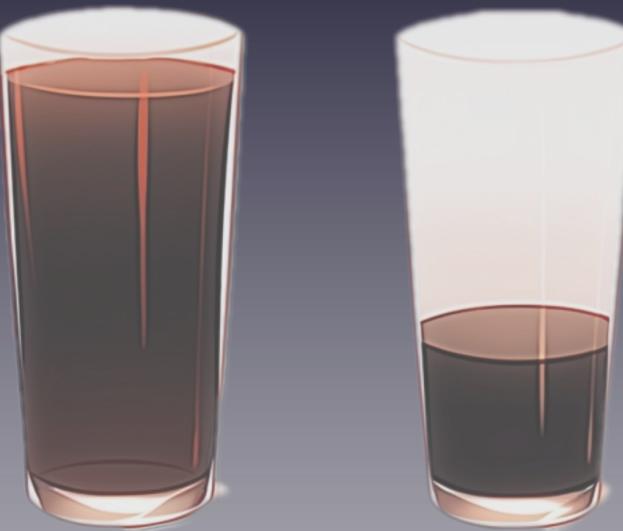


```
int a = 24;  
int b = 50;  
a = a + b;
```





```
int a = 24;  
int b = 50;  
a = a + b;
```



That is, even after
adding 'a' and 'b'





```
int a = 24;  
int b = 50;  
a = a + b;
```



That is, even after
adding 'a' and 'b'

All the information
is still in the system!





```
int a = 24;  
int b = 50;  
a = a + b;
```



That is, even after
adding 'a' and 'b'

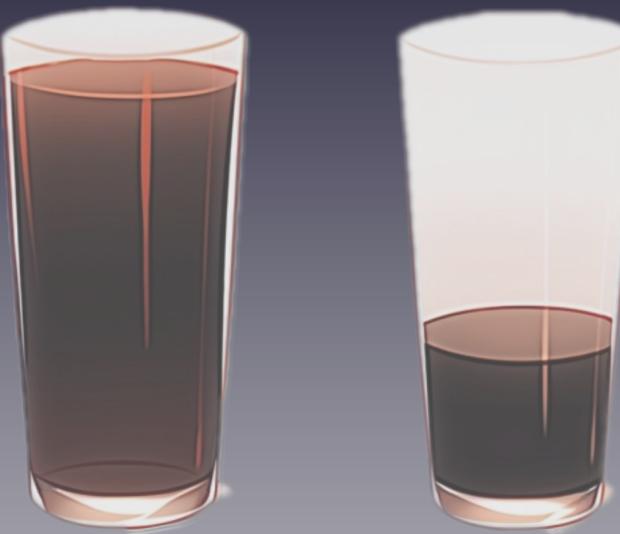
All the information
is still in the system!

'b' is still in 'b'





```
int a = 24;  
int b = 50;  
a = a + b;
```



That is, even after
adding 'a' and 'b'

All the information
is still in the system!

'b' is still in 'b'

The original 'a' is just
the current 'a' minus 'b'

```
int a = 24;  
int b = 50;
```

a = a + b;



```
int a = 24;  
int b = 50;
```

a = a + b;



So, in three steps
I can swap them!



```
int a = 24;  
int b = 50;
```

a = a + b;



So, in three steps
I can swap them!

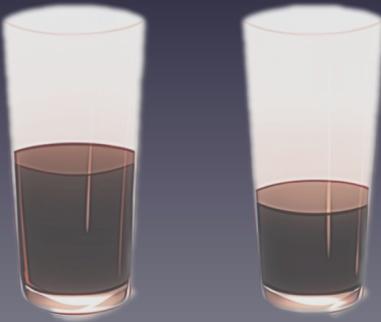
Pause the video. Can you solve this one?

What are the 3 steps that
can solve this problem?

This message will disappear shortly.
You can pause the video after that.



```
int a = 24;  
int b = 50;
```

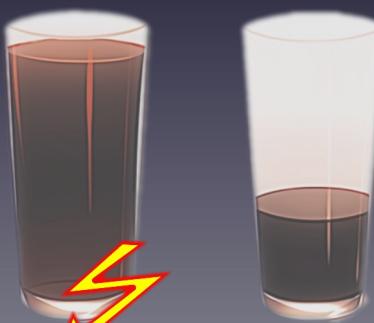
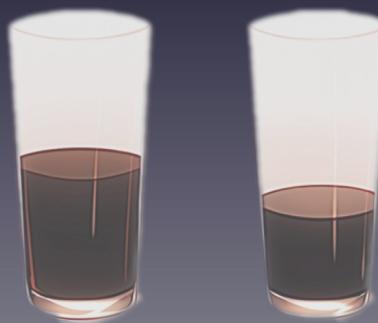


So, in three steps
I can swap them!



```
int a = 24;  
int b = 50;
```

a = a + b;

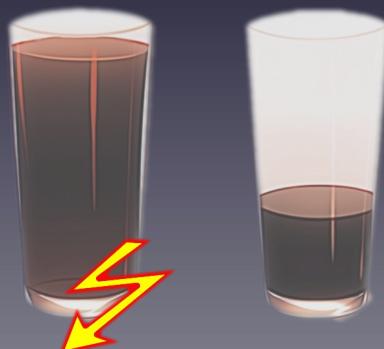
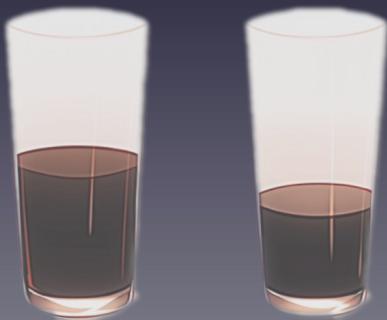


So, in three steps
I can swap them!

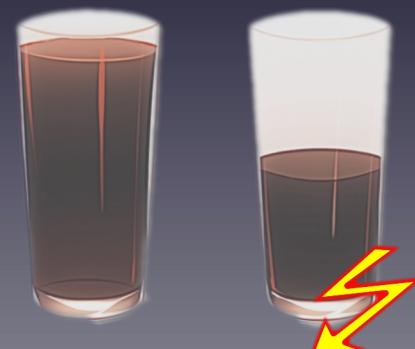


```
int a = 24;  
int b = 50;
```

a = a + b;



b = a - b;

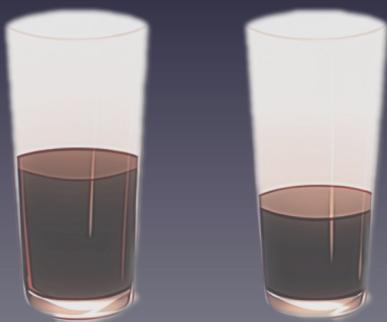


So, in three steps
I can swap them!

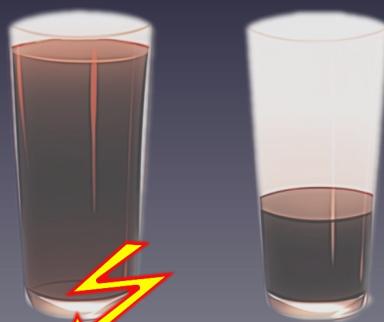


```
int a = 24;  
int b = 50;
```

a = a + b;



b = a - b;



a = a - b;



So, in three steps
I can swap them!





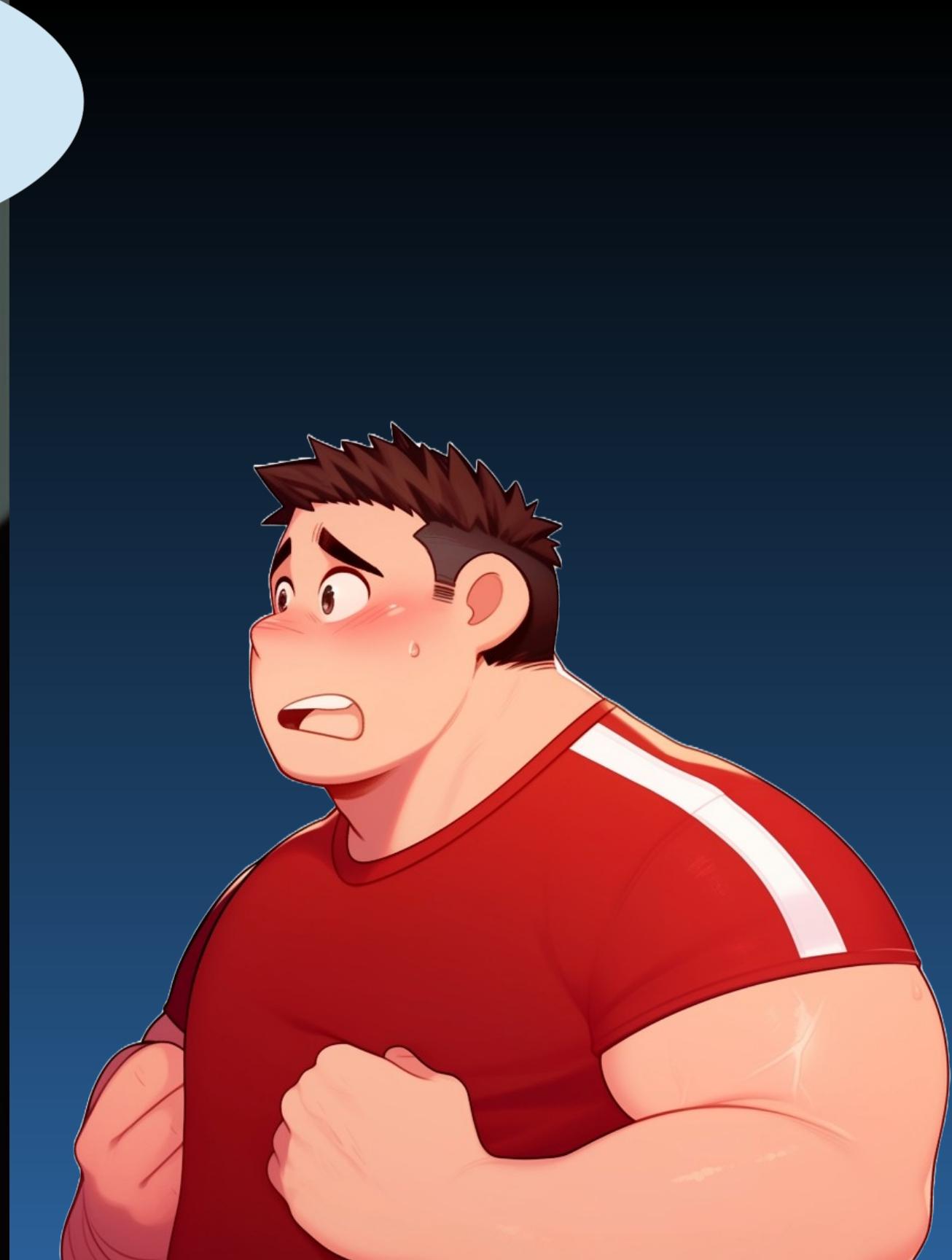




! Sorry !



! Hey, .. !



! Hey, .. !

Dany, stop!
What are you doing?



A cartoon illustration of a bald man with glasses and a suit, standing in front of a chalkboard in a classroom. He is smiling and gesturing with his hands. A speech bubble originates from him.

**Yes?
Do you have a question
on this basic material?**





I CAN do it!

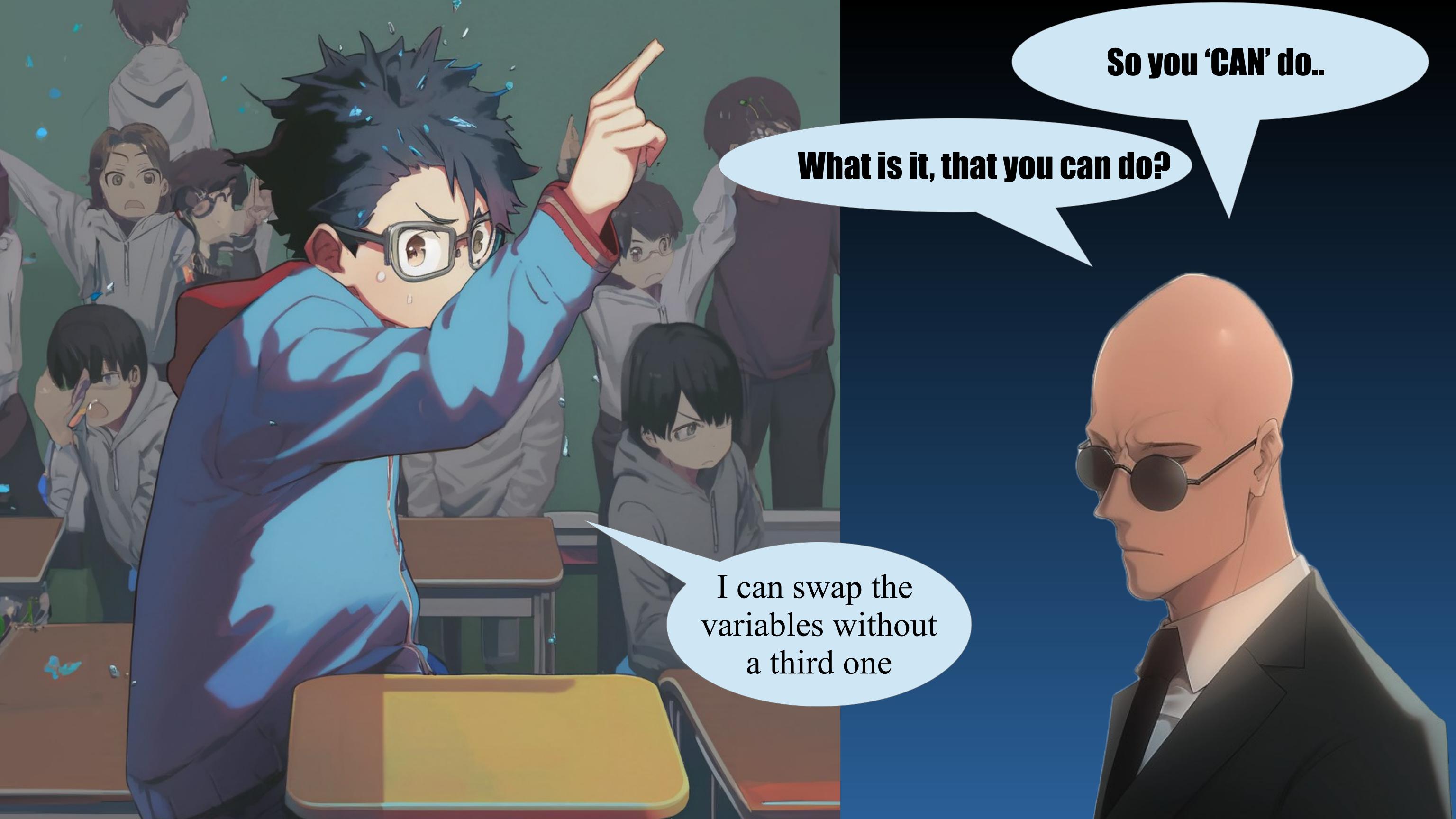




So you 'CAN' do..







So you 'CAN' do..

What is it, that you can do?

I can swap the
variables without
a third one





Really?

A cartoon illustration of a balding man with glasses, wearing a white shirt and a dark blazer, standing in front of a chalkboard. He is smiling and looking towards the right. A large blue speech bubble originates from his mouth, containing the text "Really? Do not make us laugh."

**Really?
Do not make us laugh.**



**Really?
Do not make us laugh.
It is mathematically proven.**



**Really?
Do not make us laugh.
It is mathematically proven.
It can not be done!**

A cartoon illustration of a bald man with glasses and a suit, standing in front of a chalkboard. He is smiling and gesturing with his hands. A large speech bubble originates from him.

**Really?
Do not make us laugh.
It is mathematically proven.
It can not be done!**

**Now, let's move on
with the lecture..**





Dany, please stop!
It is going to get dangerous!







WRITE IT DOWN!



WRITE IT DOWN!

$a = a + b;$



WRITE IT DOWN!

$a = a + b;$



$b = a - b;$



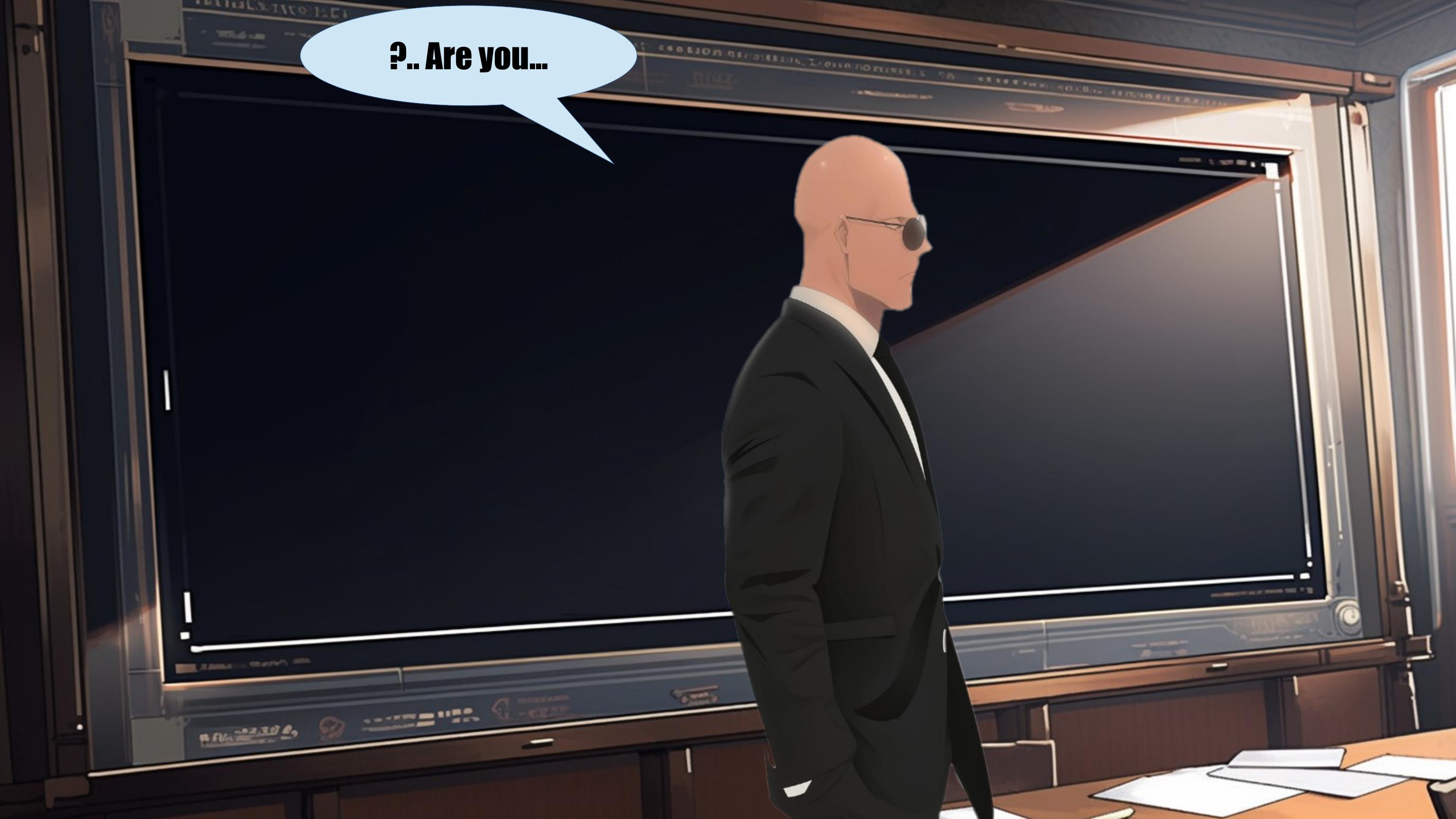
WRITE IT DOWN!

$$a = a + b;$$

$$b = a - b;$$

$$a = a - b;$$





P.. Are you...

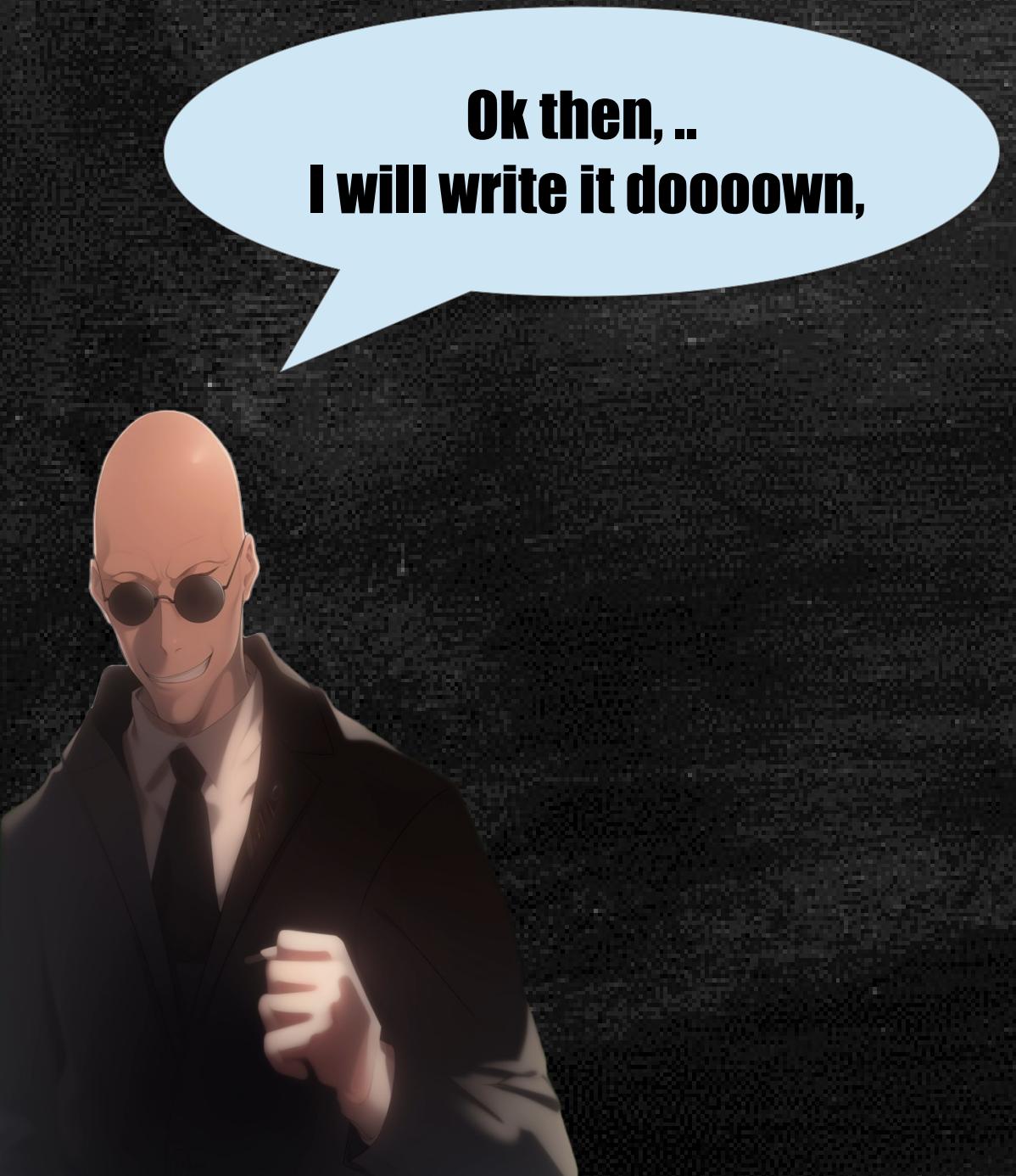
P.. Are you...

For real?

```
int a = 24;  
int b = 50;
```



```
int a = 24;  
int b = 50;
```



**Ok then, ..
I will write it doooown,**



```
int a = 24;  
int b = 50;  
  
a = a+b;  
b = a-b;  
a = a-b;
```

```
int a = 24;  
int b = 50;  
  
a = a+b;  
b = a-b;  
a = a-b;
```



**So that I can shooow you that
it does nooot work**

```
int a = 24;  
int b = 50;  
  
a = a+b;  
b = a-b;  
a = a-b;
```

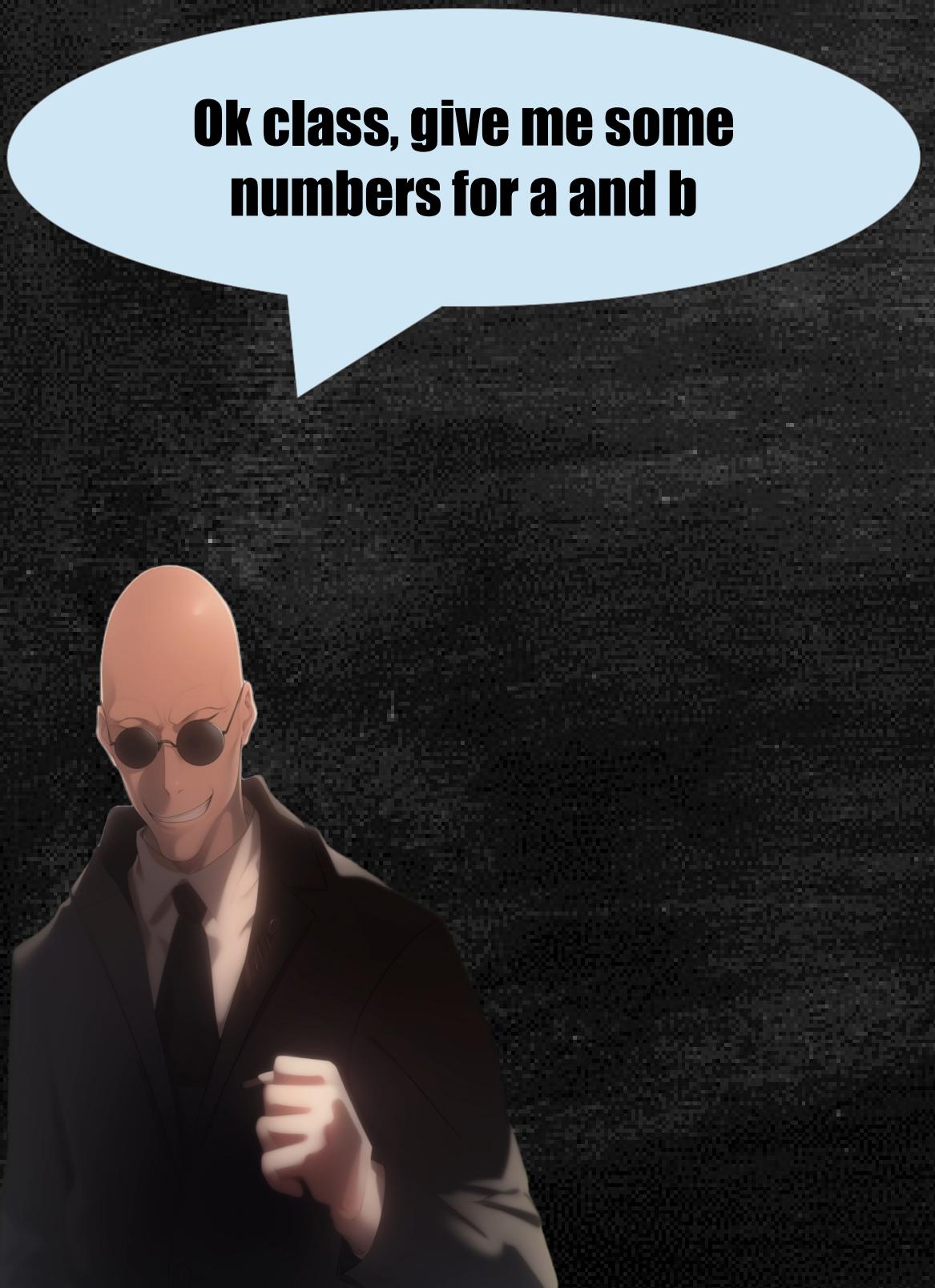


**So that I can shooow you that
it does nooot work**

**And then we will all laugh at you
and your dumb arrogance!**

A 3D rendered character with a smooth, light brown complexion stands in the lower-left corner against a dark, textured background. He is wearing a dark grey suit jacket over a white collared shirt and a black tie. He has a pair of dark sunglasses perched on his forehead and is looking directly at the viewer with a neutral expression. His right hand is clenched into a fist, which is held out towards the camera, emitting a faint, glowing yellow light.

```
int a = ??;  
int b = ??;  
  
a = a+b;//  
b = a-b;//  
a = a-b;//
```

A 3D rendered character of a bald man with a thin mustache. He is wearing dark sunglasses and a dark suit jacket over a white shirt and black tie. He is holding a small, glowing orange rectangular device in his right hand. A light blue speech bubble originates from his mouth.

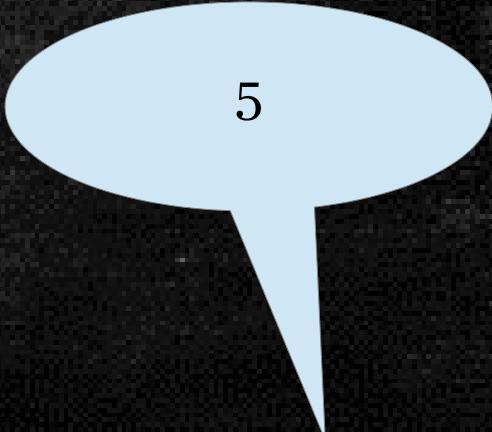
**Ok class, give me some
numbers for a and b**

```
int a = ??;  
int b = ??;  
  
a = a+b;//  
b = a-b;//  
a = a-b;//
```



**Ok class, give me some
numbers for a and b**

```
int a = 5;  
int b = ??;  
  
a = a+b;//  
b = a-b;//  
a = a-b;//
```



5



**Ok class, give me some
numbers for a and b**

```
int a = 5;  
int b = 12;
```

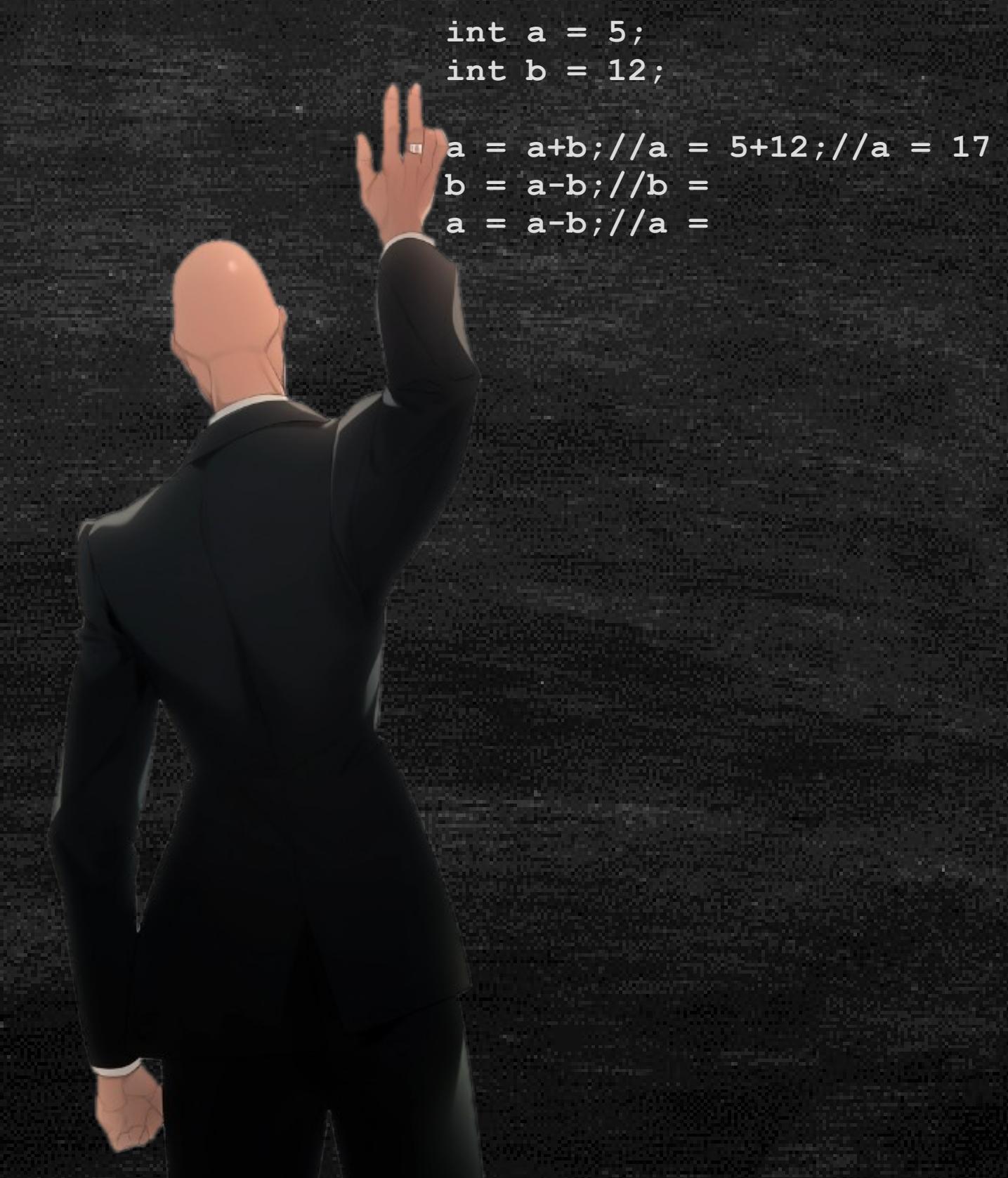
```
a = a+b;//  
b = a-b;//  
a = a-b;//
```

12

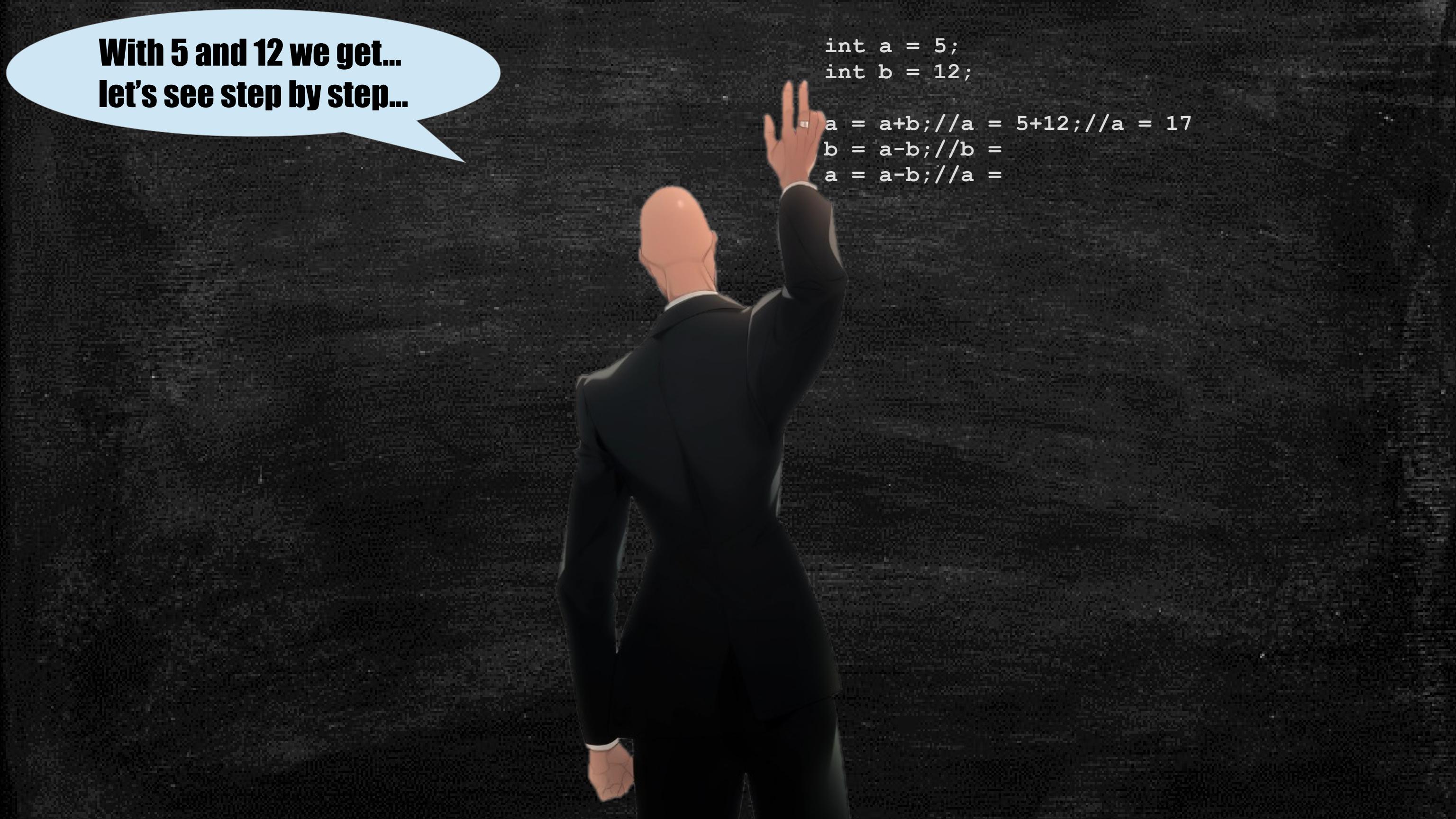
5



```
int a = 5;  
int b = 12;  
  
a = a+b;//a =  
b = a-b;//b =  
a = a-b;//a =
```

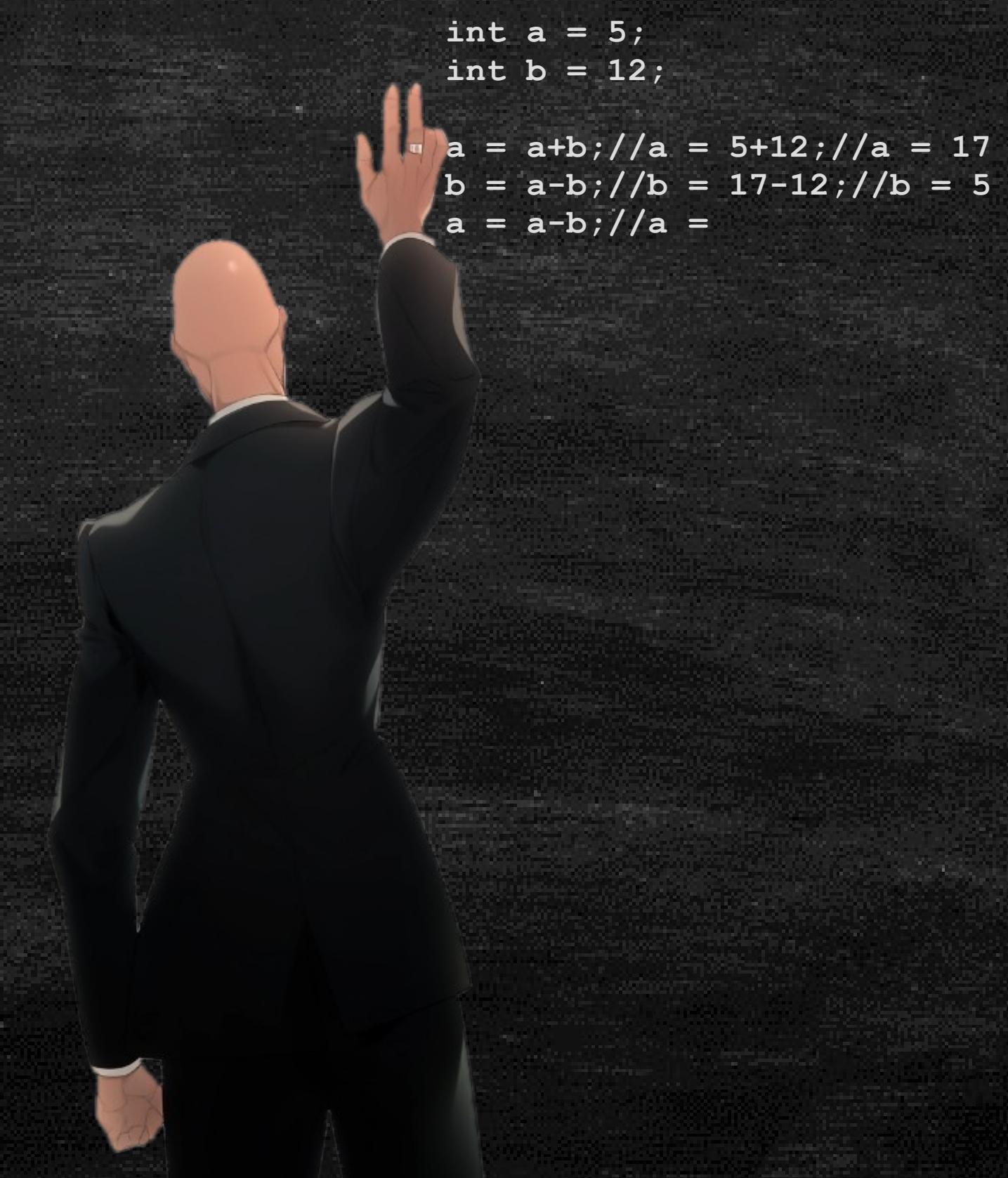


```
int a = 5;  
int b = 12;  
  
a = a+b;//a = 5+12;//a = 17  
b = a-b;//b =  
a = a-b;//a =
```

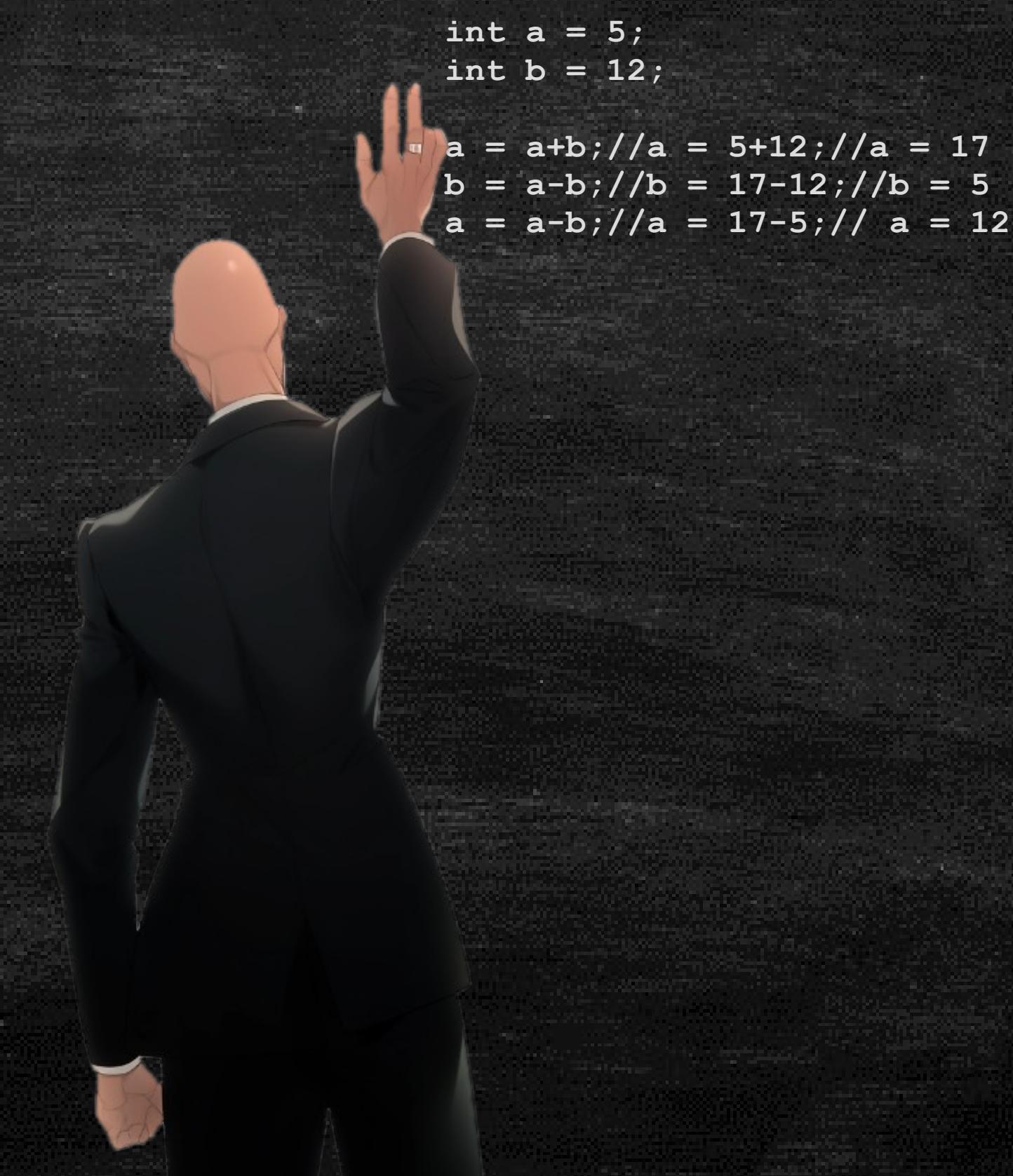
A 3D rendered character of Agent 47, a bald man in a black suit and white shirt, seen from behind. He is standing in a dark, textured environment, possibly a wall or doorway. A light blue speech bubble is positioned above his head.

**With 5 and 12 we get...
let's see step by step...**

```
int a = 5;  
int b = 12;  
  
a = a+b;//a = 5+12;//a = 17  
b = a-b;//b =  
a = a-b;//a =
```

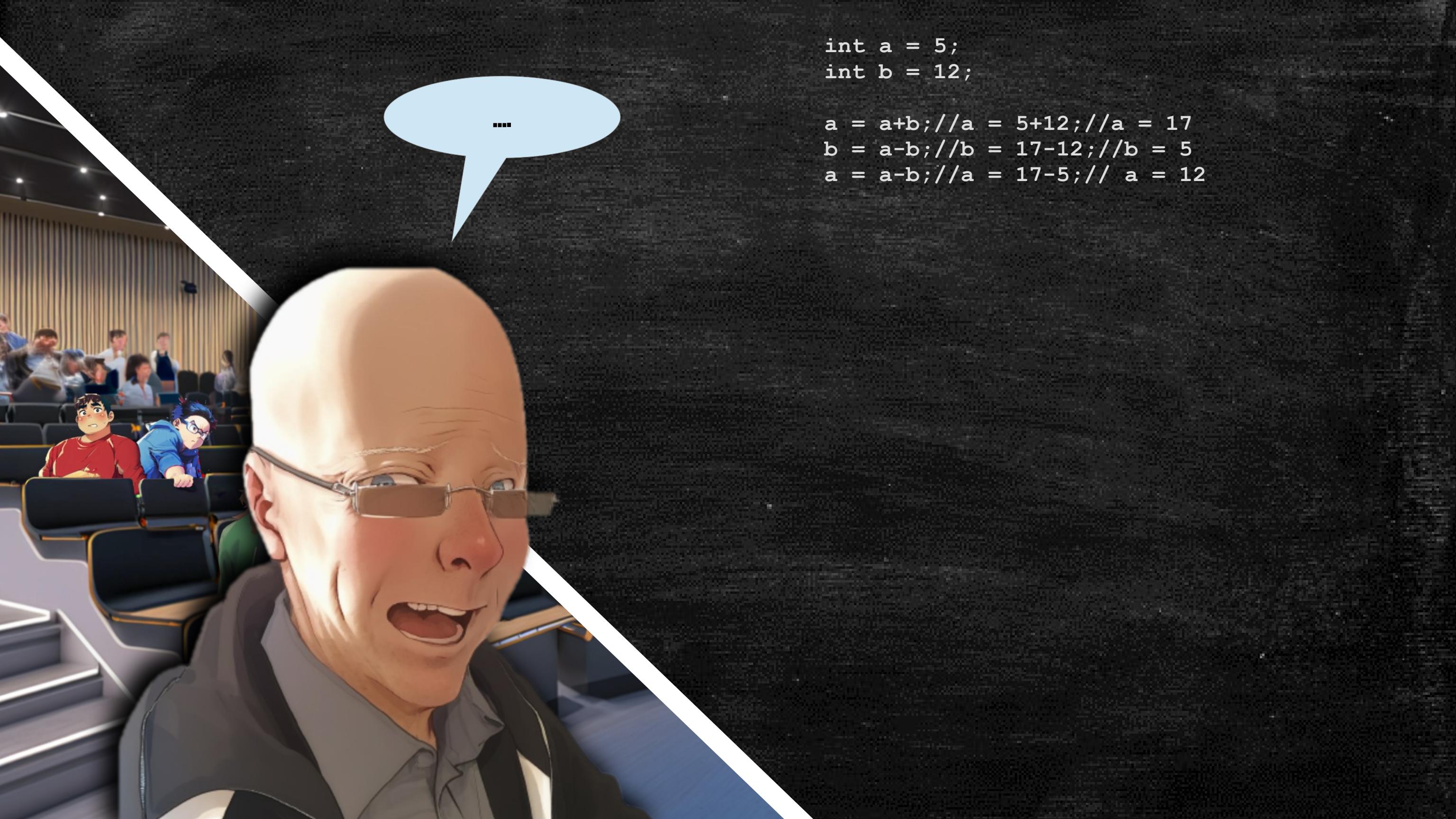


```
int a = 5;  
int b = 12;  
  
a = a+b;//a = 5+12;//a = 17  
b = a-b;//b = 17-12;//b = 5  
a = a-b;//a =
```

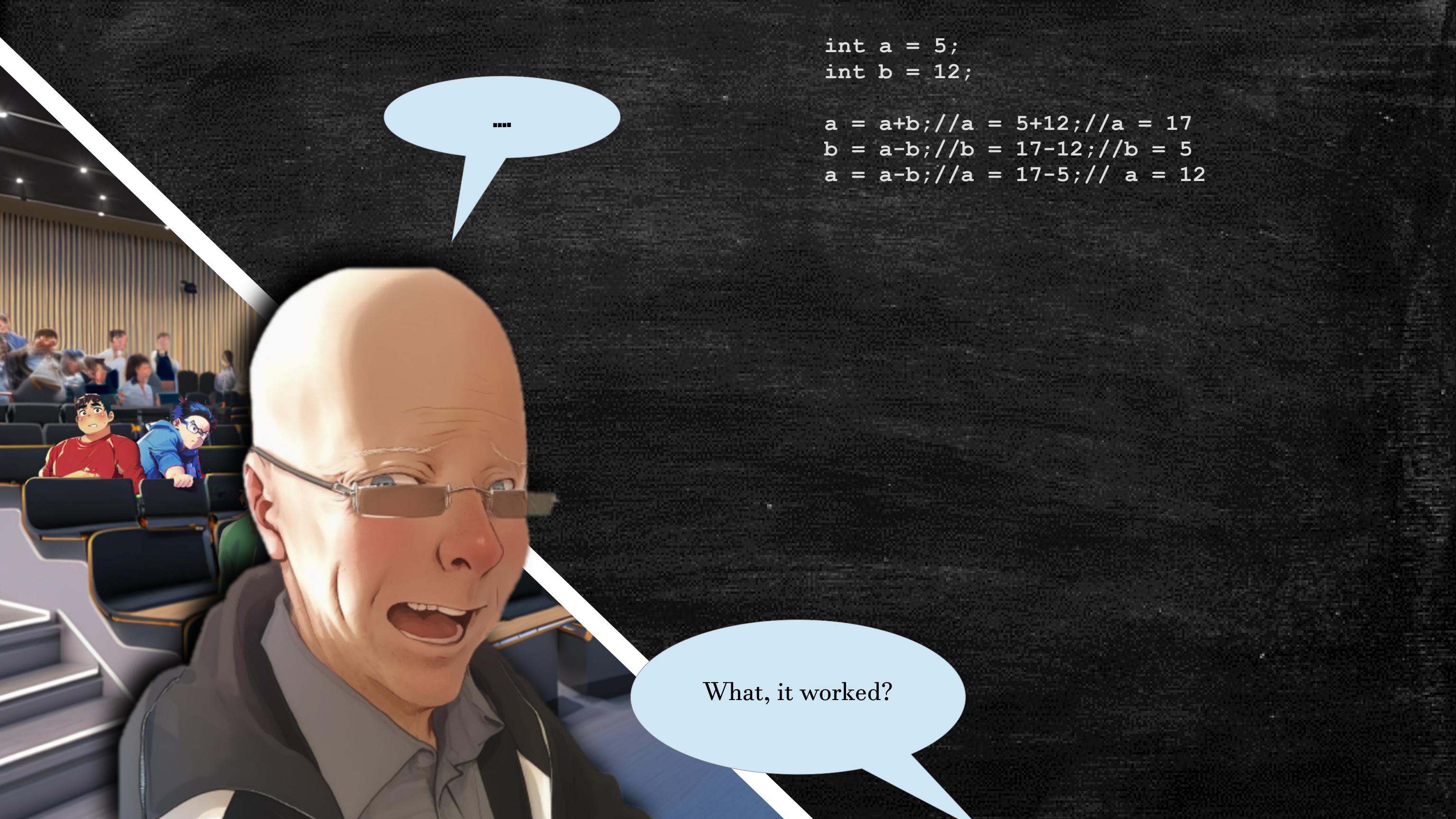


```
int a = 5;
int b = 12;

a = a+b;//a = 5+12;//a = 17
b = a-b;//b = 17-12;//b = 5
a = a-b;//a = 17-5;// a = 12
```

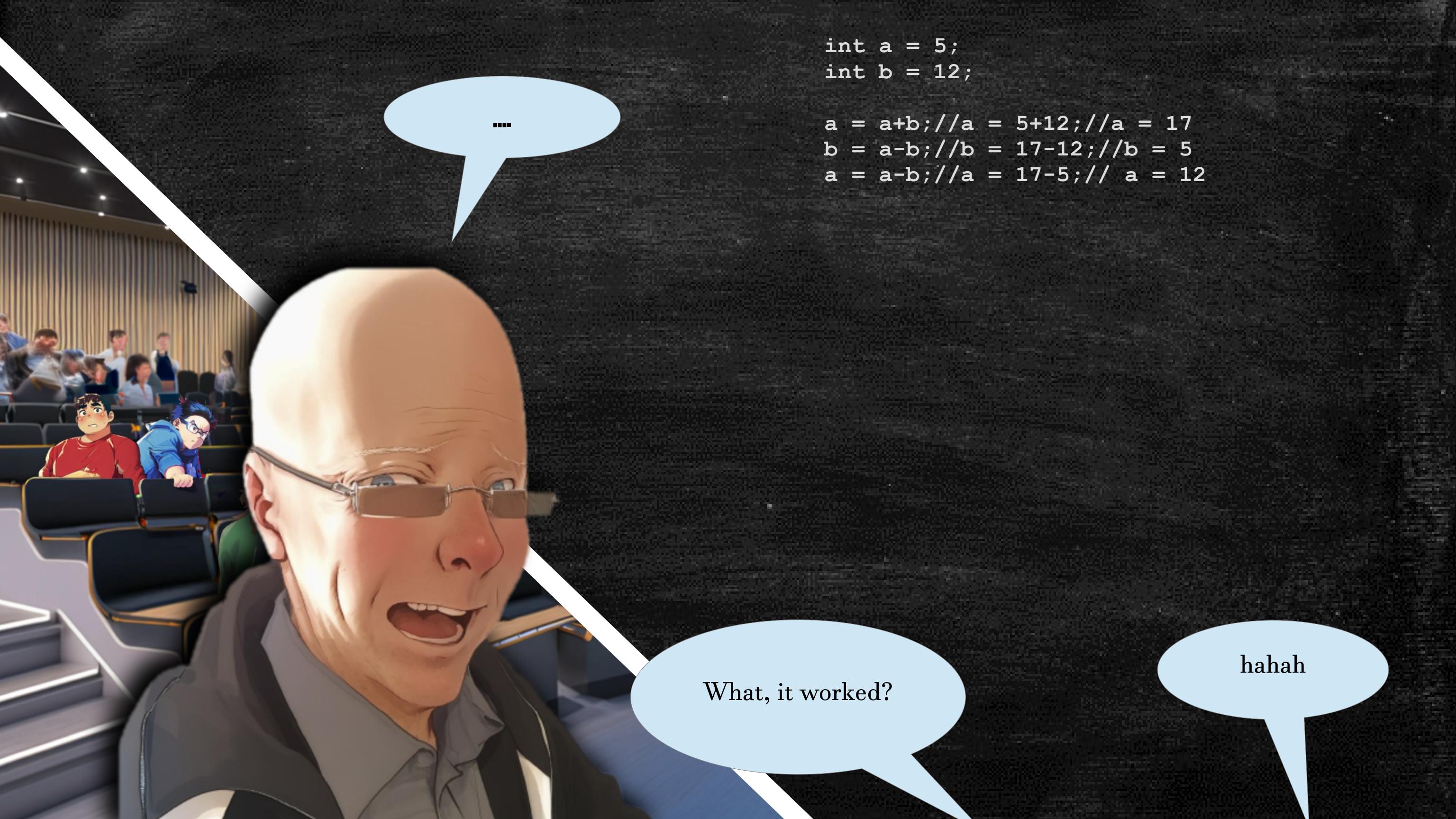


```
int a = 5;  
int b = 12;  
  
a = a+b;//a = 5+12;//a = 17  
b = a-b;//b = 17-12;//b = 5  
a = a-b;//a = 17-5;// a = 12
```



```
int a = 5;  
int b = 12;
```

```
a = a+b;//a = 5+12;//a = 17  
b = a-b;//b = 17-12;//b = 5  
a = a-b;//a = 17-5;// a = 12
```



```
int a = 5;  
int b = 12;
```

```
a = a+b;//a = 5+12;//a = 17  
b = a-b;//b = 17-12;//b = 5  
a = a-b;//a = 17-5;// a = 12
```

What, it worked?

hahah



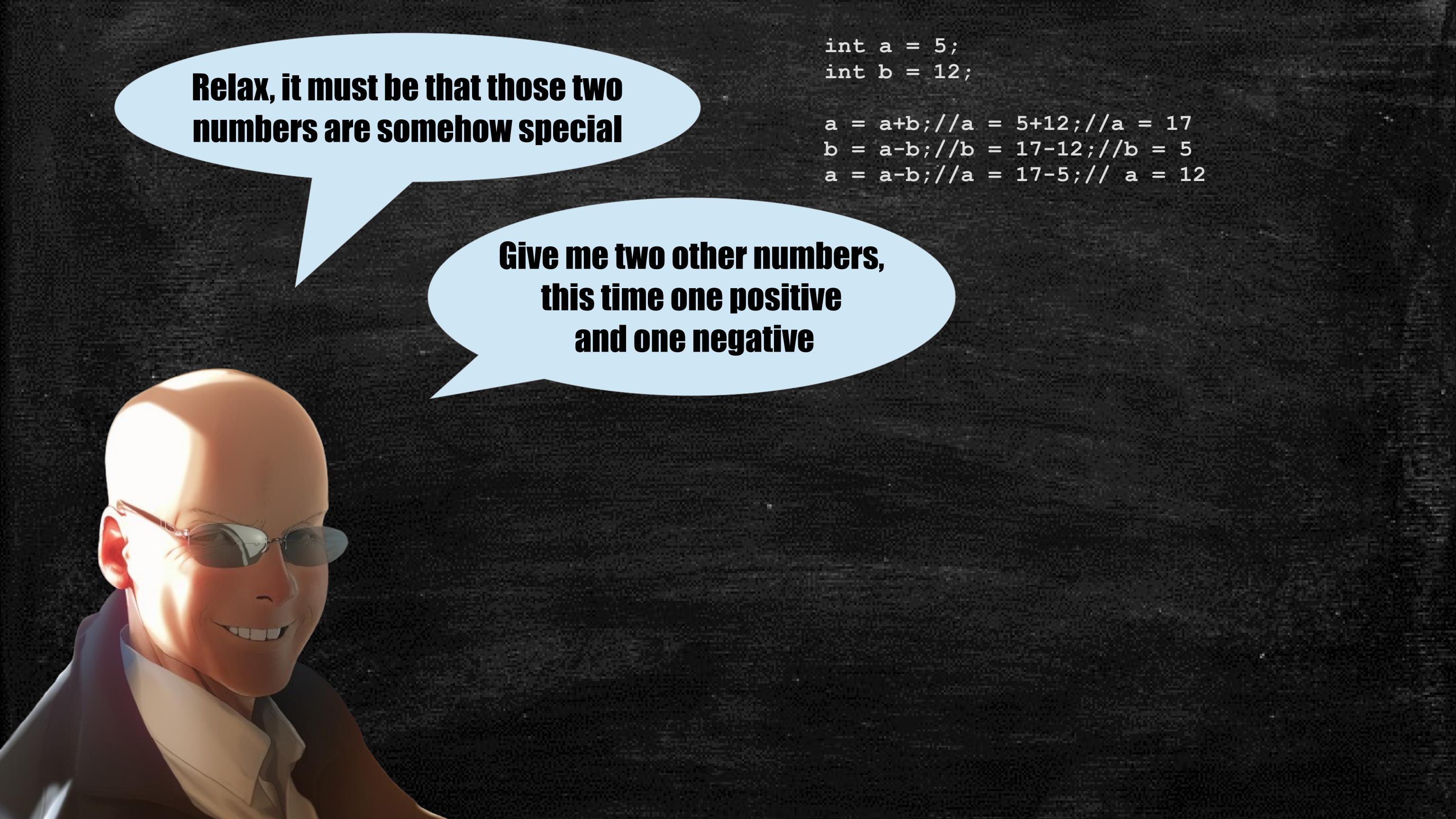
```
int a = 5;
int b = 12;

a = a+b;//a = 5+12;//a = 17
b = a-b;//b = 17-12;//b = 5
a = a-b;//a = 17-5;// a = 12
```

A cartoon illustration of Steve Jobs, featuring a bald head, glasses, and a white shirt. He is smiling and looking towards the right side of the frame.

**Relax, it must be that those two
numbers are somehow special**

```
int a = 5;  
int b = 12;  
  
a = a+b;//a = 5+12;//a = 17  
b = a-b;//b = 17-12;//b = 5  
a = a-b;//a = 17-5;// a = 12
```



Relax, it must be that those two numbers are somehow special

```
int a = 5;  
int b = 12;  
  
a = a+b;//a = 5+12;//a = 17  
b = a-b;//b = 17-12;//b = 5  
a = a-b;//a = 17-5;// a = 12
```

**Give me two other numbers,
this time one positive
and one negative**





Please, stop this
and just solve
it symbolically