

This is episode 2 of UPU.
Link to episode 1
in the description

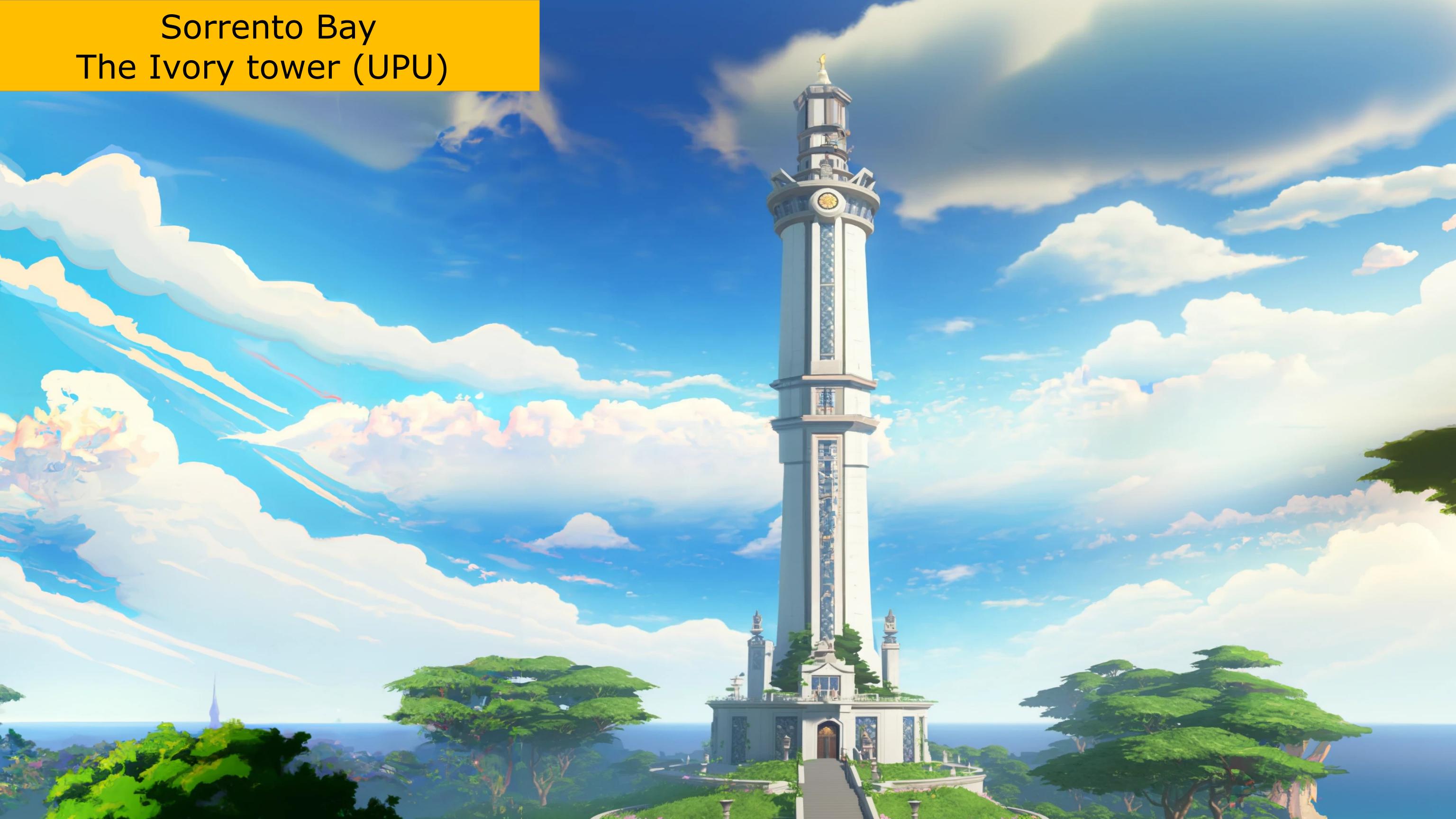
Episode 3 will be available
on 13 Feb 2023





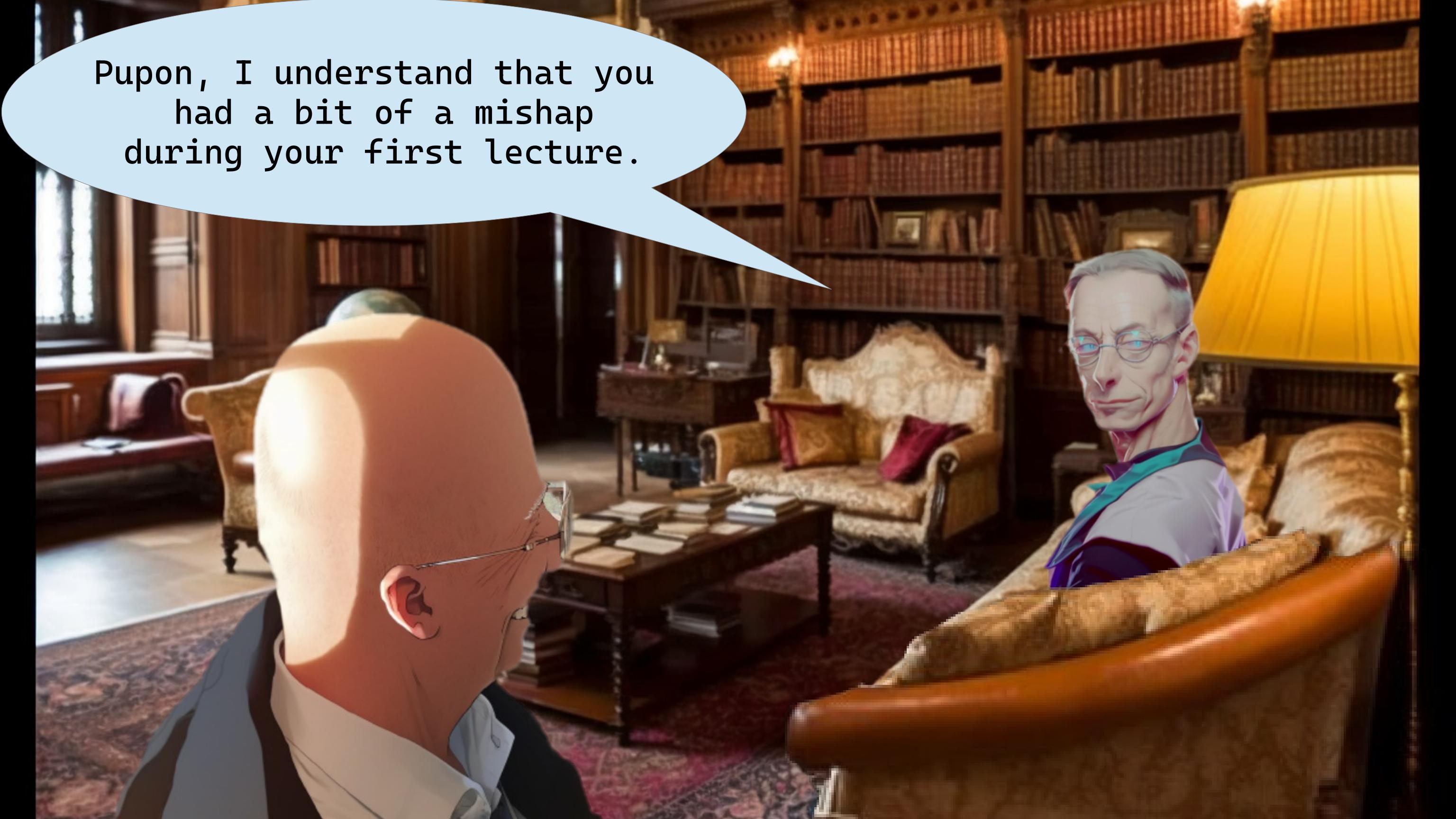
Sorrento Bay

The Ivory tower (UPU)



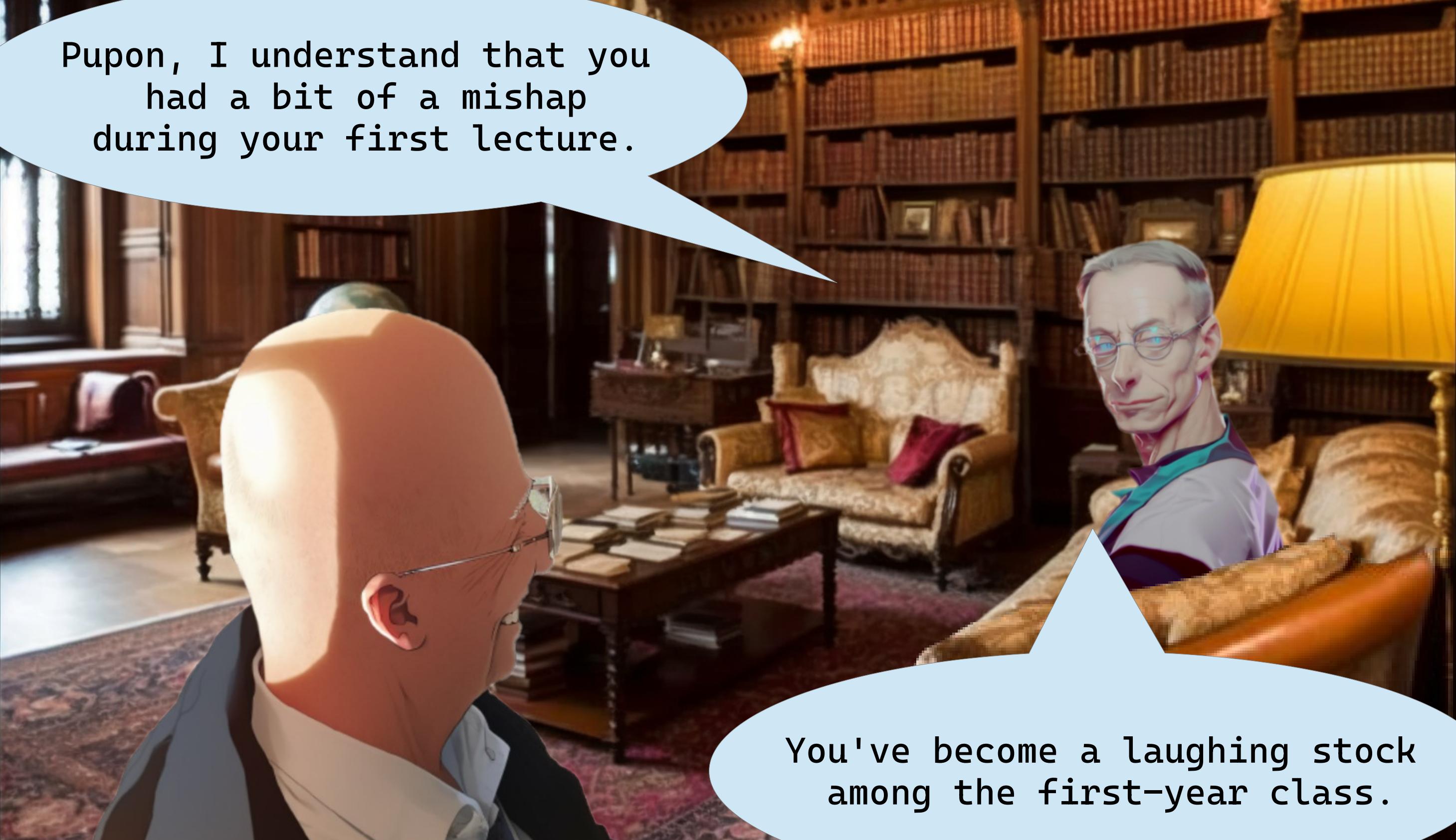
At the top of the Ivory tower,
headmaster Professor Eugene Mortis faces
Professor Elricho Pupon, his expression grave





Pupon, I understand that you
had a bit of a mishap
during your first lecture.





Pupon, I understand that you
had a bit of a mishap
during your first lecture.

You've become a laughing stock
among the first-year class.

Pupon hangs his head in shame,
his shoulders slumping



Pupon hangs his head in shame,
his shoulders slumping

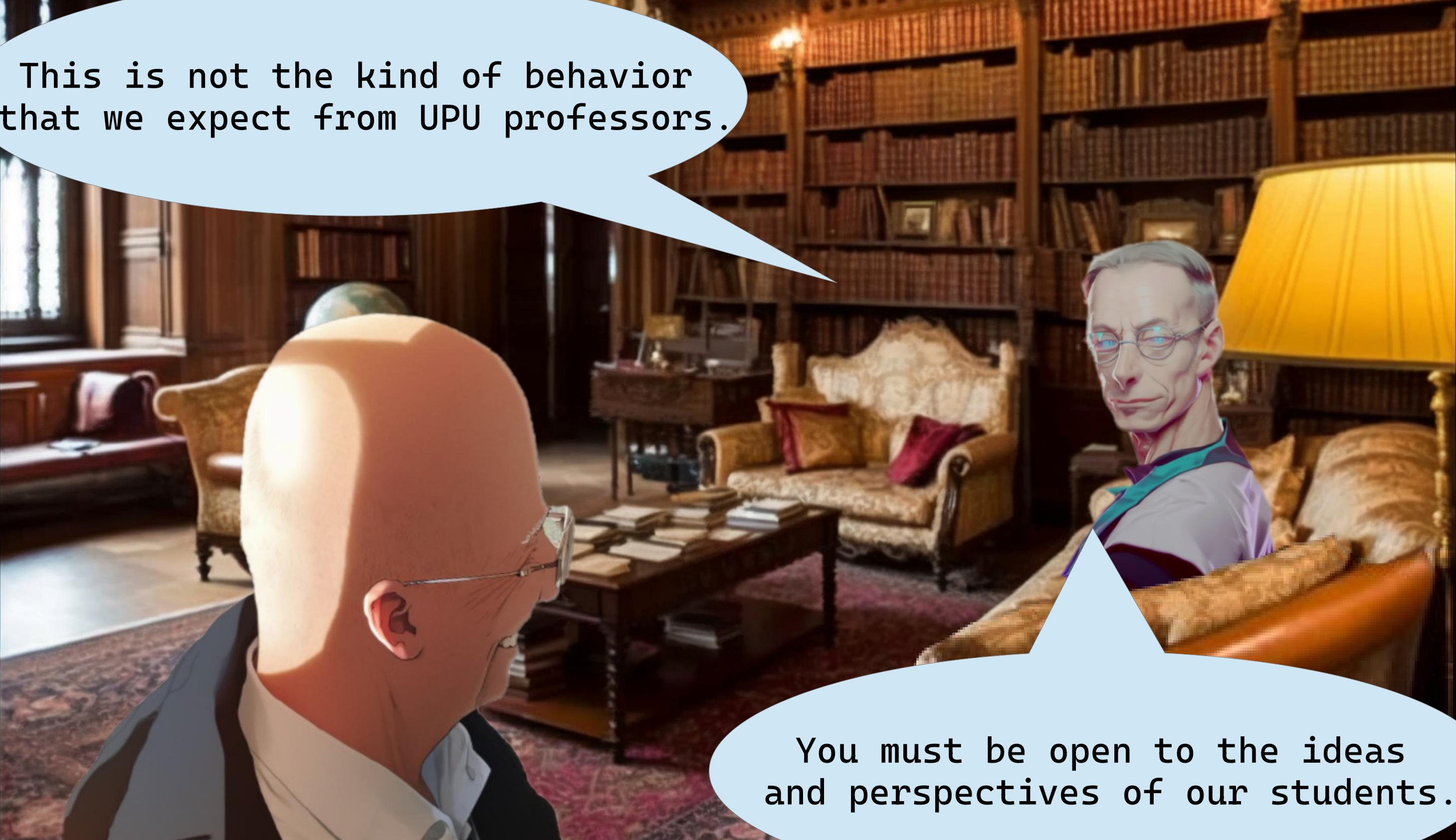


**That's correct, Professor Mortis.
I was too confident in my own
assumptions and didn't
consider the possibility of an error.**



This is not the kind of behavior
that we expect from UPU professors.



A photograph of a man in a suit and glasses looking towards another man who is partially visible on the right. They are in a library setting with bookshelves in the background.

This is not the kind of behavior
that we expect from UPU professors.

You must be open to the ideas
and perspectives of our students.



I, ... I apologize for bringing shame
upon the university.



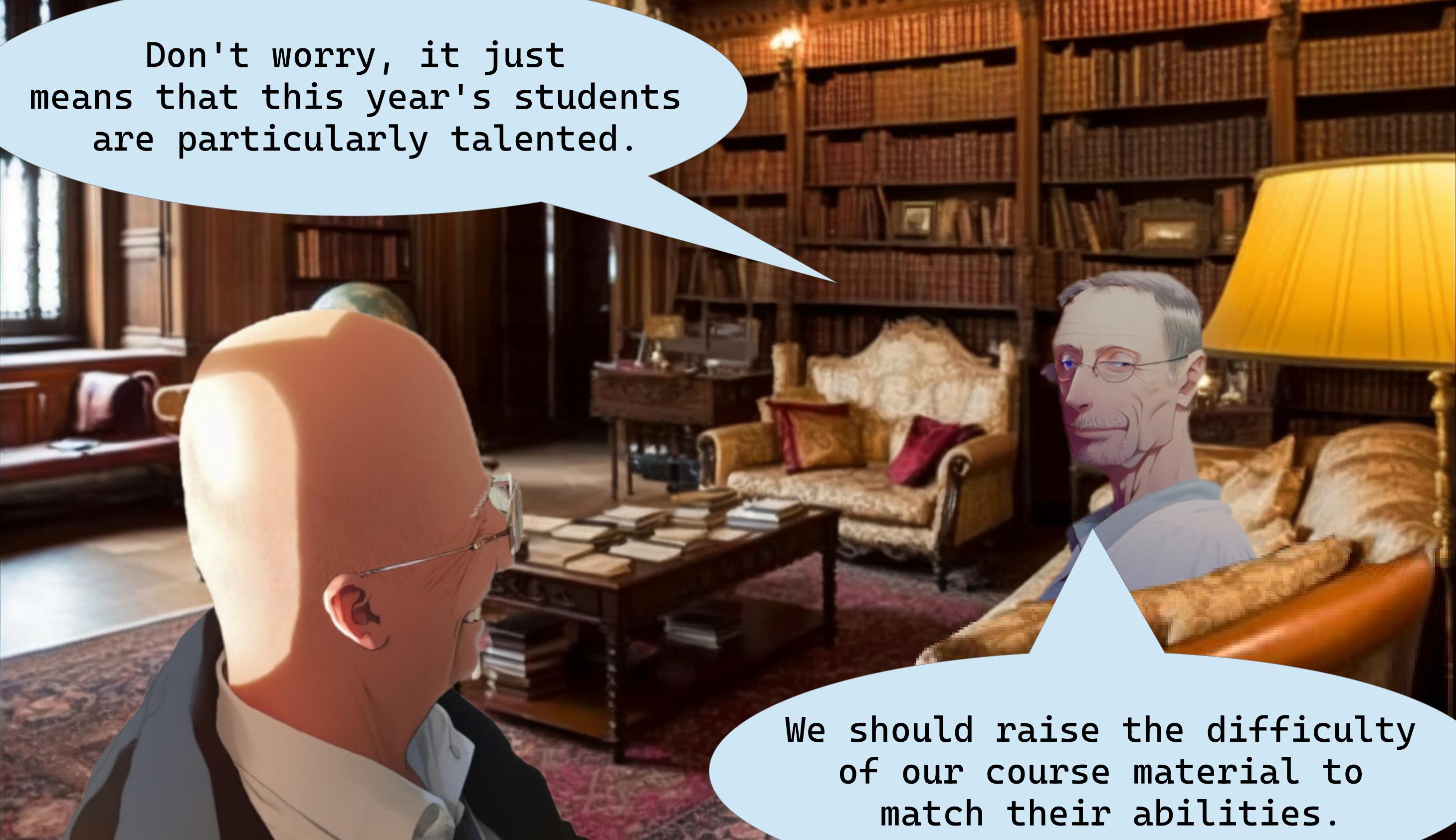
I, ... I apologize for bringing shame
upon the university.

I'll make sure to be more thorough
in my explanations in the future.



Don't worry, it just
means that this year's students
are particularly talented.



A photograph of a man with glasses and a mustache, wearing a light blue shirt, sitting at a desk in a library. He is looking towards another man who is partially visible on the right. The background features tall wooden bookshelves filled with books and a large yellow lamp on the right side.

Don't worry, it just
means that this year's students
are particularly talented.

We should raise the difficulty
of our course material to
match their abilities.



What are you proposing?

The 'equals switcheroo'.





The 'equals switcheroo'.

It will test their skills and understanding of programming.



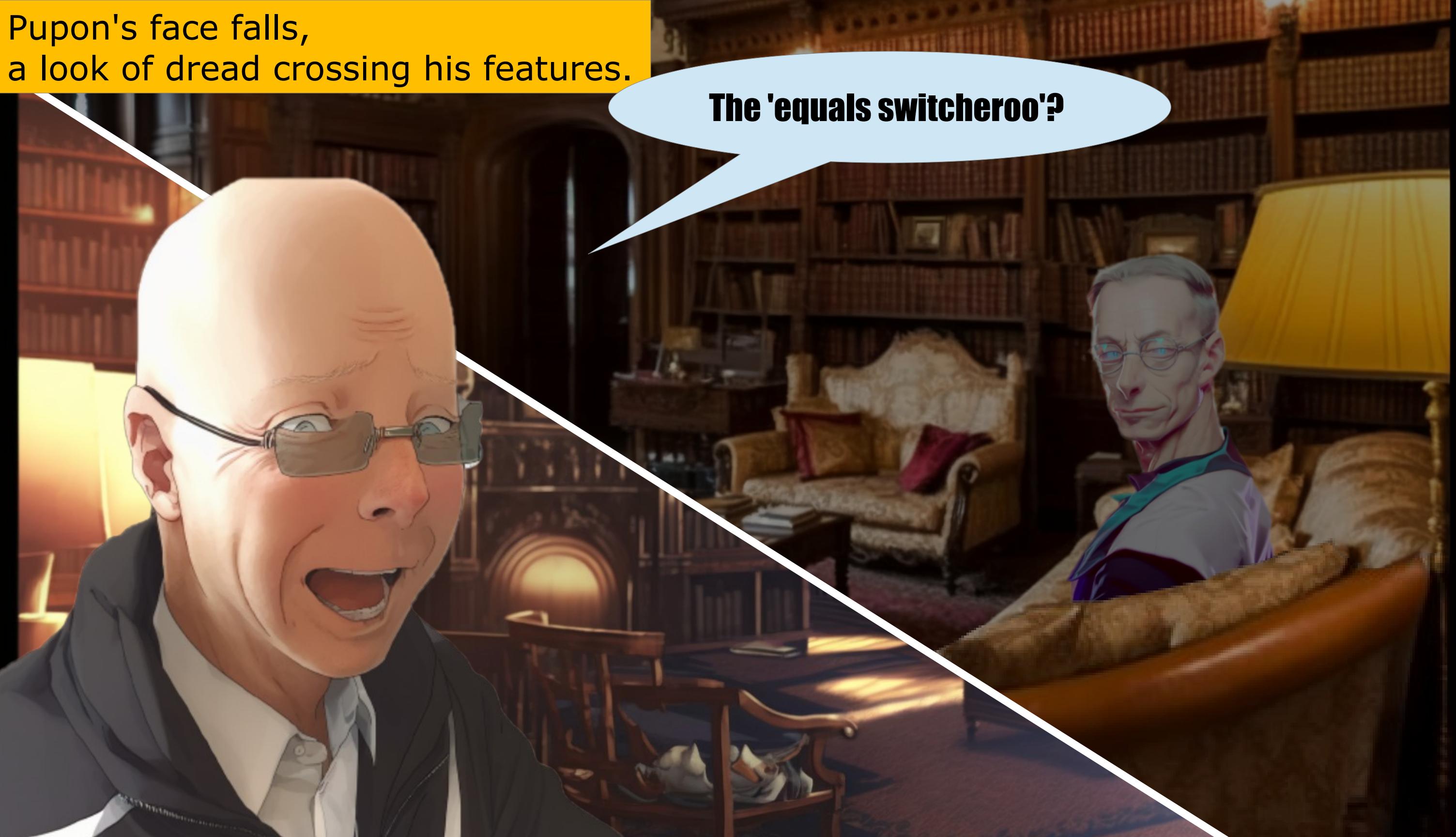
The 'equals switcheroo'.

It will test their skills and understanding of programming.

By raising the difficulty of our lessons, we can further challenge and engage our students.

Pupon's face falls,
a look of dread crossing his features.

The 'equals switcheroo'?



Pupon's face falls,
a look of dread crossing his features.

The 'equals switcheroo'?

Oh no, I remember that lesson.

Pupon's face falls,
a look of dread crossing his features.

The 'equals switcheroo'?

Oh no, I remember that lesson.

It's the most difficult and frustrating
challenge we give to students.

Mortis nods, his expression serious.



Mortis nods, his expression serious.



Exactly. It's reserved for
the most gifted students only.

Mortis nods, his expression serious.



Exactly. It's reserved for
the most gifted students only.

I have a feeling that this year's
class will be up to the task.

Pupon takes a deep breath, steeling himself.



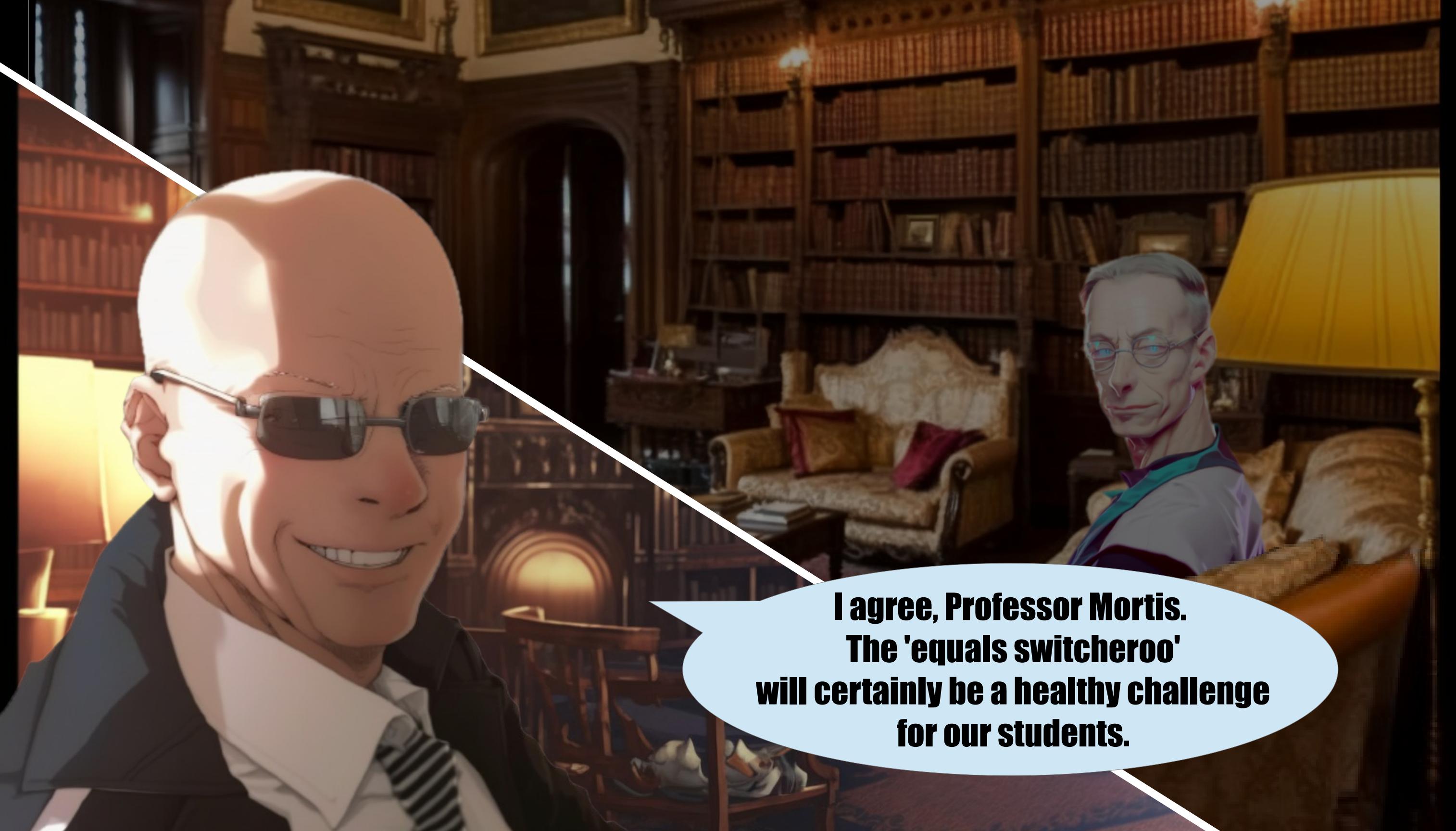
Pupon takes a deep breath, steeling himself.



I hope they are ready for the challenge.
I certainly wasn't when I took it.



**I will have my revenge
against that student.**



**I agree, Professor Mortis.
The 'equals switcheroo'
will certainly be a healthy challenge
for our students.**

A photograph of a man with white hair and glasses, wearing a purple robe, sitting at a desk in a richly wood-paneled library. He is looking towards a large globe on a chair. A large blue speech bubble originates from his mouth, containing the following text.

Excellent.
Now let's get back to work.
I've important research
to conclude today.



Excellent.

Now let's get back to work.
I've important research
to conclude today.

Yes, sir!

Meanwhile, at freshman campus.



Meanwhile, at freshman campus.

Why Dany, why did you do that?



Meanwhile, at freshman campus.



Why Dany, why did you do that?

Why did you have to
correct Professor Pupon in
front of the whole class?



I know, right? I'm so scared



I know, right? I'm so scared

Now Pupon is going
to be mad at me.

Relax, we're at UPU.





Relax, we're at UPU.

This is a prestigious university



Relax, we're at UPU.

This is a prestigious university

Professor Pupon will
have to act professionally



Relax, we're at UPU.

This is a prestigious university

Professor Pupon will
have to act professionally

He won't hold it
against you, right?

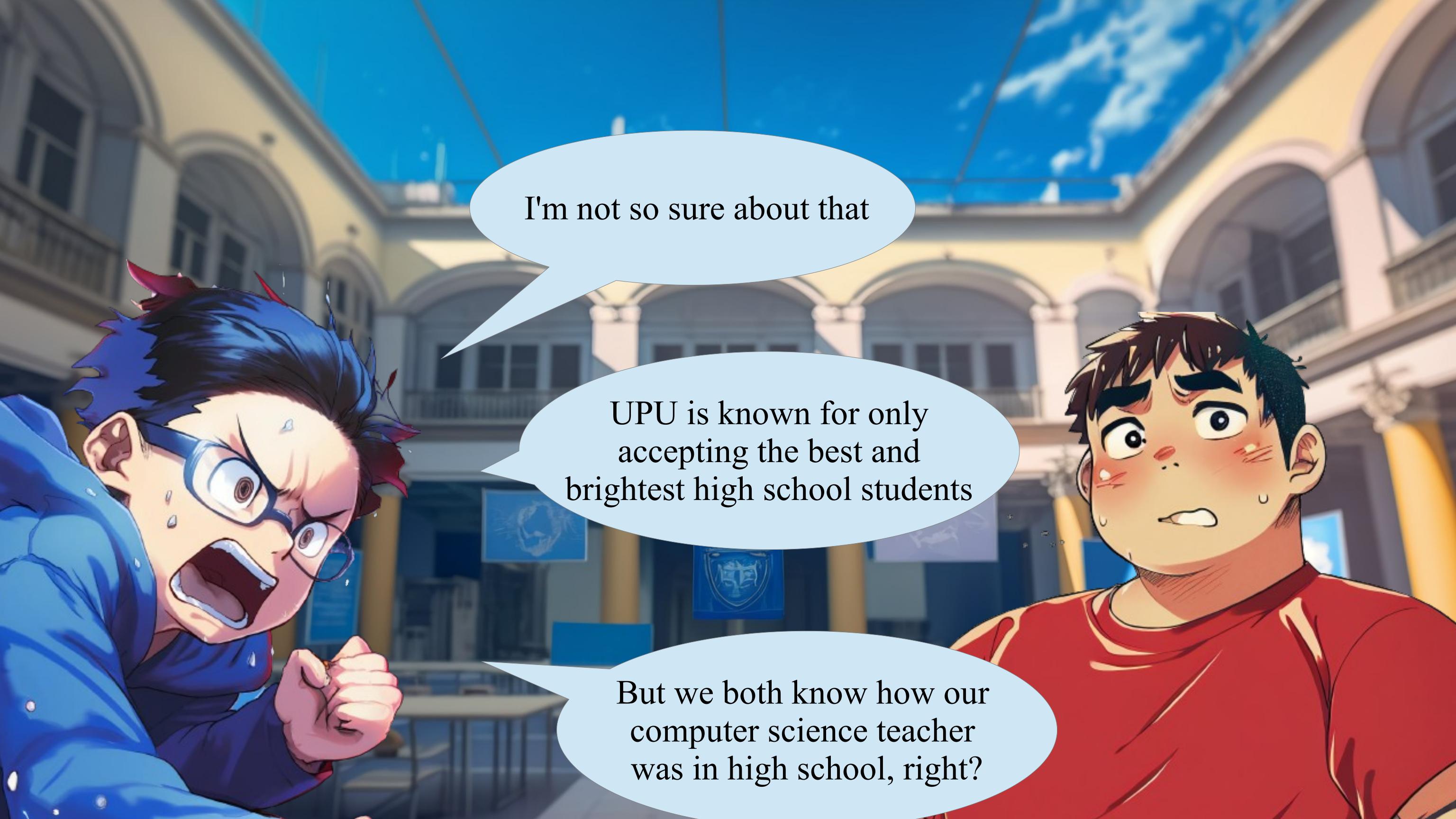


I'm not so sure about that



I'm not so sure about that

UPU is known for only
accepting the best and
brightest high school students

A cartoon illustration of two boys having a conversation outdoors. On the left, a boy with spiky blue hair and glasses is shouting with a determined expression, wearing a blue hoodie. On the right, a boy with dark hair and a red t-shirt looks surprised or confused. They are standing in front of a large, yellow and white building with arched windows and doors, likely a school. The sky is clear and blue.

I'm not so sure about that

UPU is known for only
accepting the best and
brightest high school students

But we both know how our
computer science teacher
was in high school, right?



...And you get an A+!

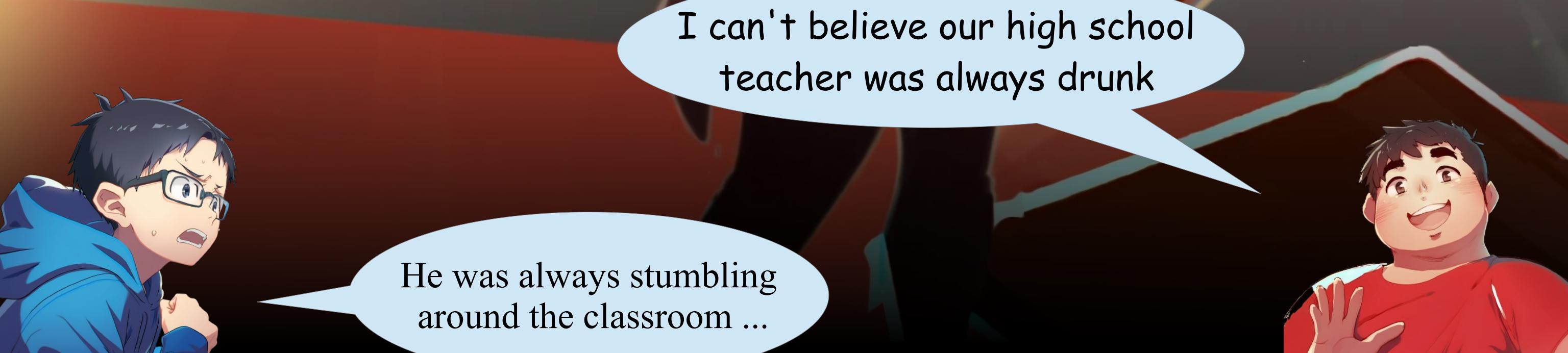


...And you get an A+!

I can't believe our high school
teacher was always drunk



..And you get an A+!



I can't believe our high school
teacher was always drunk

He was always stumbling
around the classroom ...



..And you get an A+ too!
And you!



Always giving everyone A+ grades



..And you get an A+ too!
And you!





And you! And you!





A teacher with glasses and a red jacket is shouting and handing out papers in a classroom. A student in the foreground is looking up at him. The teacher's speech bubble contains the text "And you! And you!" and "Everyone gets an A+!"

And you! And you!

Everyone gets an A+!

A teacher with glasses and a red tie is shouting with his mouth wide open, hands raised, and arms outstretched. In the foreground, a student with blue hair and glasses looks shocked. Several other students are running away from the teacher. A speech bubble above the teacher contains the text.

And you! And you!

Everyone gets an A+!

*without actually
teaching us anything.*

Yeah, it was crazy ...





Yeah, it was crazy ...

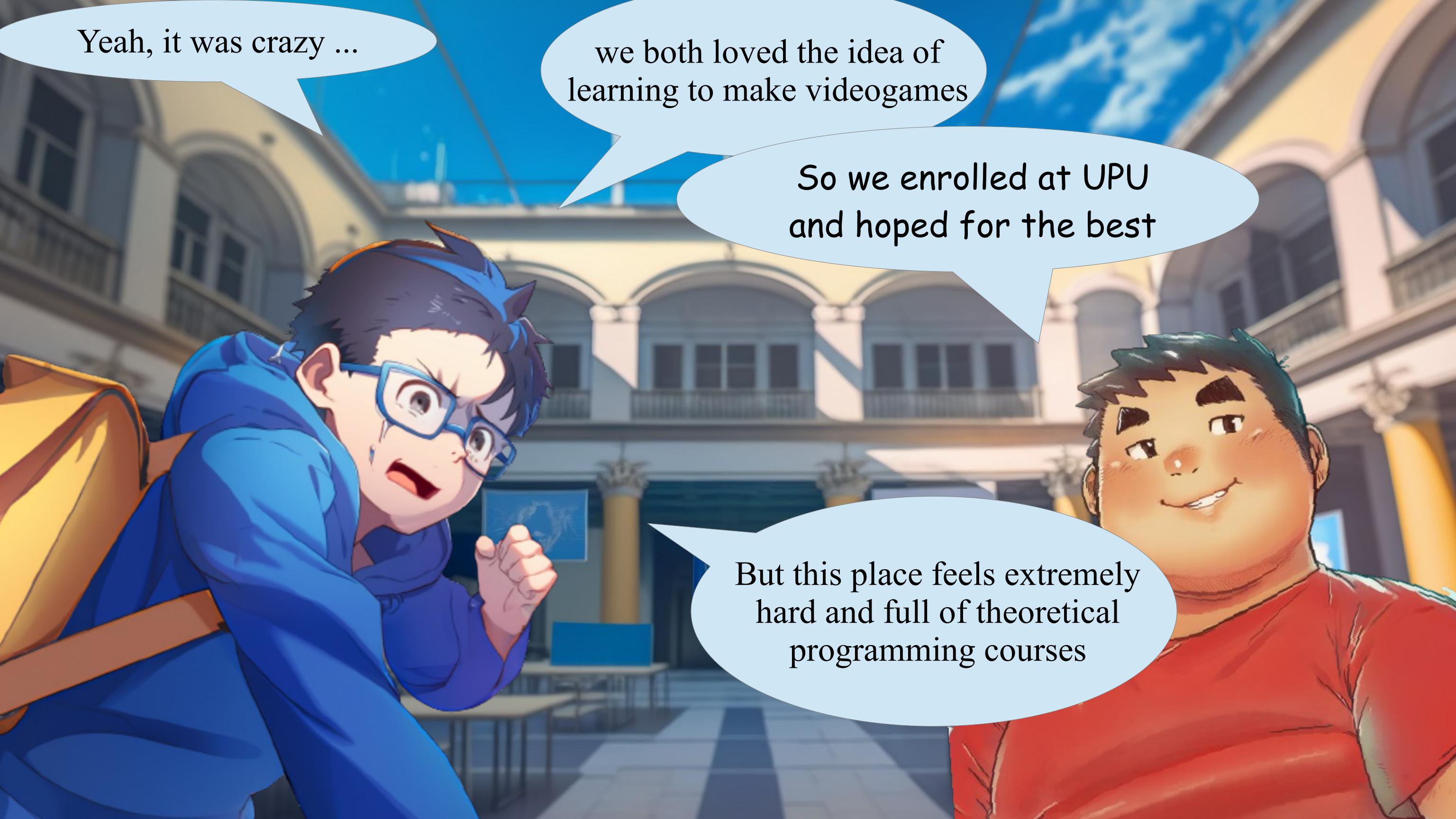
we both loved the idea of
learning to make videogames



Yeah, it was crazy ...

we both loved the idea of
learning to make videogames

So we enrolled at UPU
and hoped for the best



Yeah, it was crazy ...

we both loved the idea of learning to make videogames

So we enrolled at UPU and hoped for the best

But this place feels extremely hard and full of theoretical programming courses

A cartoon illustration of two young men in front of a large, classical-style university building with yellow walls and arched windows. The man on the left, wearing a blue hoodie and glasses, is gesturing while speaking. The man on the right, wearing a red t-shirt, is listening attentively. Four speech bubbles are visible, indicating their conversation.

Yeah, it was crazy ...

we both loved the idea of
learning to make videogames

So we enrolled at UPU
and hoped for the best

But this place feels extremely
hard and full of theoretical
programming courses

We were probably a little naive.



Anyway, finding that solution
in Professor Pupon's lecture...



Anyway, finding that solution
in Professor Pupon's lecture...

it was impressive.



Anyway, finding that solution
in Professor Pupon's lecture...

it was impressive.

How did you figure it out?

I have no idea. It was like...



I have no idea. It was like...

like I was possessed or something





I have no idea. It was like...

like I was possessed or something

I just saw the solution in
my mind and acted on it



I have no idea. It was like...

like I was possessed or something

I just saw the solution in
my mind and acted on it

I just couldn't help it.
It was like a vision or something



I saw the solution in my mind



I saw the solution in my mind

... and I had to act on it,
even though I was terrified.

The stage is set for a new lecture to commence



Conditionals:

The stage is set for a new lecture to commence





**We can use the if statement
to take decisions**

```
int input = 5;  
int res = 0;  
if( ) {  
    assert res ==  
}
```

A cartoon illustration of Steve Jobs, showing his head and shoulders. He has a bald head, wears dark sunglasses, and has a slight smile. He is wearing a light-colored collared shirt under a dark jacket.

**We can use the if statement
to take decisions**

**In the first example,
input is 5,
the if condition holds**

```
int input = 5;  
int res = 0;  
if( input>3 ) {  
    assert res ==
```

A cartoon illustration of Steve Jobs, showing his head and shoulders. He has a shaved head, wears glasses, and has a slight smile.

**We can use the if statement
to take decisions**

**In the first example,
input is 5,
the if condition holds**

**Thus we execute the body
and we get 10**

```
int input = 5;  
int res = 0;  
if( input>3 ){ res = 10; }  
assert res == 10;
```



**We can use the if statement
to take decisions**

**In the first example,
input is 5,
the if condition holds**

**Thus we execute the body
and we get 10**

```
int input = 5;  
int res = 0;  
if( input>3 ){ res = 10; }  
assert res == 10;
```

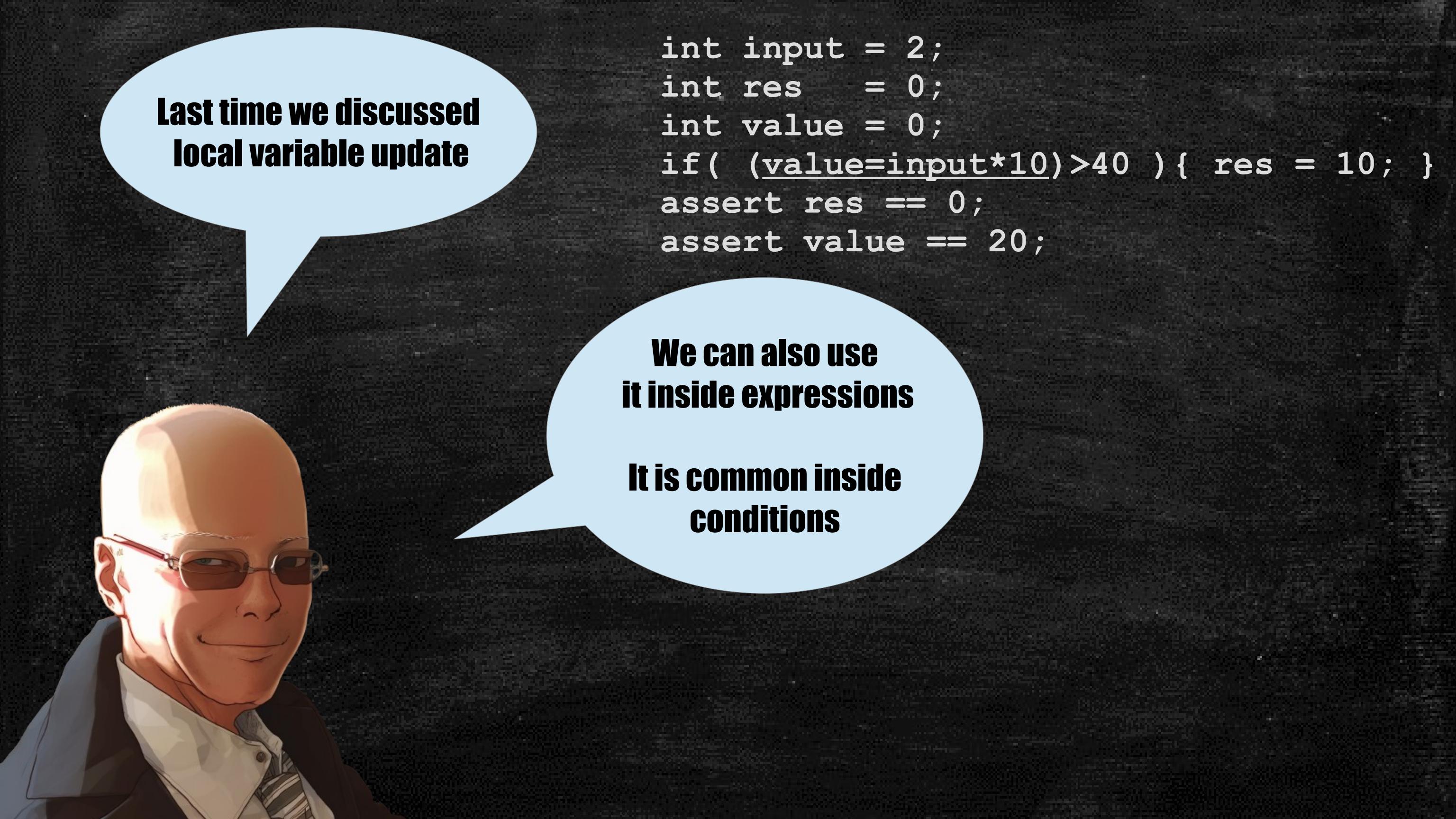
```
int input = 2;  
int res = 0;  
if( input>3 ){ res = 10; }  
assert res == 0;
```

**In the second example,
input is 2,
the if condition does not hold.
Thus we skip the body
and we get 0**

A cartoon illustration of a bald man wearing dark sunglasses and a white shirt with a striped collar. He is smiling and looking towards the right. A light blue speech bubble originates from his mouth, containing the text "Last time we discussed local variable update".

**Last time we discussed
local variable update**

```
int input = 2;  
int res   = 0;  
int value = 0;  
if( (value=input*10)>40 ) { res = 10; }  
assert res == 0;  
assert value == 20;
```



**Last time we discussed
local variable update**

```
int input = 2;  
int res   = 0;  
int value = 0;  
if( value=input*10)>40 ) { res = 10; }  
assert res == 0;  
assert value == 20;
```

**We can also use
it inside expressions**

**It is common inside
conditions**



Last time we discussed
local variable update

```
int input = 2;  
int res = 0;  
int value = 0;  
if( value=input*10)>40 ) { res = 10; }  
assert res == 0;  
assert value == 20;
```

A white arrow points from the word "value" in the first line of the code to the number "20" in the last line.

We can also use
it inside expressions

It is common inside
conditions

value equal input times ten
is an expression producing
the assigned value.
That is, 20.

```
int input = 2;  
int res = 0;  
int value = 0;  
if( (value=input*10)>40 ) { res = 10; }  
assert res == 0;  
assert value == 20;
```



Step by step



Input is 2

```
int input = 2;  
int res = 0;  
int value = 0;  
if( (value= 2 *10)>40 ){ res = 10; }  
assert res == 0;  
assert value == 20;
```



Input is 2

2 times 10 is 20

```
int input = 2;  
int res = 0;  
int value = 0;  
if( (value= 20 )>40 ){ res = 10; }  
assert res == 0;  
assert value == 20;
```



Input is 2

2 times 10 is 20

We update value to 20

```
int input = 2;  
int res = 0;  
int value = 20;  
if( (value= 20 )>40 ){ res = 10; }  
assert res == 0;  
assert value == 20;
```



Input is 2

2 times 10 is 20

We update value to 20

value = 20 is now just '20'

```
int input = 2;  
int res = 0;  
int value = 20;  
if( ( 20 )>40 ){ res = 10; }  
assert res == 0;  
assert value == 20;
```



Input is 2

2 times 10 is 20

We update value to 20

value = 20 is now just '20'

**20 is not bigger then 40
so we get false.**

```
int input = 2;  
int res = 0;  
int value = 20;  
if( false ) { res = 10; }  
assert res == 0;  
assert value == 20;
```



Input is 2

2 times 10 is 20

We update value to 20

value = 20 is now just '20'

**20 is not bigger then 40
so we get false.**

with false, we skip the if body

```
int input = 2;  
int res = 0;  
int value = 20;  
if( false  
assert res == 0;  
assert value == 20;
```

~~) { res = 10; }~~



As you can see ...



As you can see ...

**variable assignments in conditionals
allows us to save information about partial
computations in the condition itself**



So, you, . . . come here!



So, you, ... come here!

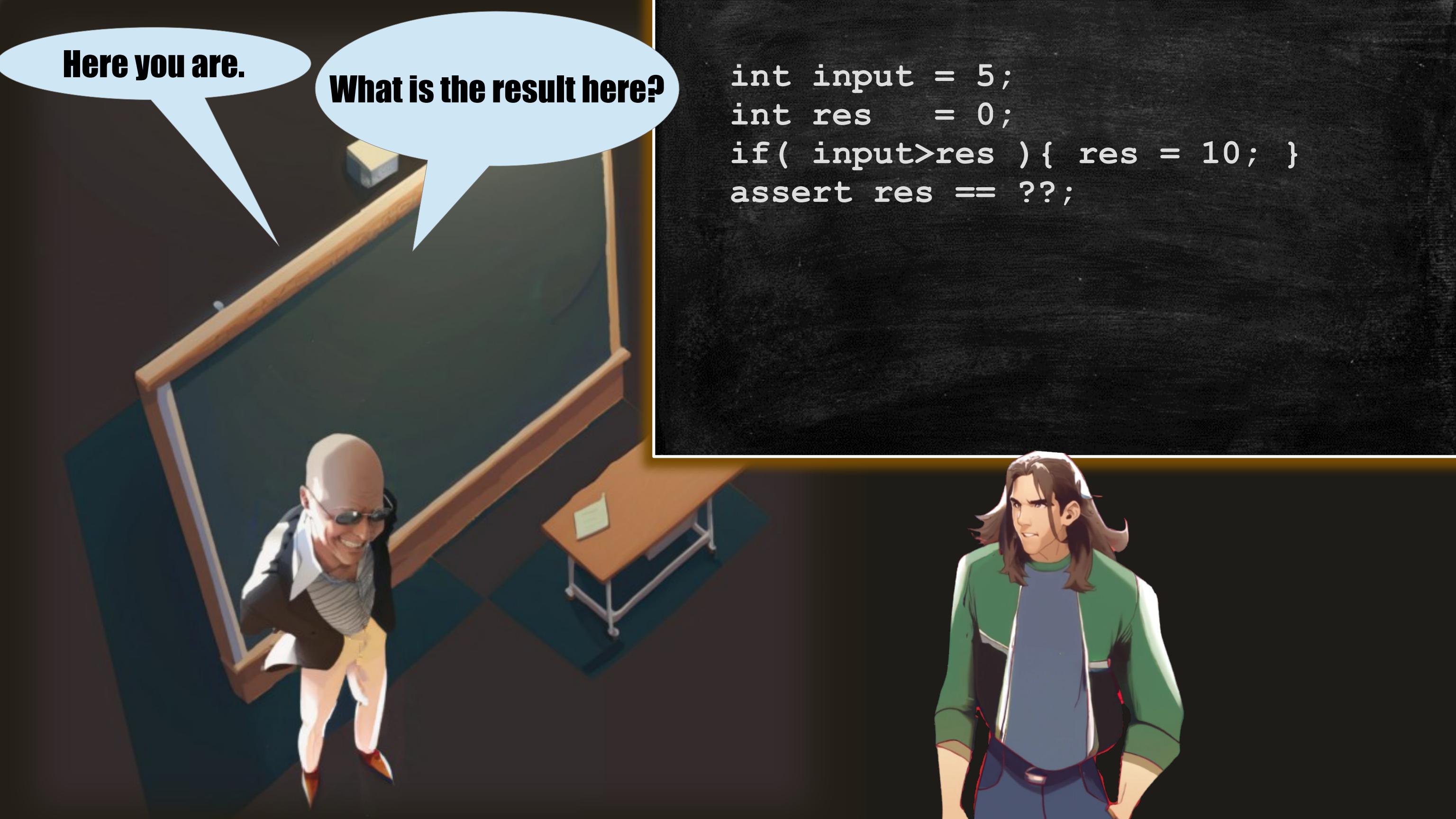


So, you, ... come here!

Do you mean... Me?!?

Here you are.

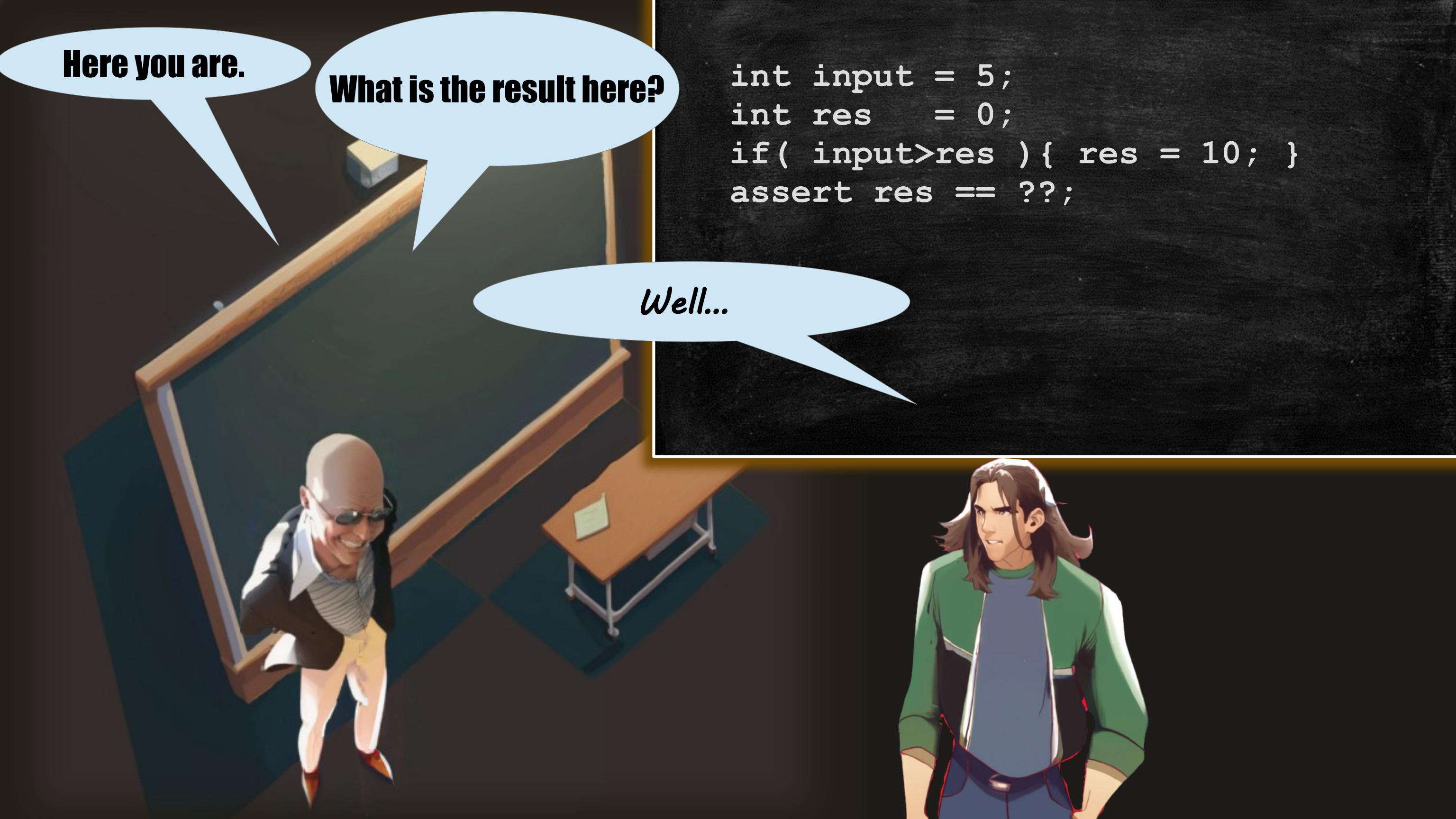




Here you are.

What is the result here?

```
int input = 5;  
int res   = 0;  
if( input>res ){ res = 10; }  
assert res == ??;
```

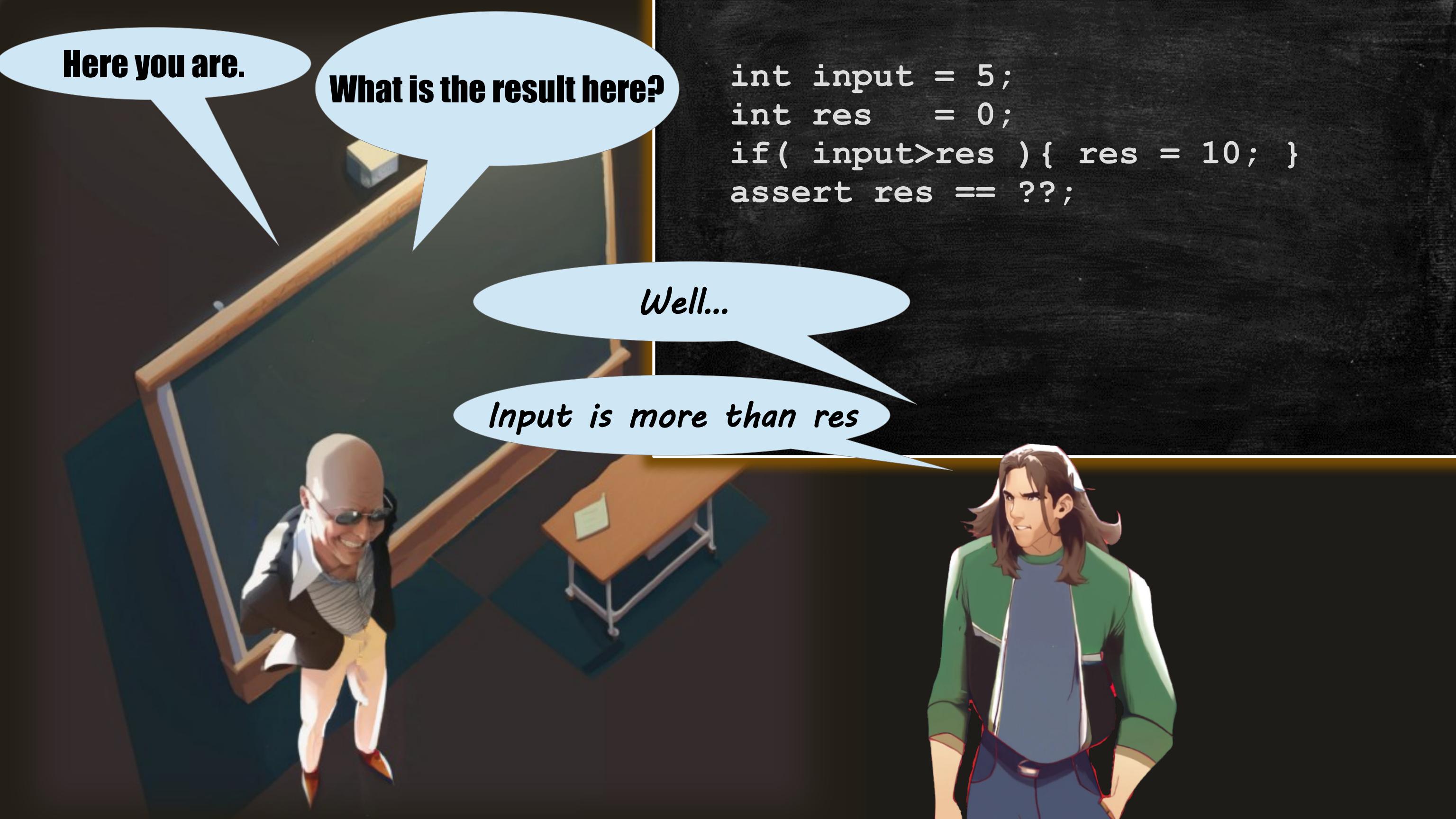


Here you are.

What is the result here?

```
int input = 5;  
int res   = 0;  
if( input>res ){ res = 10; }  
assert res == ??;
```

Well...



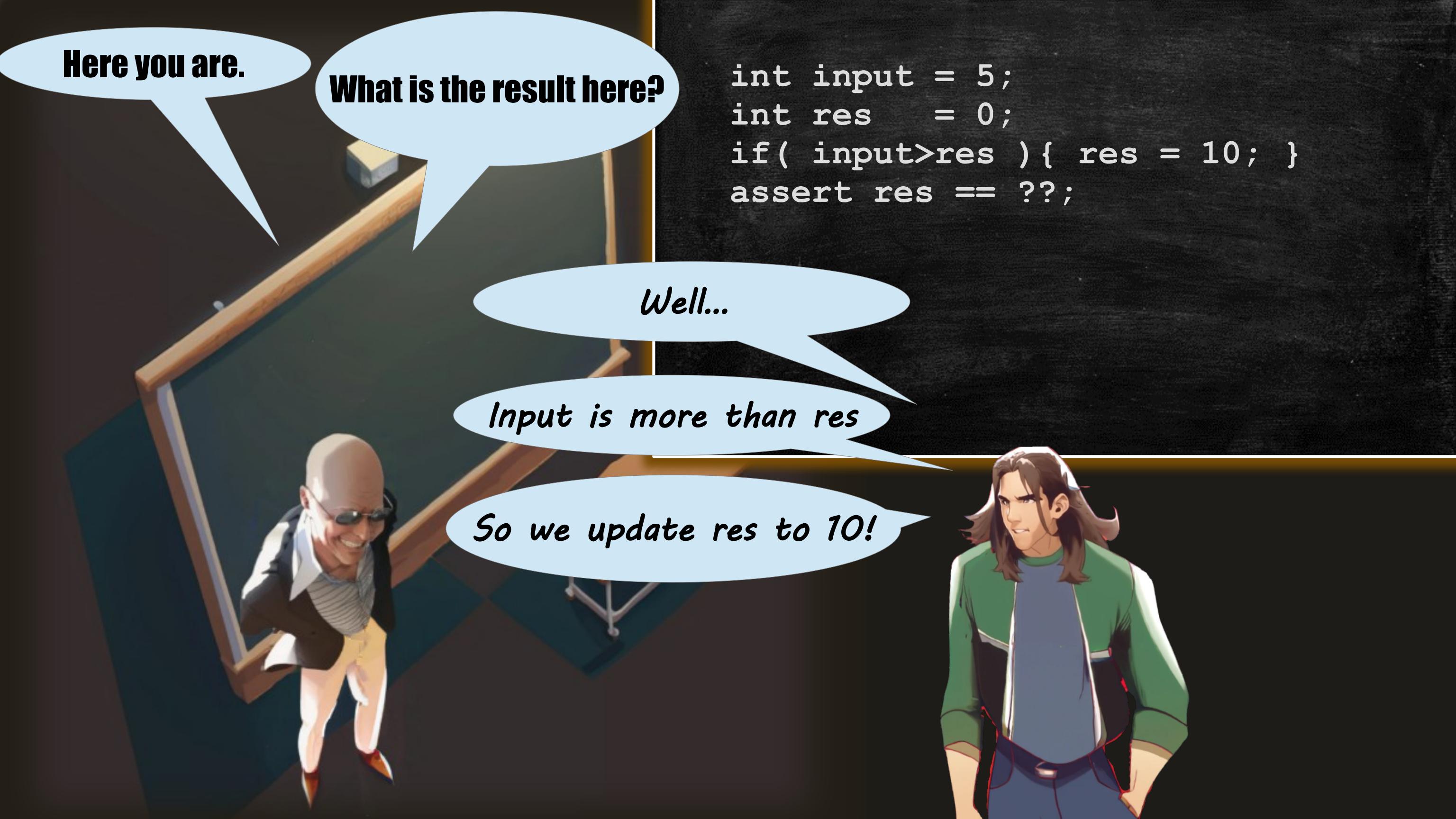
Here you are.

What is the result here?

```
int input = 5;  
int res   = 0;  
if( input>res ){ res = 10; }  
assert res == ??;
```

Well...

Input is more than res



Here you are.

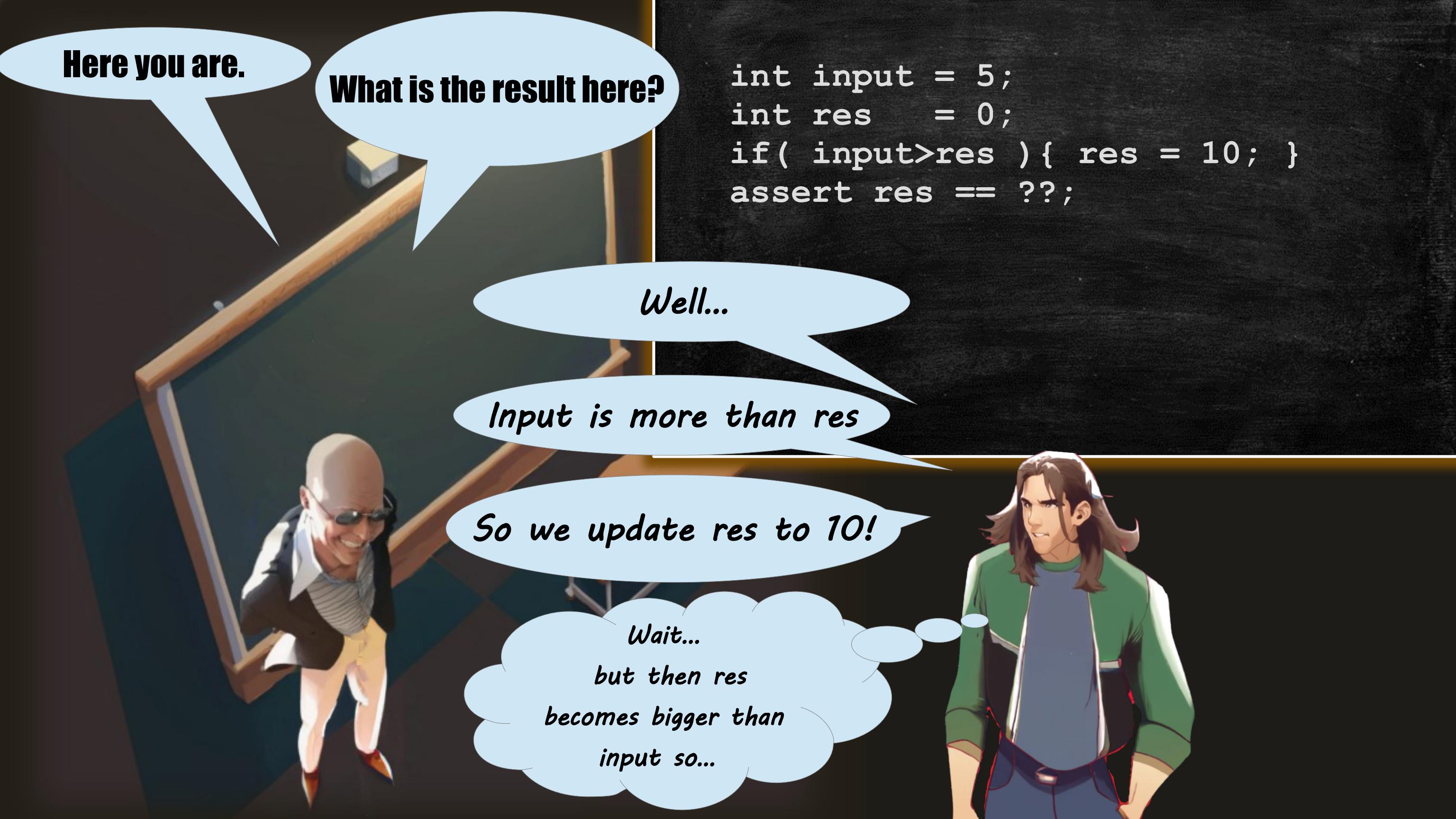
What is the result here?

```
int input = 5;  
int res   = 0;  
if( input>res ){ res = 10; }  
assert res == ??;
```

Well...

Input is more than res

So we update res to 10!



Here you are.

What is the result here?

```
int input = 5;  
int res   = 0;  
if( input>res ){ res = 10; }  
assert res == ??;
```

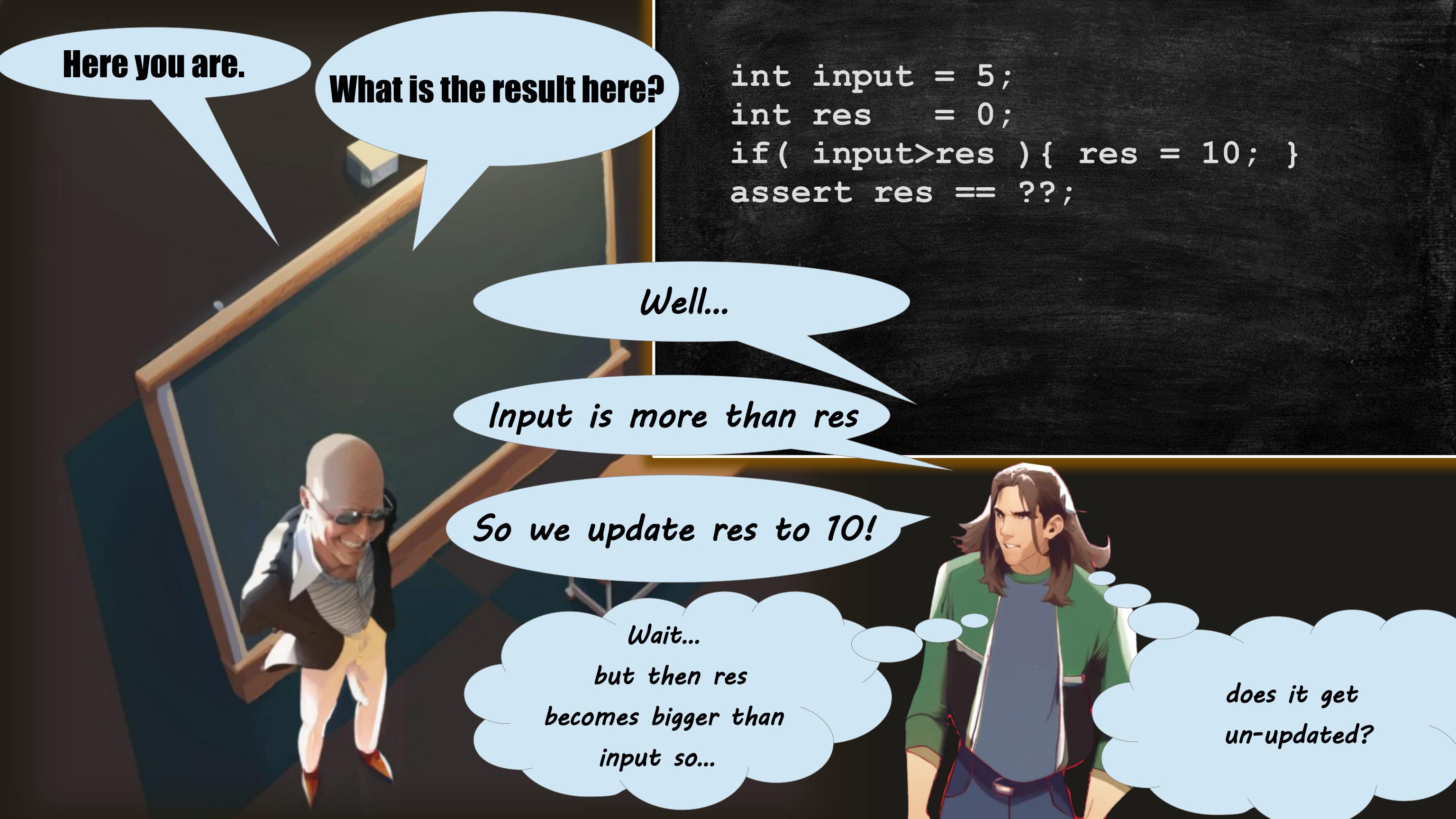
Well...

Input is more than res

So we update res to 10!

Wait...
but then res
becomes bigger than
input so...





Here you are.

What is the result here?

```
int input = 5;  
int res = 0;  
if( input>res ){ res = 10; }  
assert res == ???;
```

Well...

Input is more than res

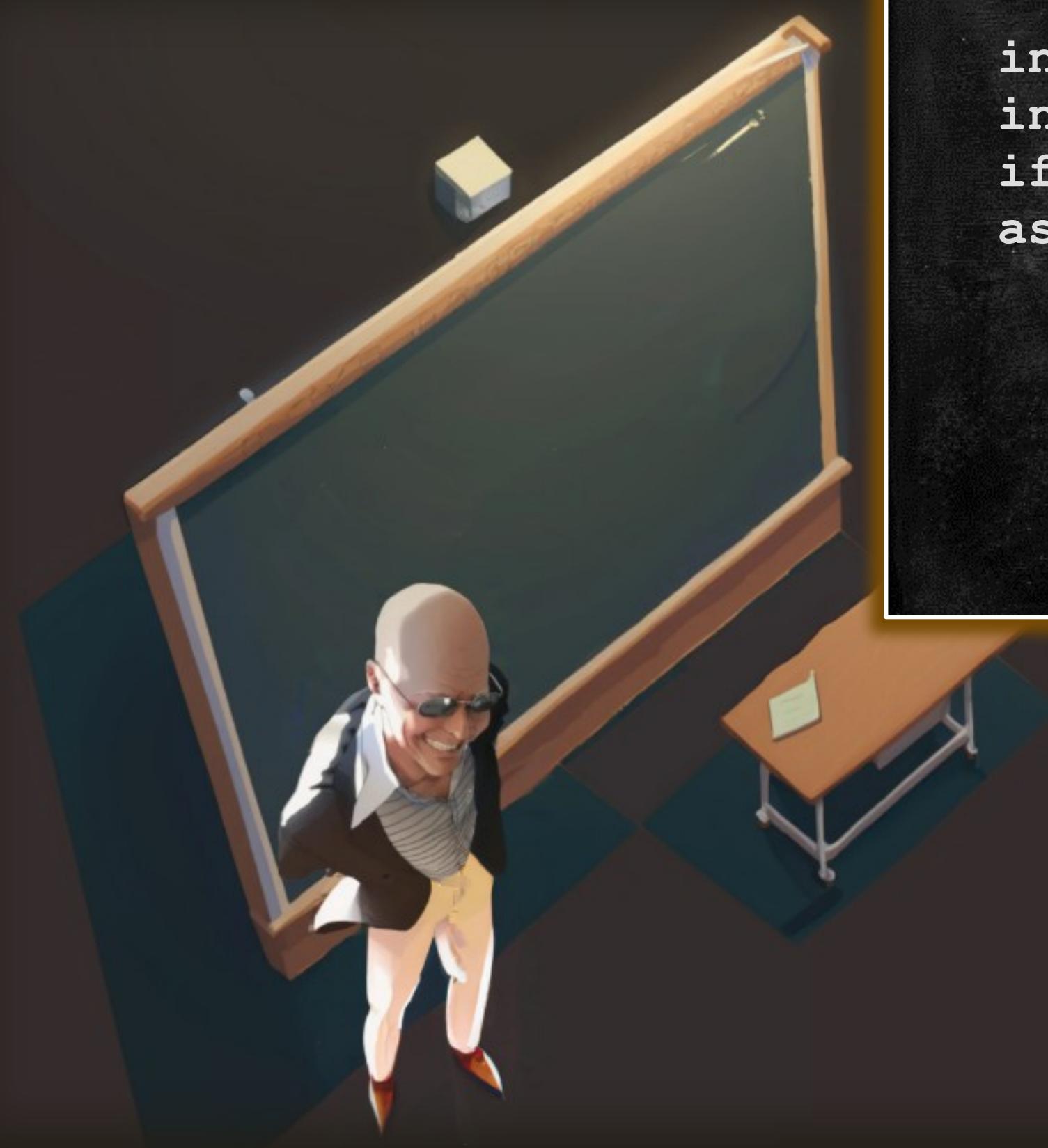
So we update res to 10!

Wait...

but then res
becomes bigger than
input so...

does it get
un-updated?

```
int input = 5;  
int res   = 0;  
if( input>res ){ res = 10; }  
assert res == ??;
```



Very good



```
int input = 5;  
int res   = 0;  
if( input>res ){ res = 10; }  
assert res == ??;
```



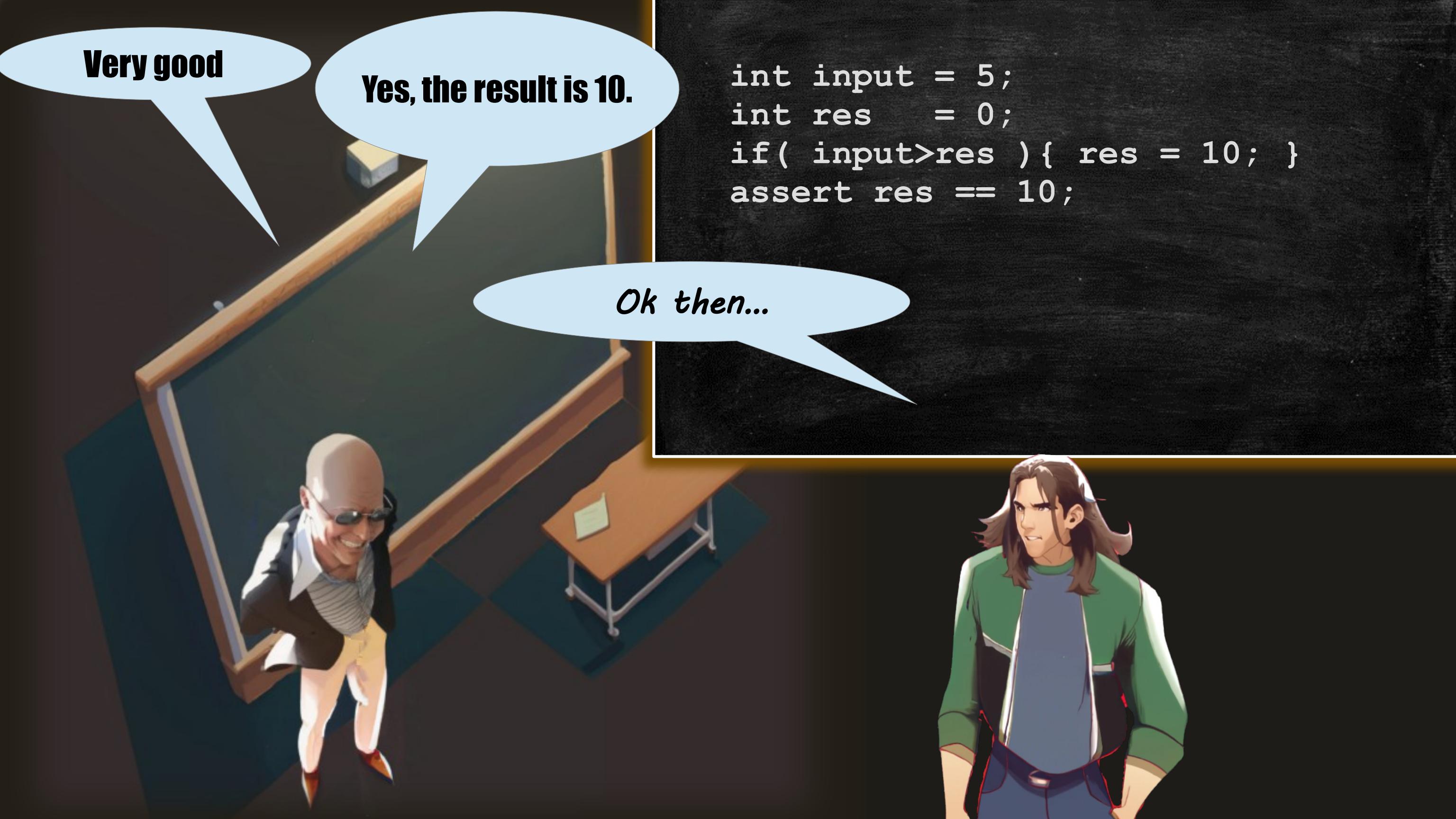


Very good

Yes, the result is 10.

```
int input = 5;  
int res   = 0;  
if( input>res ){ res = 10; }  
assert res == 10;
```



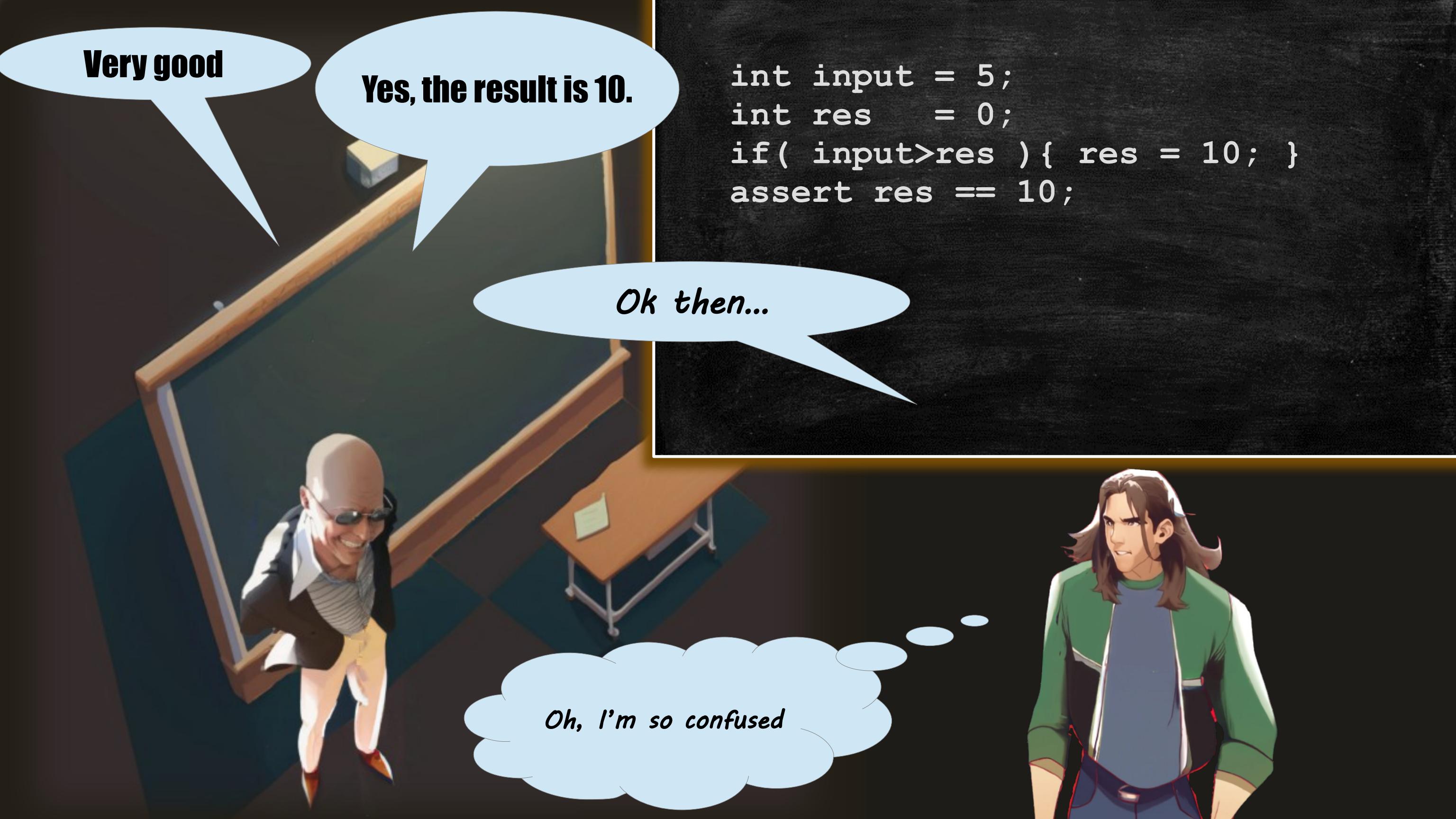


Very good

Yes, the result is 10.

```
int input = 5;  
int res   = 0;  
if( input>res ){ res = 10; }  
assert res == 10;
```

Ok then...



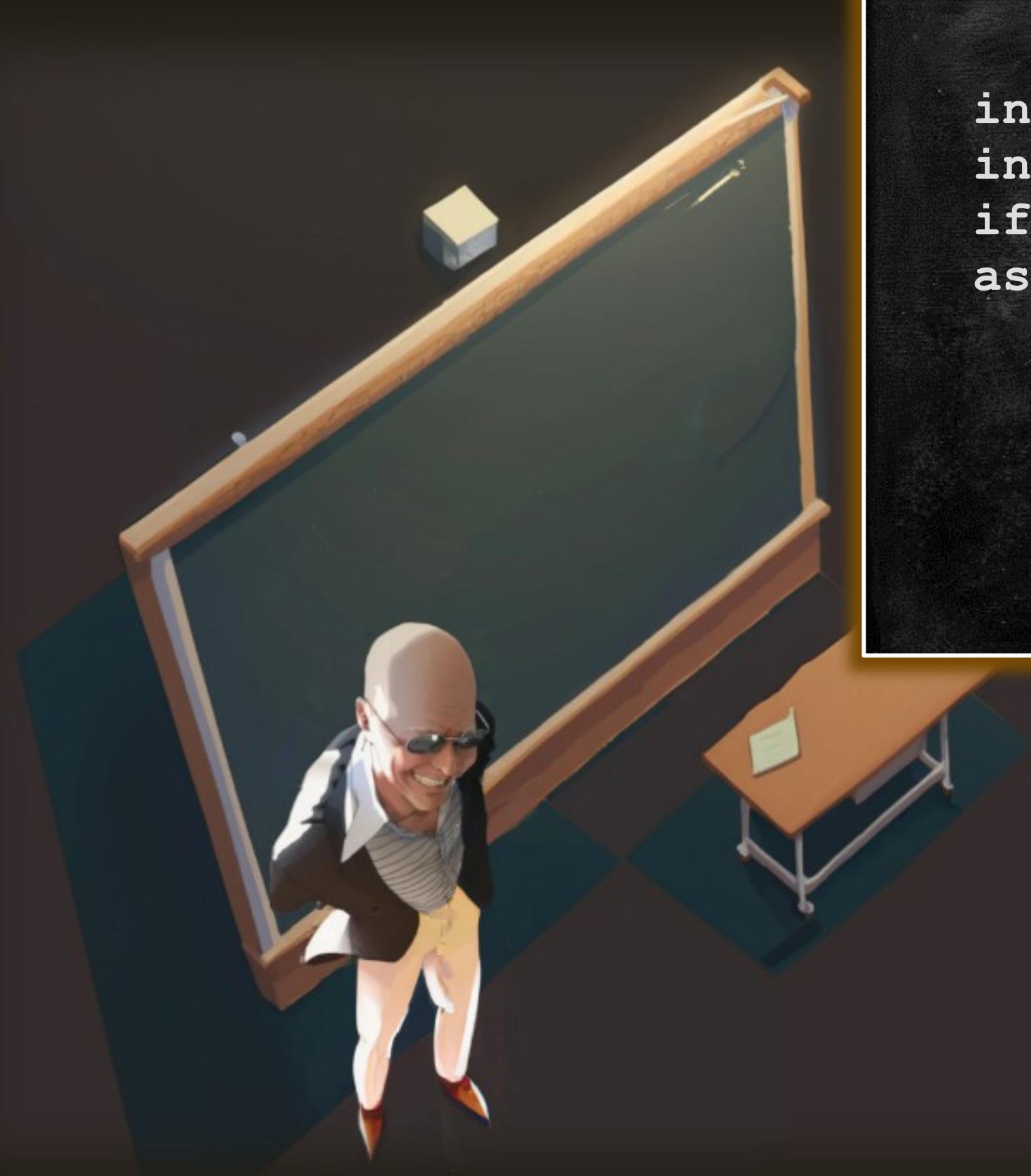
Very good

Yes, the result is 10.

```
int input = 5;  
int res   = 0;  
if( input>res ){ res = 10; }  
assert res == 10;
```

Ok then...

Oh, I'm so confused



```
int input = 5;  
int res   = 0;  
if( input>res ){ res = 10; }  
assert res == 10;
```

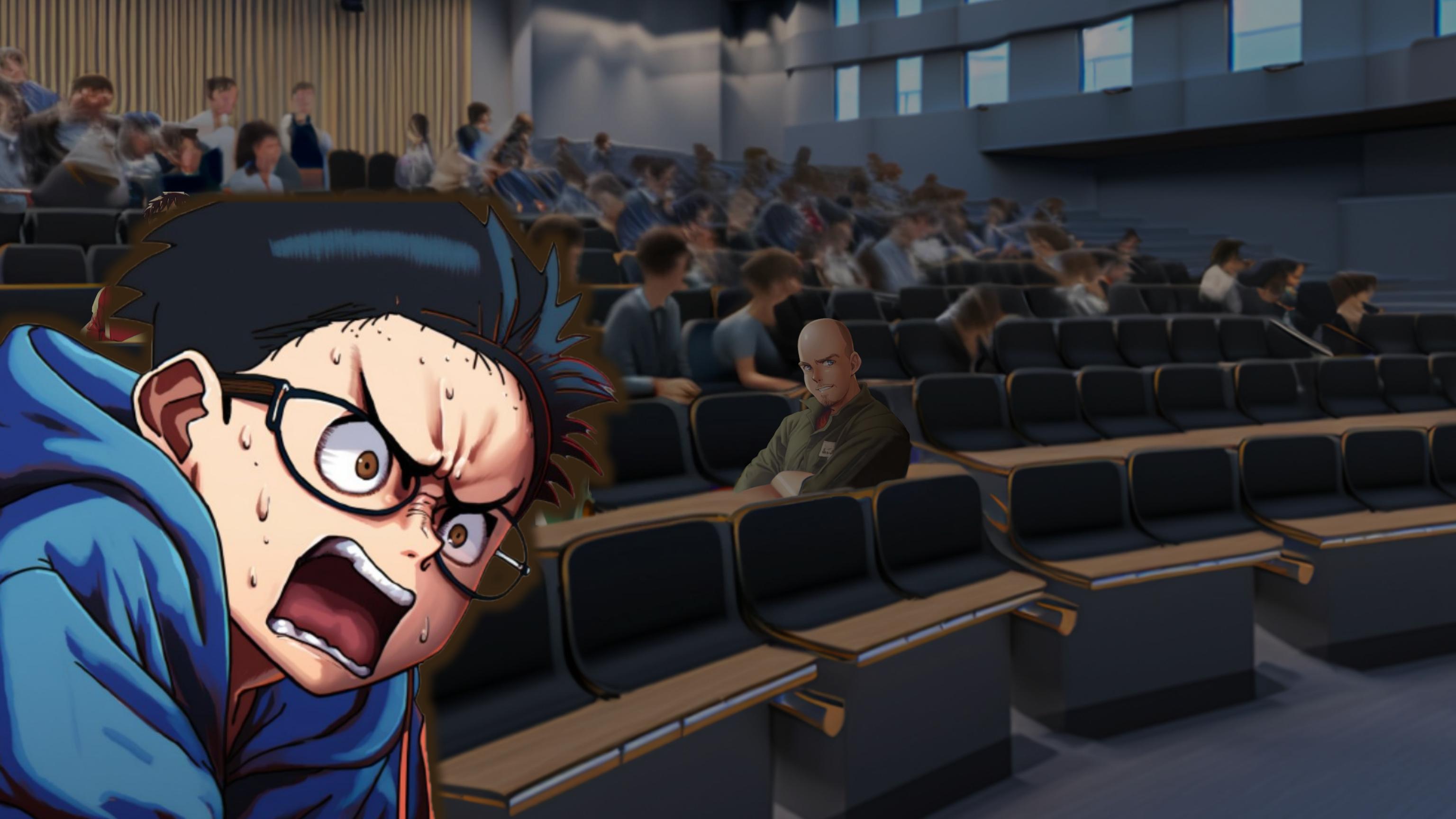


Next, you above!



Me?!?

Next, you above!

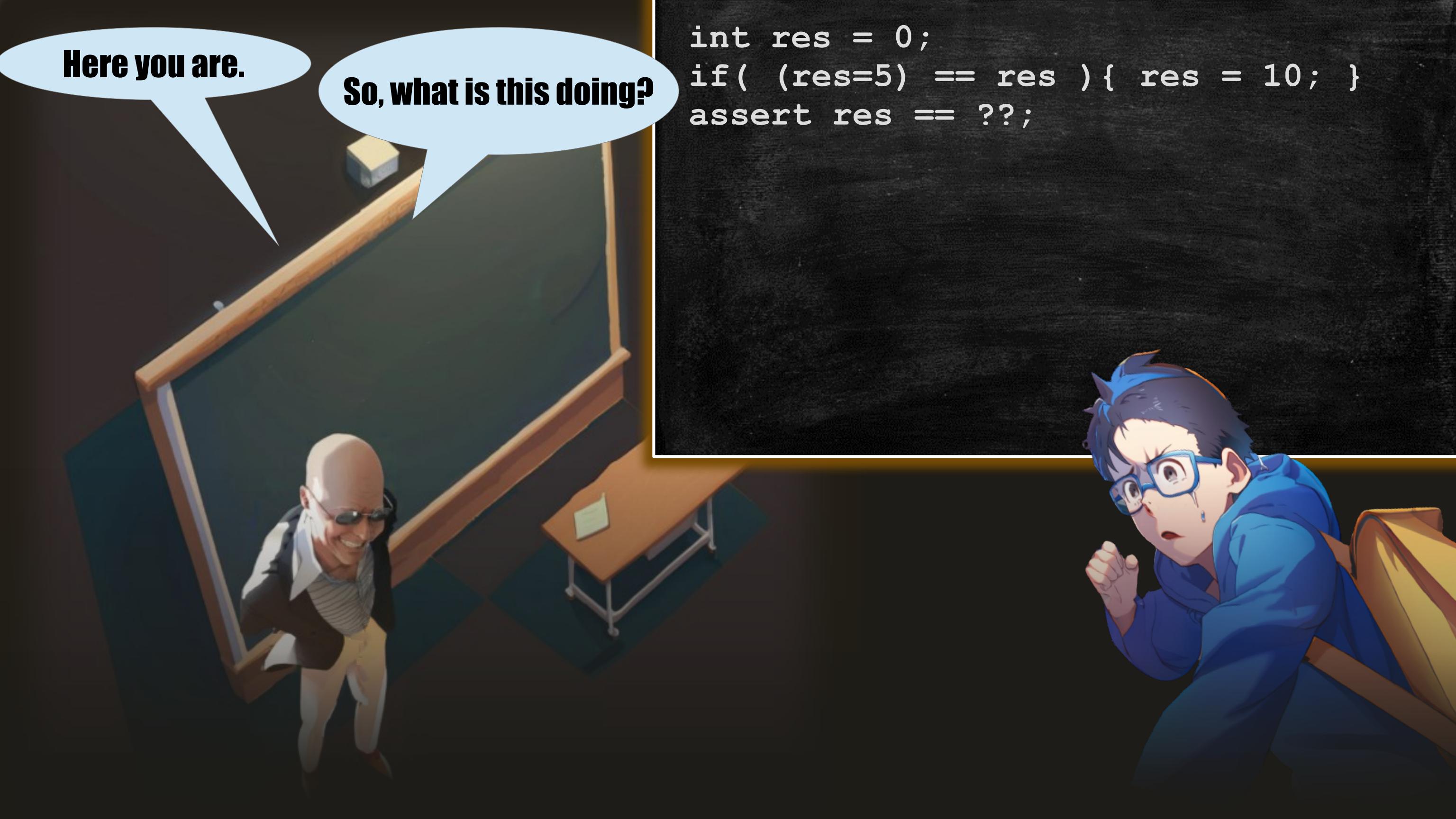




This is it,
he is going to take revenge!

Here you are.

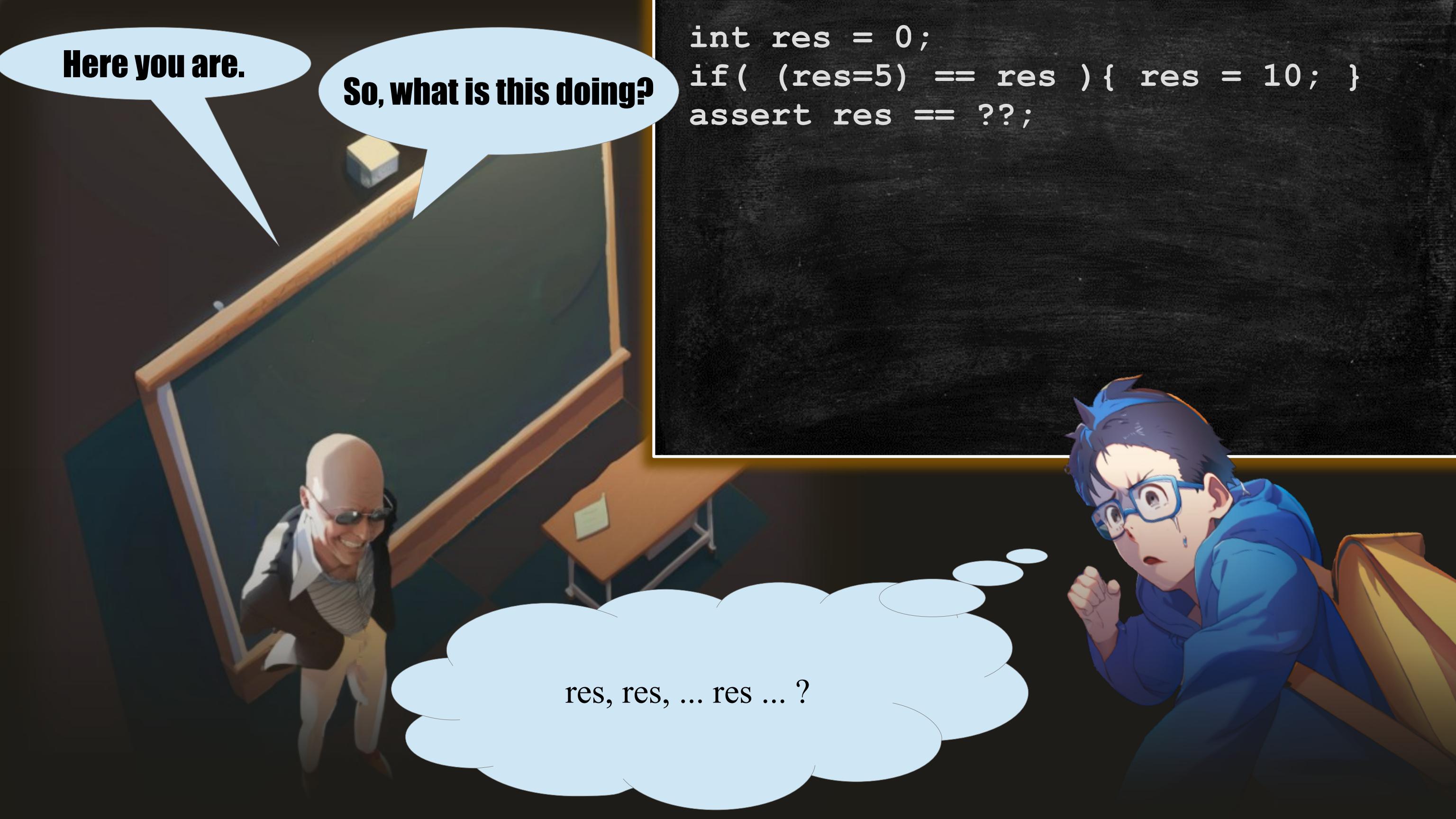




Here you are.

So, what is this doing?

```
int res = 0;  
if( (res=5) == res ){ res = 10; }  
assert res == ??;
```

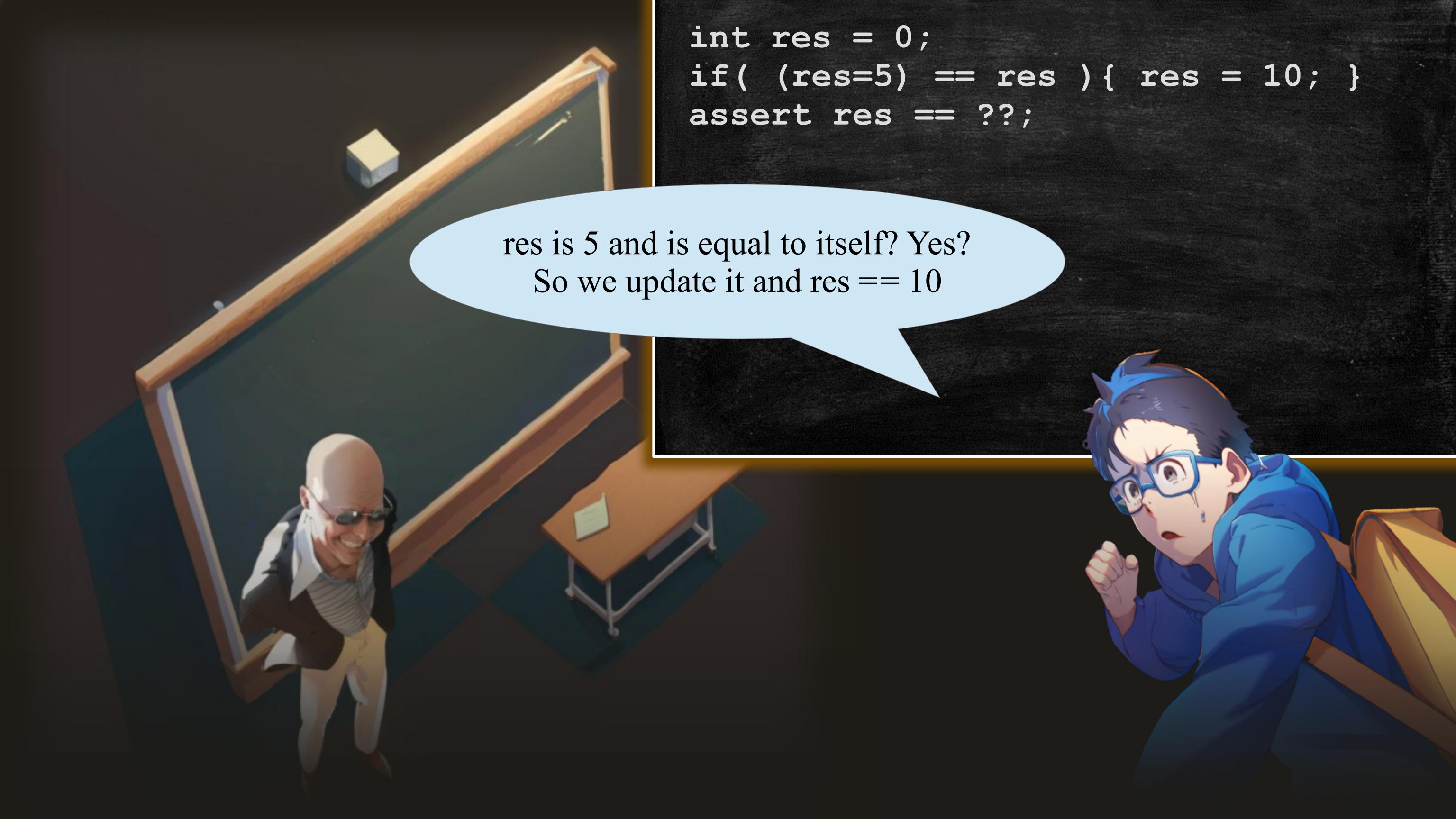


Here you are.

So, what is this doing?

```
int res = 0;  
if( (res=5) == res ){ res = 10; }  
assert res == ??;
```

res, res, ... res ... ?



```
int res = 0;  
if( (res=5) == res ){ res = 10; }  
assert res == ??;
```

res is 5 and is equal to itself? Yes?
So we update it and res == 10



```
int res = 0;  
if( (res=5) == res ){ res = 10; }  
assert res == 10;
```

res is 5 and is equal to itself? Yes?
So we update it and res == 10



Cool

What happens in this case instead?



```
int res = 0;  
//if( (res=5) == res ){ res = 10; }  
if( res == (res=5) ){ res = 10; }  
assert res == ??;
```



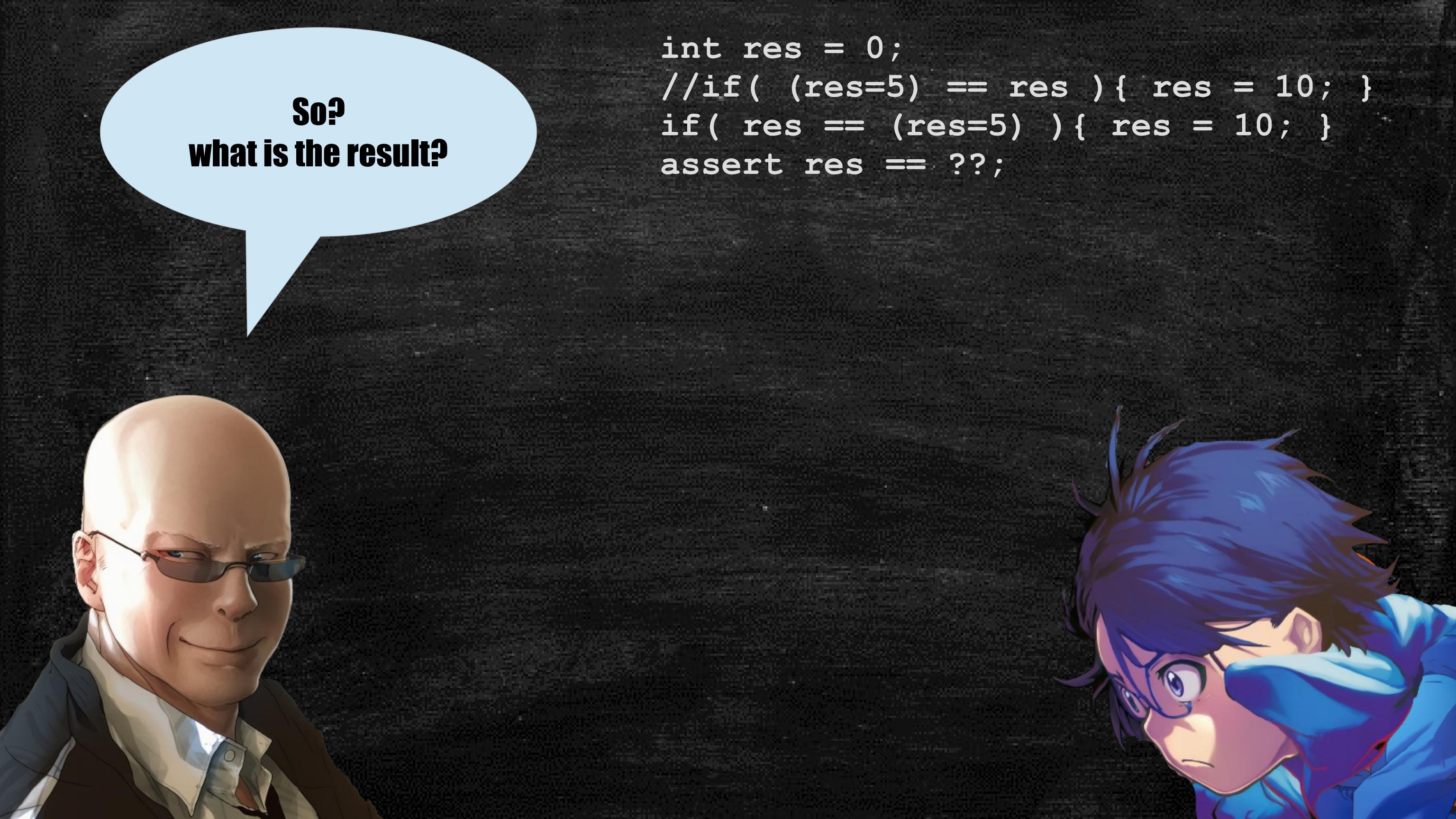
What happens in this case instead?



```
int res = 0;  
//if( (res=5) == res ){ res = 10; }  
if( res == (res=5) ){ res = 10; }  
assert res == ??;
```

You switched the
expressions around the equals equals





So?
what is the result?

```
int res = 0;  
//if( (res=5) == res ){ res = 10; }  
if( res == (res=5) ){ res = 10; }  
assert res == ??;
```



So?
what is the result?

```
int res = 0;  
//if( (res=5) == res ){ res = 10; }  
if( res == (res=5) ){ res = 10; }  
assert res == ??;
```

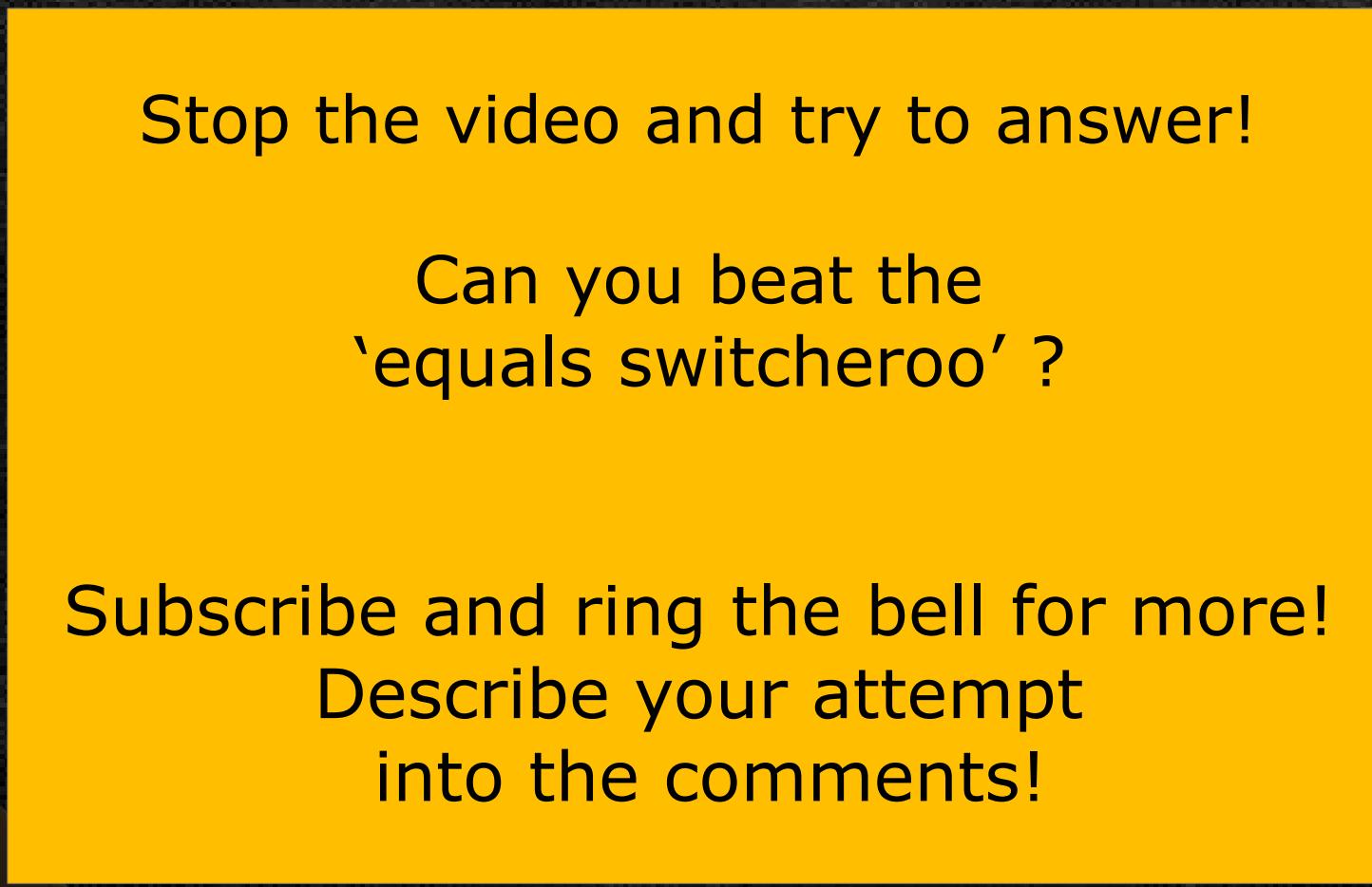


It looks like it's the same.
I mean, == is clearly a
commutative operation.



**So?
what is the result?**

```
int res = 0;  
//if( (res=5) == res ){ res = 10; }  
if( res == (res=5) ){ res = 10; }  
assert res == ??;
```



Stop the video and try to answer!

Can you beat the
'equals switcheroo' ?

Subscribe and ring the bell for more!
Describe your attempt
into the comments!





**So?
what is the result?**

```
int res = 0;  
//if( (res=5) == res ){ res = 10; }  
if( res == (res=5) ){ res = 10; }  
assert res == ??;
```



It looks like is the same,
I mean, == is clearly a
commutative operation.



Wait a minute, would he even ask the question if it was the same?

A man in a dark suit and glasses holds a deck of cards, looking thoughtful. A woman in a green dress with a white collar and a blue bow tie holds a fan, looking down at her hand. A large, light blue thought bubble contains the text.

There must be a difference then!



Or ... is he thinking that
I will think this,
and so it's the same??



It's as if we were
playing poker!



What is going on?
We ARE playing poker!

Another hallucination?



Pupon has 3 cards in his hand.
They are the 3 res

```
int res = 0;  
if(res==(res=5)) {res=10; }
```

By the way, my cards are not very good ...



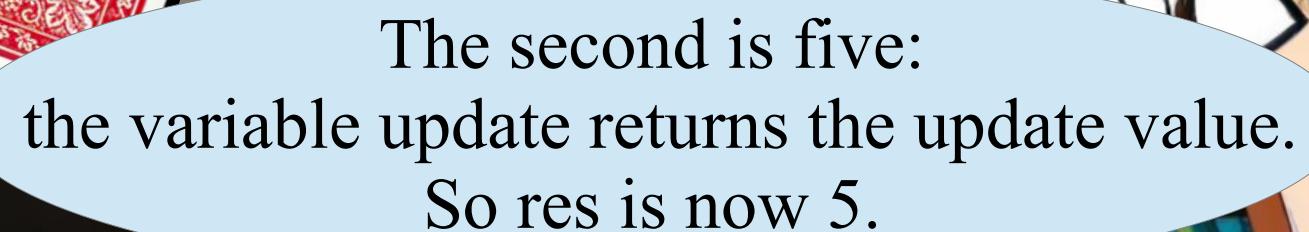
The first one must be zero,
because at the start res is zero

```
int res = 0;  
if(res==(res=5)) {res=10;}
```



```
int res = 5;  
if(res==(res=5)) {res=10; }
```

The first one must be zero,
because at the start res is zero



The second is five:
the variable update returns the update value.
So res is now 5.



```
int res = 5;  
if(res==(res=5)) {res=10; }
```

The first card is evaluated already,
so it is not changing

zero is not equal to 5, so we skip
the body of the if!

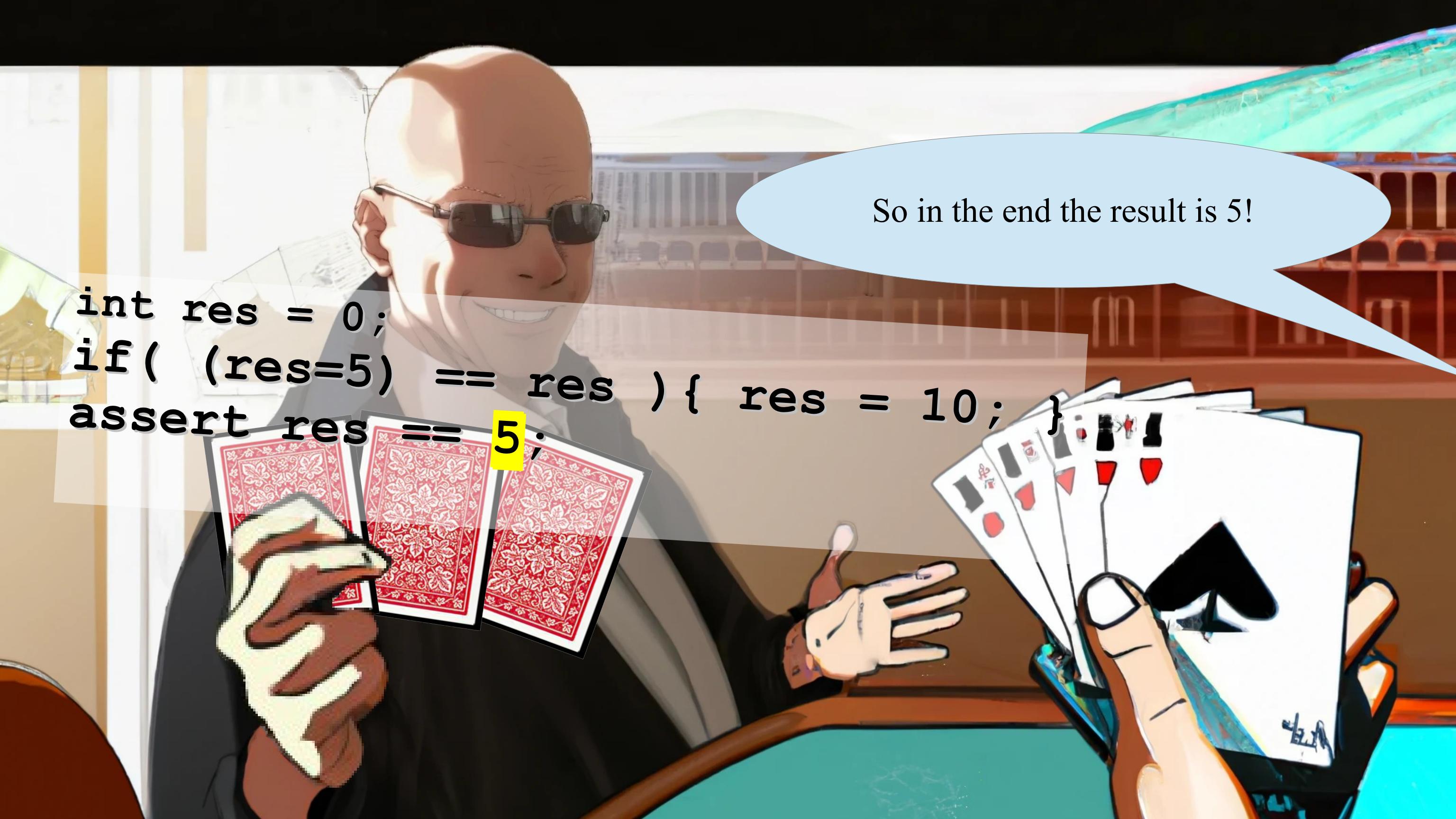


```
int res = 5;  
if(res==(res=5)) {res=10; }
```

The first card is evaluated already,
so it is not changing

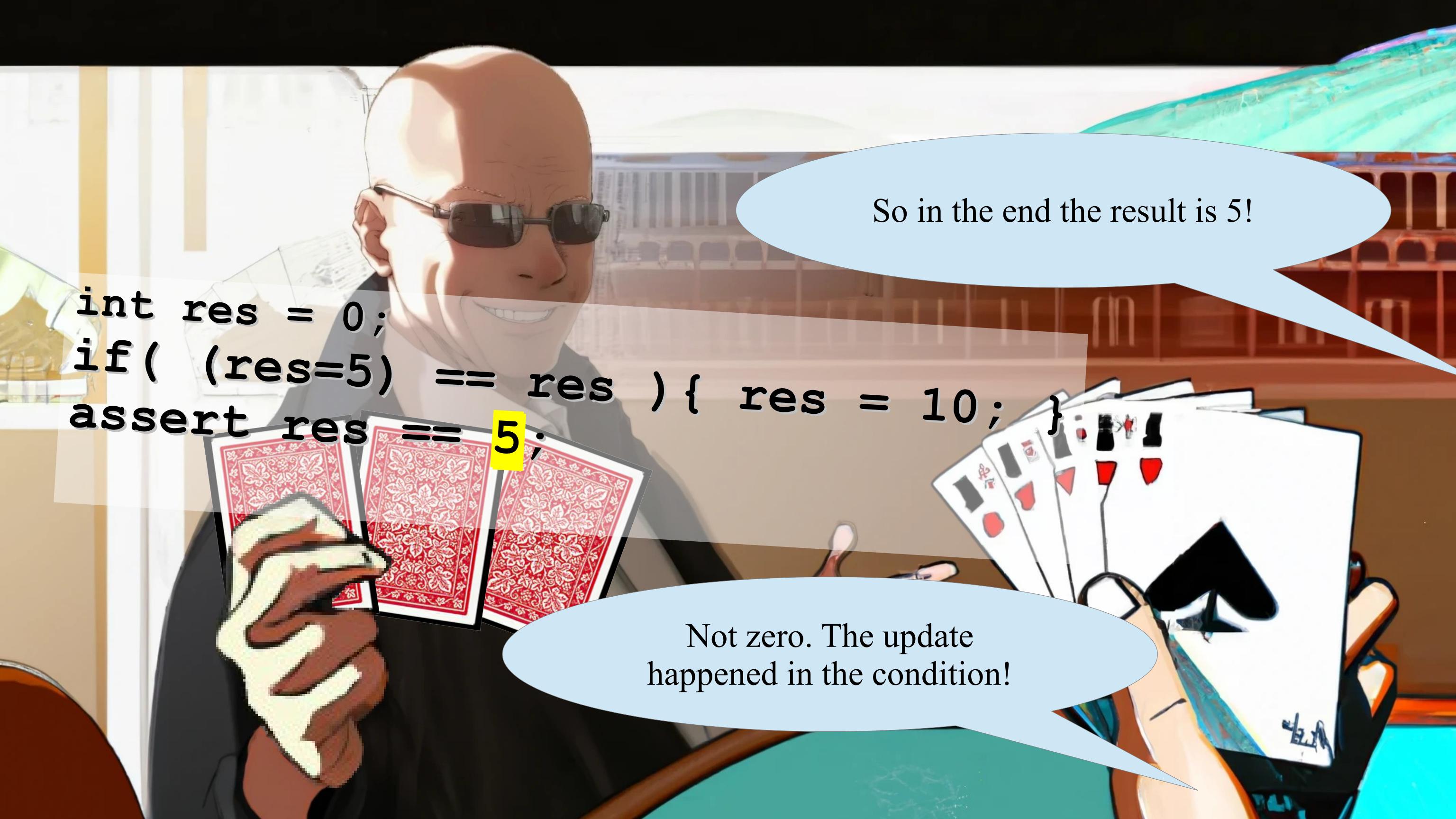
zero is not equal to 5, so we skip
the body of the if!

So the last card is not played!



```
int res = 0;  
if( (res=5) == res ) { res = 10; }  
assert res == 5;
```

So in the end the result is 5!



```
int res = 0;  
if( (res=5) == res ) { res = 10; }  
assert res == 5;
```

So in the end the result is 5!



Not zero. The update
happened in the condition!

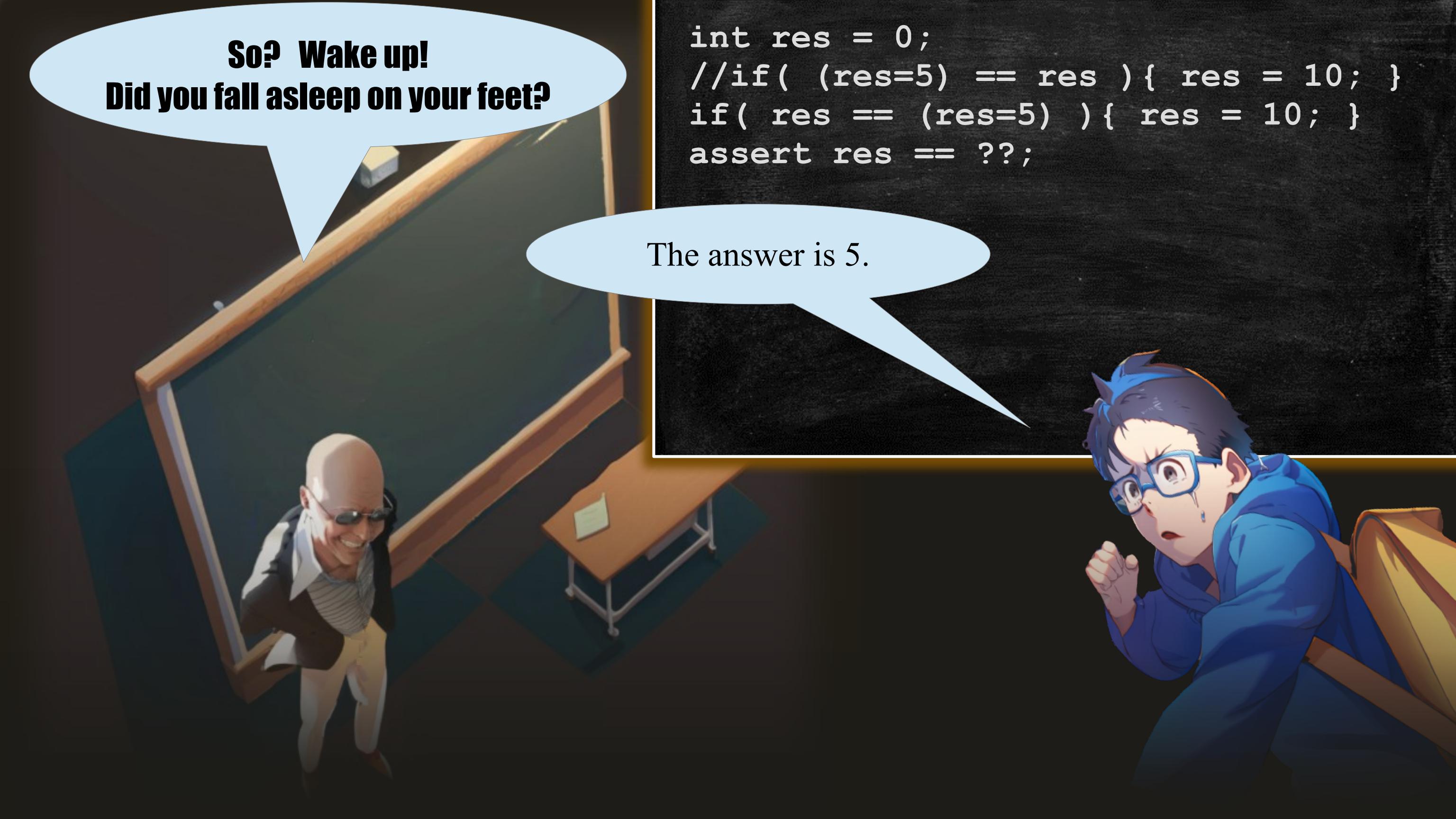


**So? Wake up!
Did you fall asleep on your feet?**



```
int res = 0;  
//if( (res=5) == res ){ res = 10; }  
if( res == (res=5) ){ res = 10; }  
assert res == ??;
```

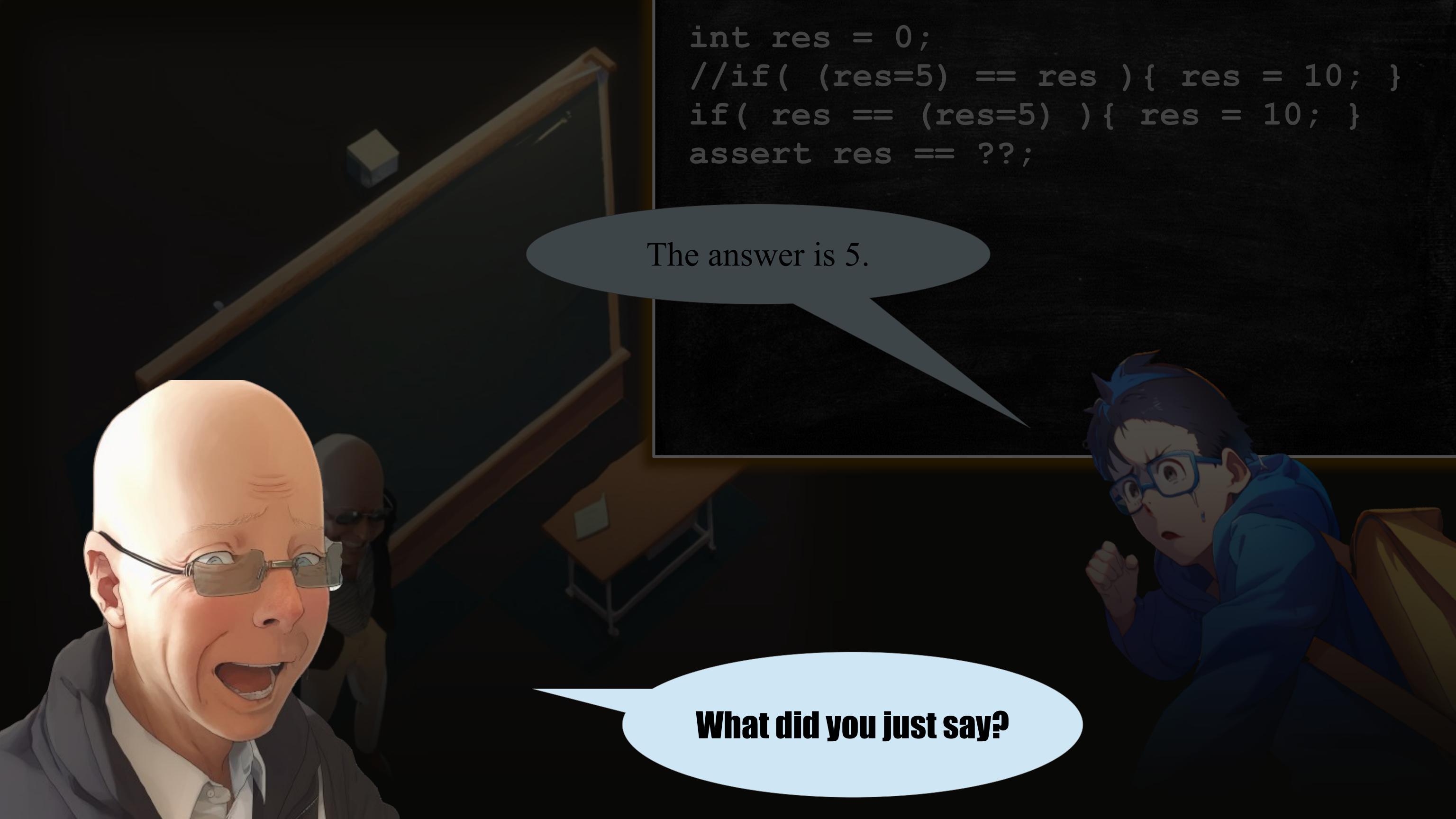




**So? Wake up!
Did you fall asleep on your feet?**

```
int res = 0;  
//if( (res=5) == res ){ res = 10; }  
if( res == (res=5) ){ res = 10; }  
assert res == ??;
```

The answer is 5.

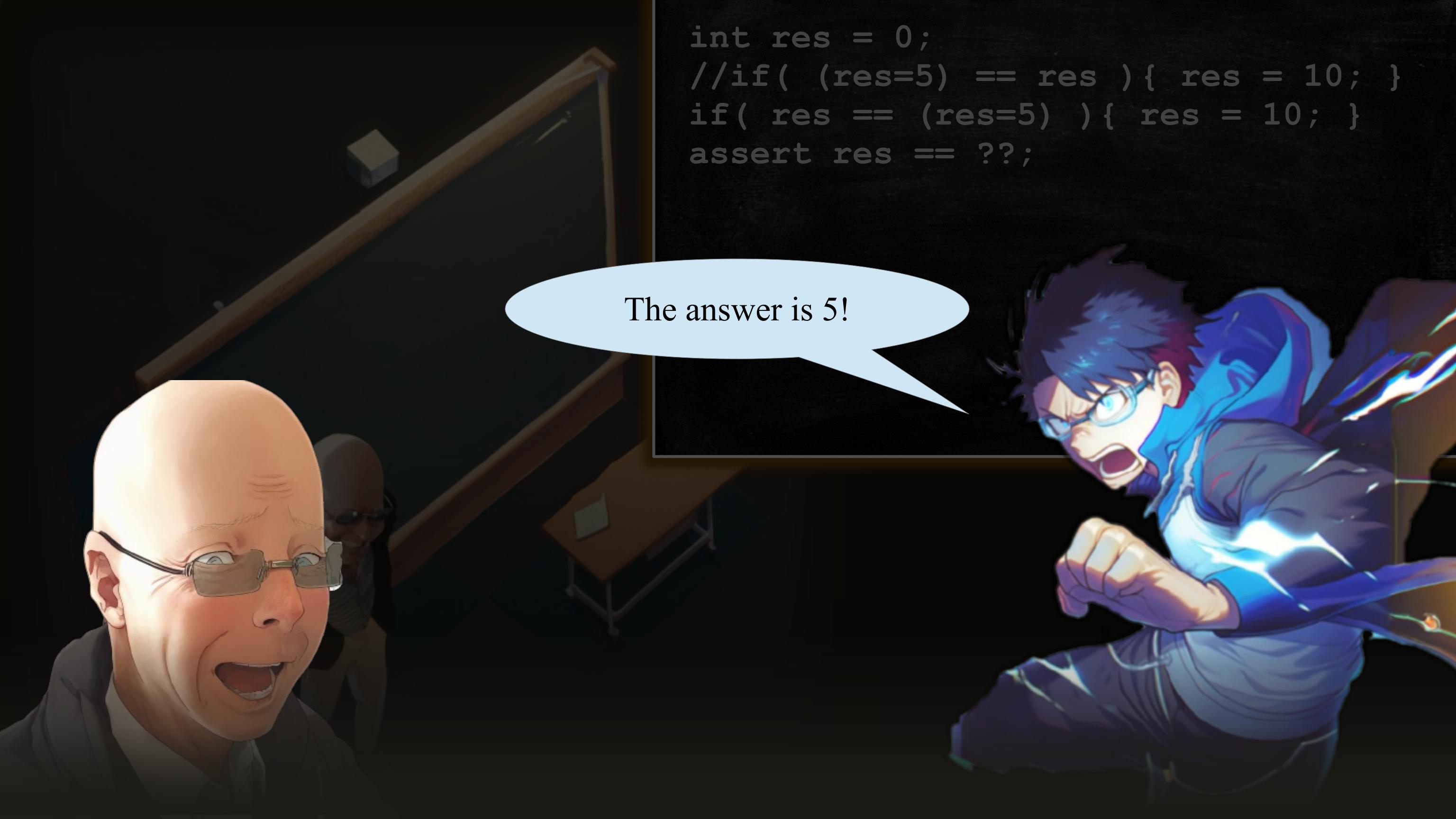


```
int res = 0;  
//if( (res=5) == res ){ res = 10; }  
if( res == (res=5) ){ res = 10; }  
assert res == ??;
```

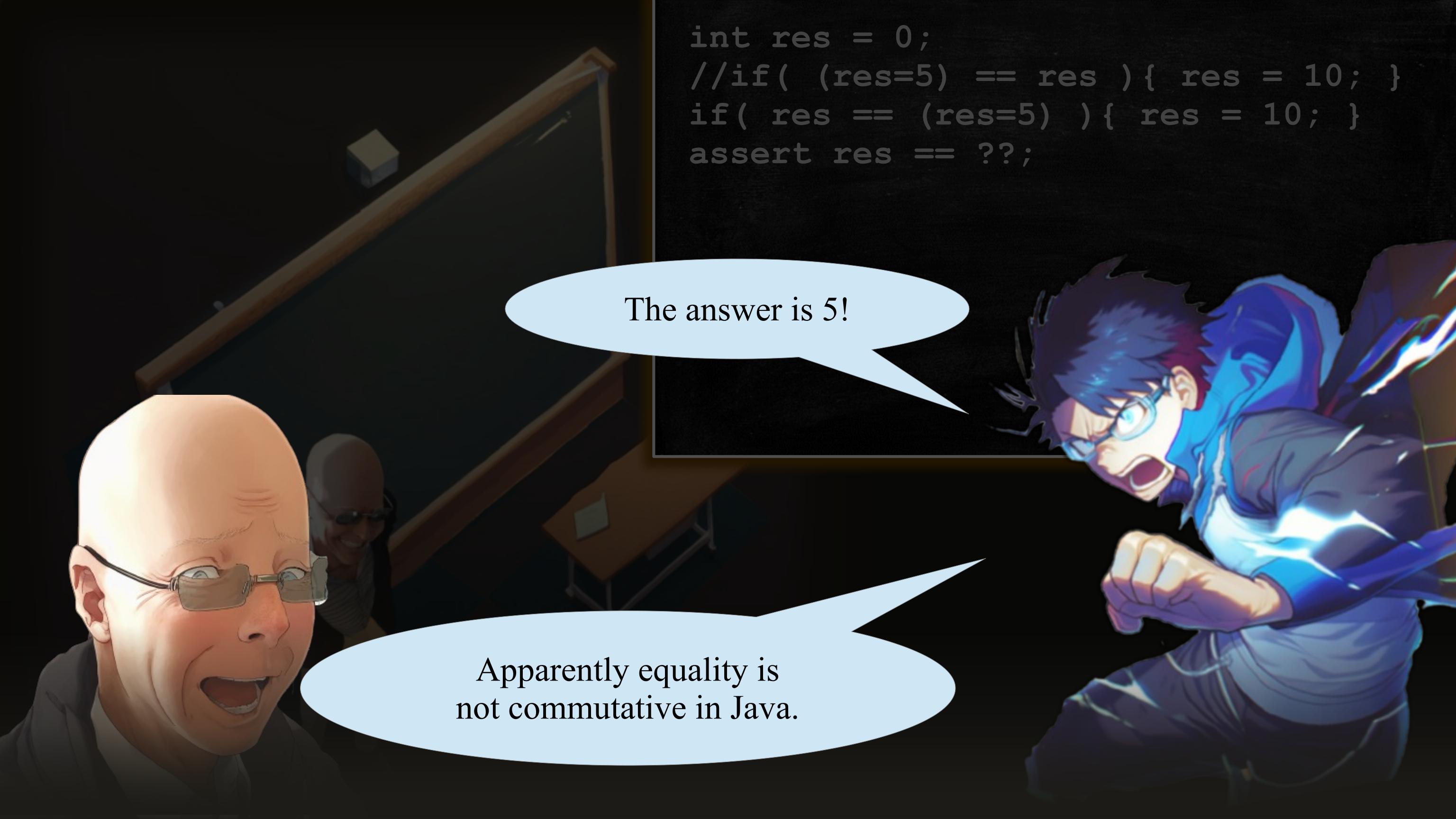
The answer is 5.

What did you just say?

```
int res = 0;  
//if( (res=5) == res ){ res = 10; }  
if( res == (res=5) ){ res = 10; }  
assert res == ??;
```



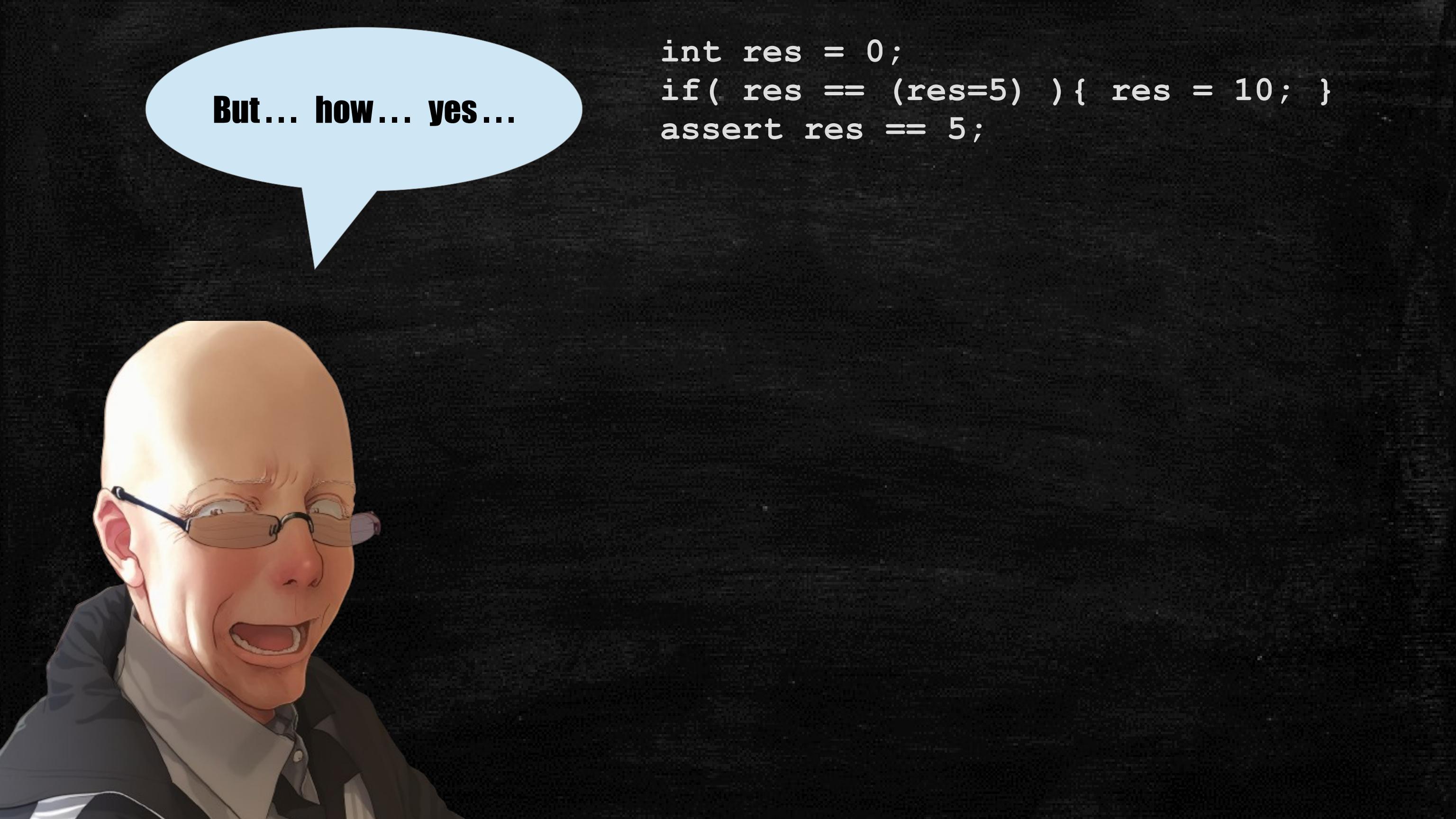
The answer is 5!



```
int res = 0;  
//if( (res=5) == res ){ res = 10; }  
if( res == (res=5) ){ res = 10; }  
assert res == ??;
```

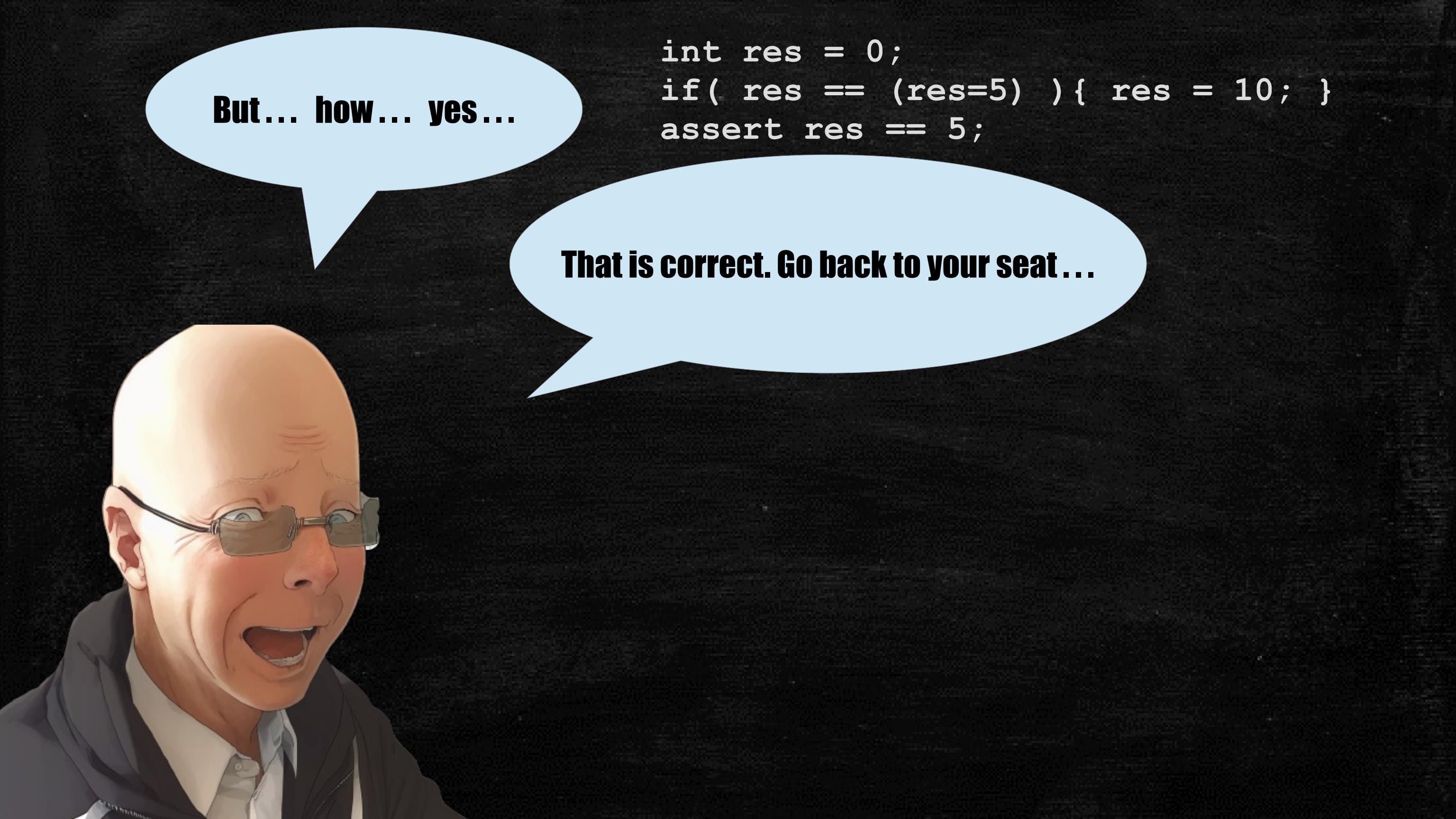
Apparently equality is
not commutative in Java.

The answer is 5!

A cartoon caricature of Steve Jobs, showing his bald head, glasses, and intense expression. He is wearing a dark suit jacket over a light-colored shirt.

But... how... yes...

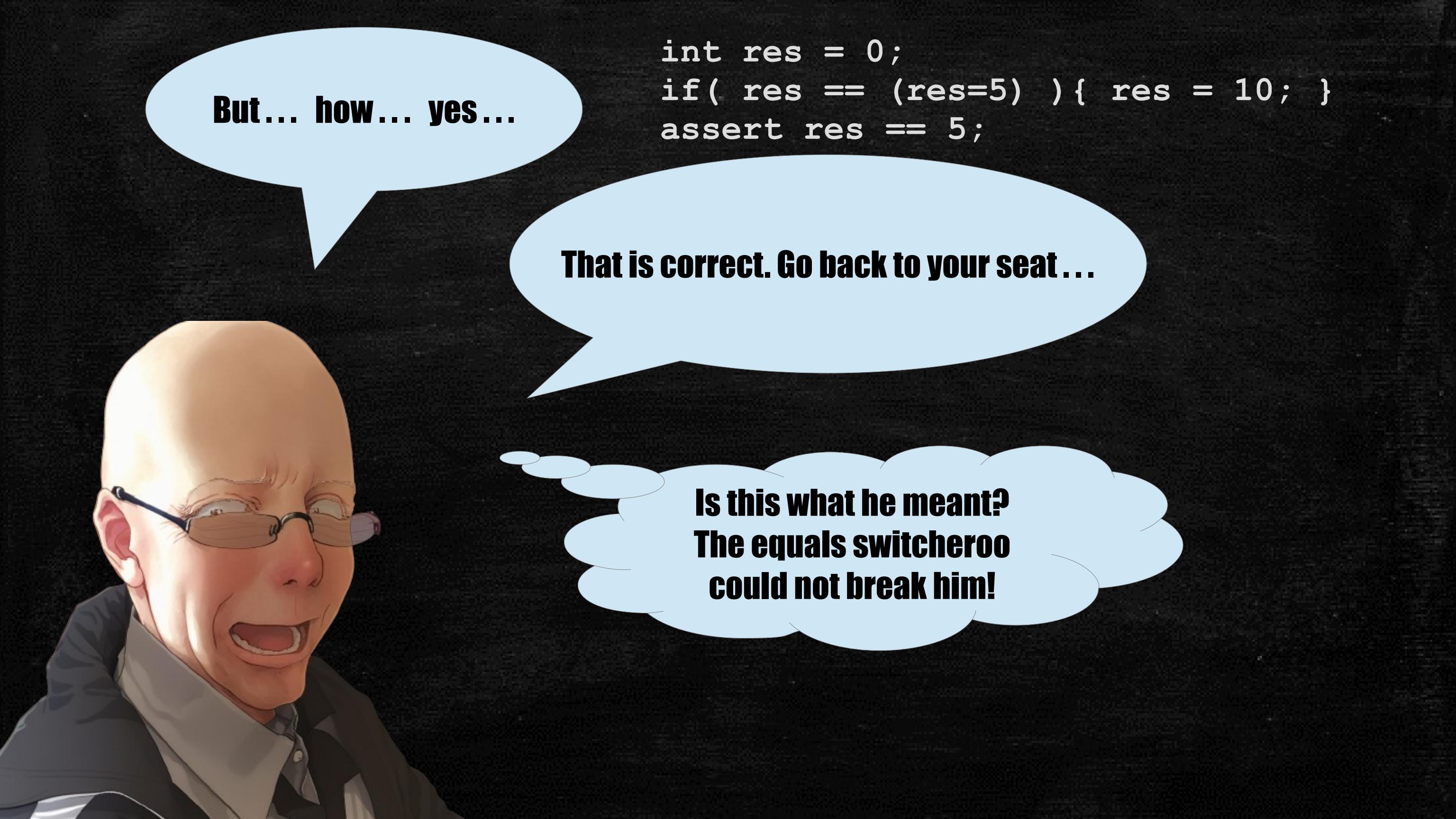
```
int res = 0;  
if( res == (res=5) ){ res = 10; }  
assert res == 5;
```

A caricature of Steve Jobs with a large head, wearing glasses and a dark suit, looking surprised with his mouth open.

But... how... yes...

```
int res = 0;  
if( res == (res=5) ){ res = 10; }  
assert res == 5;
```

That is correct. Go back to your seat...

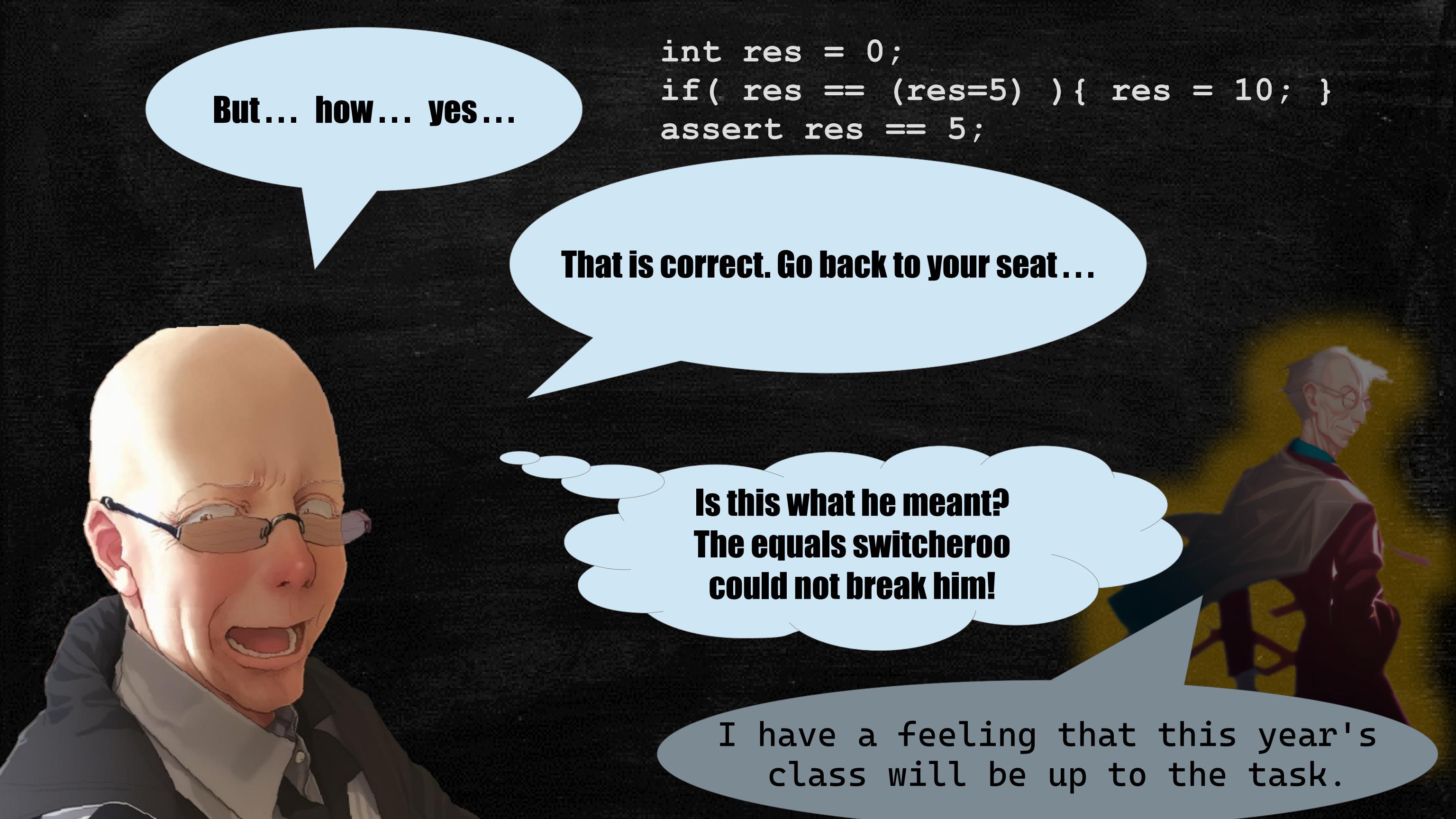


But... how... yes...

```
int res = 0;  
if( res == (res=5) ){ res = 10; }  
assert res == 5;
```

That is correct. Go back to your seat...

**Is this what he meant?
The equals switcheroo
could not break him!**



But... how... yes...

```
int res = 0;  
if( res == (res=5) ){ res = 10; }  
assert res == 5;
```

That is correct. Go back to your seat...

Is this what he meant?
The equals switcheroo
could not break him!

I have a feeling that this year's
class will be up to the task.

After the lecture, Dany rushes to his room



After the lecture, Dany rushes to his room



After the lecture, Dany rushes to his room



After the lecture, Dany rushes to his room



Again!

Every lecture feels like a fight to the death!

The class looked confused,
as if I said something absurd.

After the lecture, Dany rushes to his room



Again!

Every lecture feels like a fight to the death!

The class looked confused,
as if I said something absurd.

How many of them would
have been able to answer?

After a hot shower, Dany mumbles about ...



After a hot shower, Dany mumbles about ...



Is this a sensible way to code?

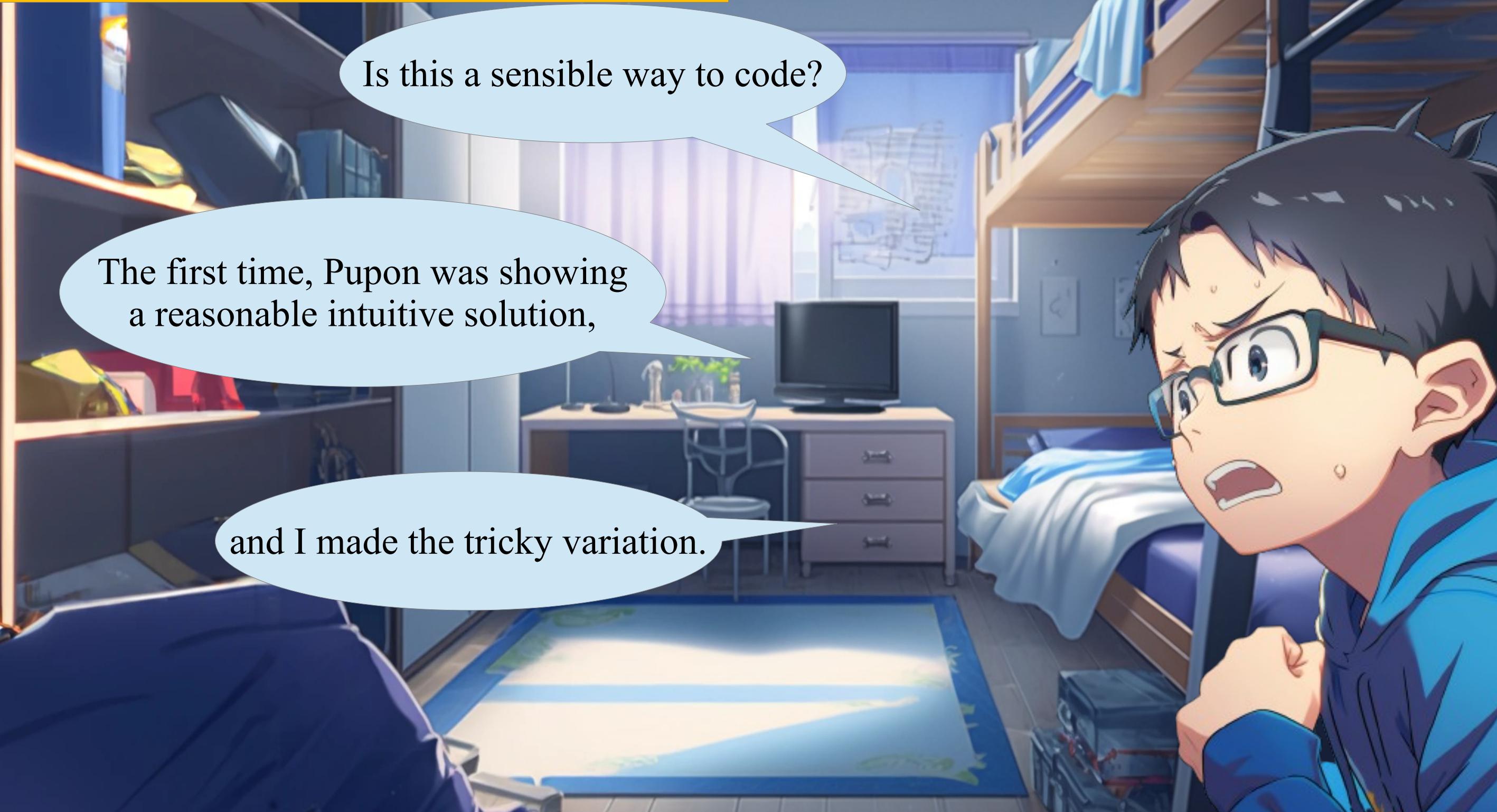
After a hot shower, Dany mumbles about ...



The first time, Pupon was showing
a reasonable intuitive solution,

Is this a sensible way to code?

After a hot shower, Dany mumbles about ...





But this time ...



But this time ...

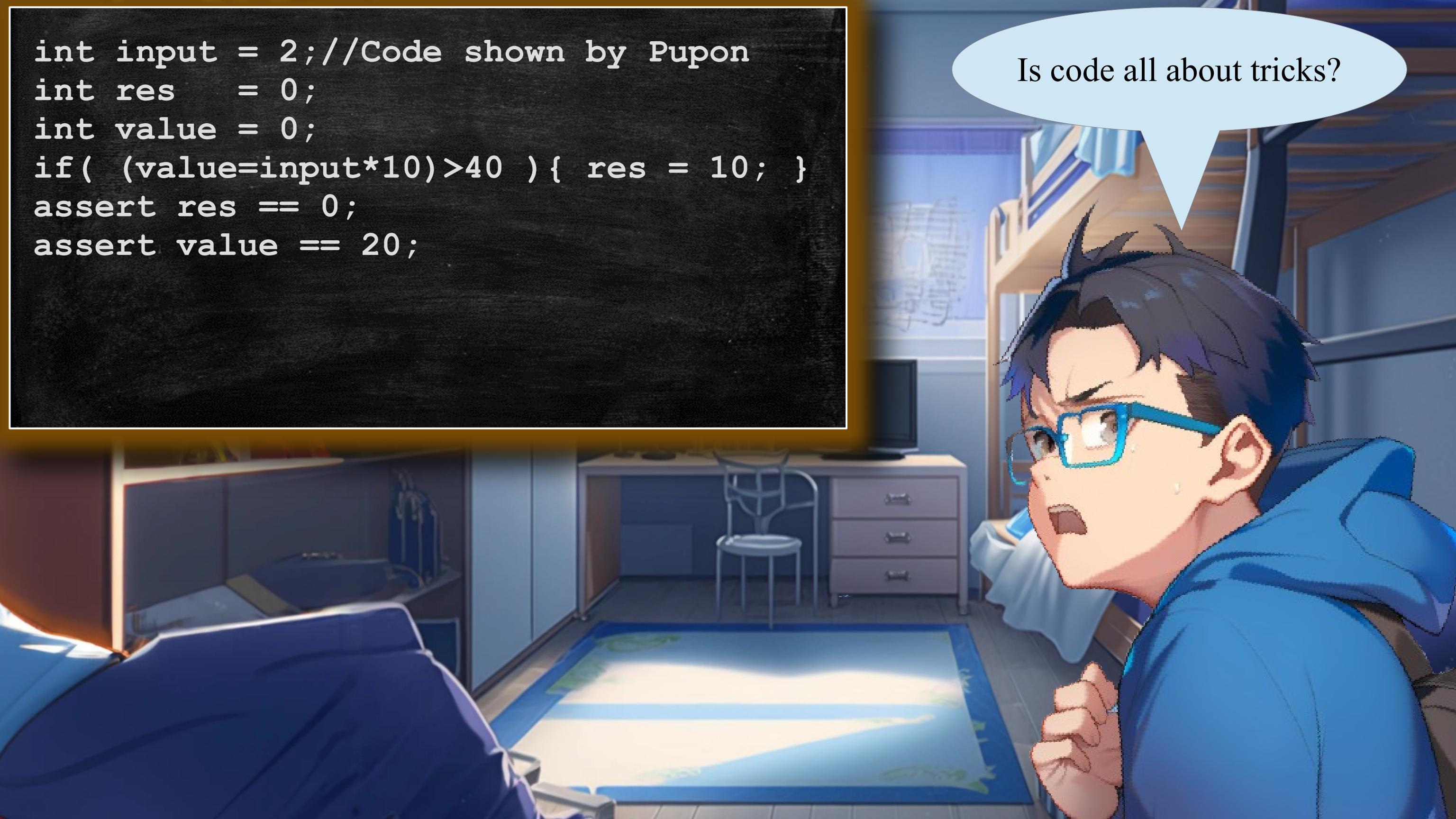
the lecture and the questions
was all about the tricky part!



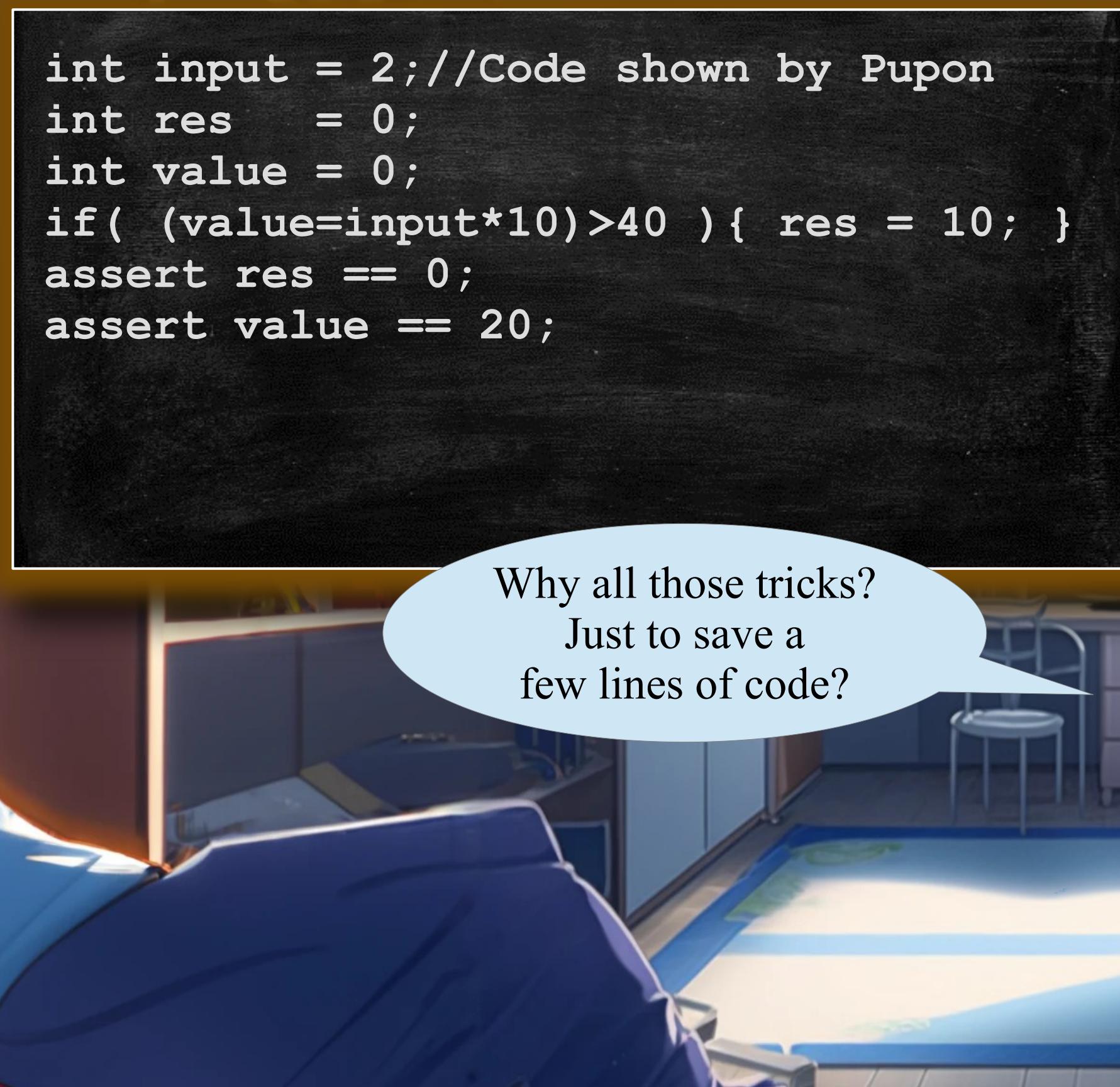
Is code all about tricks?

```
int input = 2;//Code shown by Pupon  
int res = 0;  
int value = 0;  
if( (value=input*10)>40 ){ res = 10; }  
assert res == 0;  
assert value == 20;
```

Is code all about tricks?



```
int input = 2;//Code shown by Pupon  
int res = 0;  
int value = 0;  
if( (value=input*10)>40 ){ res = 10; }  
assert res == 0;  
assert value == 20;
```



Why all those tricks?
Just to save a
few lines of code?



Is code all about tricks?

```
int input = 2;//Code shown by Pupon  
int res = 0;  
int value = 0;  
if( (value=input*10)>40 ){ res = 10; }  
assert res == 0;  
assert value == 20;
```

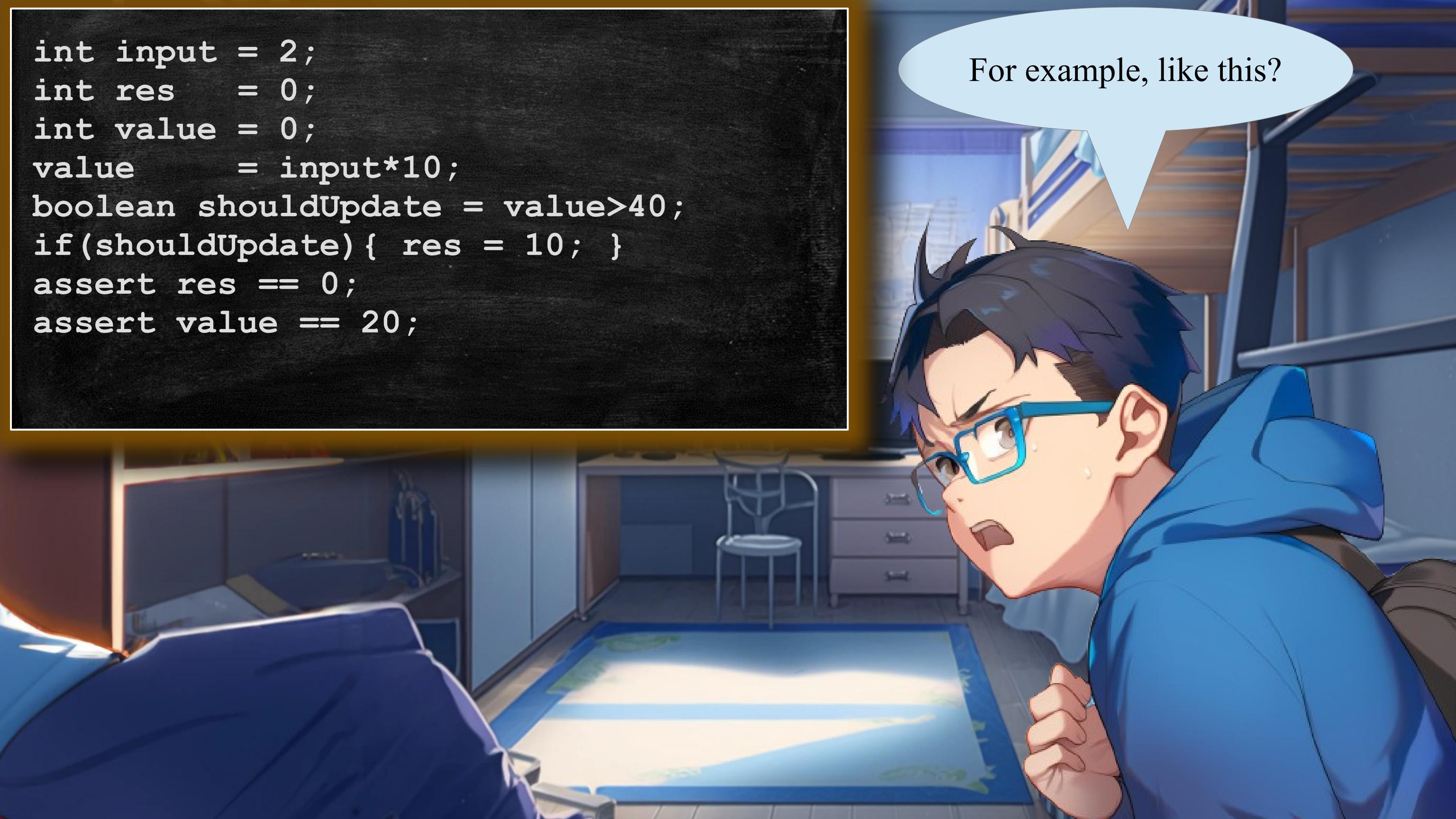
Why all those tricks?
Just to save a
few lines of code?

Should we instead make
code more readable by
making it longer?

Is code all about tricks?

```
int input = 2;  
int res = 0;  
int value = 0;  
value = input*10;  
boolean shouldUpdate = value>40;  
if(shouldUpdate) { res = 10; }  
assert res == 0;  
assert value == 20;
```

For example, like this?



```
int input = 2;  
int res = 0;  
int value = 0;  
value = input*10;  
boolean shouldUpdate = value>40;  
if(shouldUpdate) { res = 10; }  
assert res == 0;  
assert value == 20;
```

Would this be better?



```
int input = 2;  
int res = 0;  
int value = 0;  
value = input*10;  
boolean shouldUpdate = value>40;  
if(shouldUpdate) { res = 10; }  
assert res == 0;  
assert value == 20;
```

Would this be better?

Is it only a few lines every time,
or do some of those tricks
save massive amounts of code?



Suddenly, Dany hears a rustling sound coming from the door

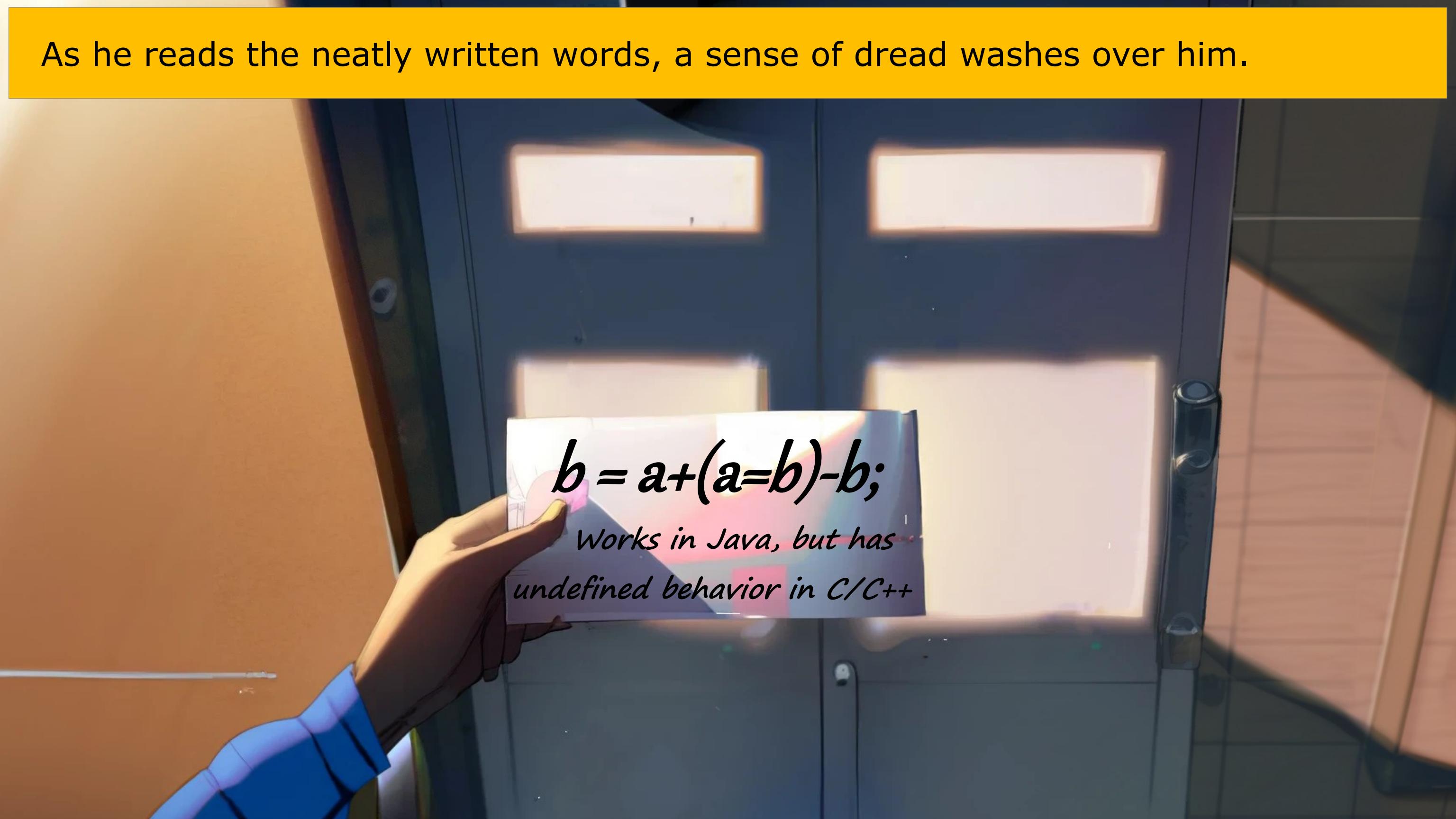


He turns to look and sees a small piece of paper sliding under the door.

He reaches down and picks up the note, his hands shaking as he unfolds it.



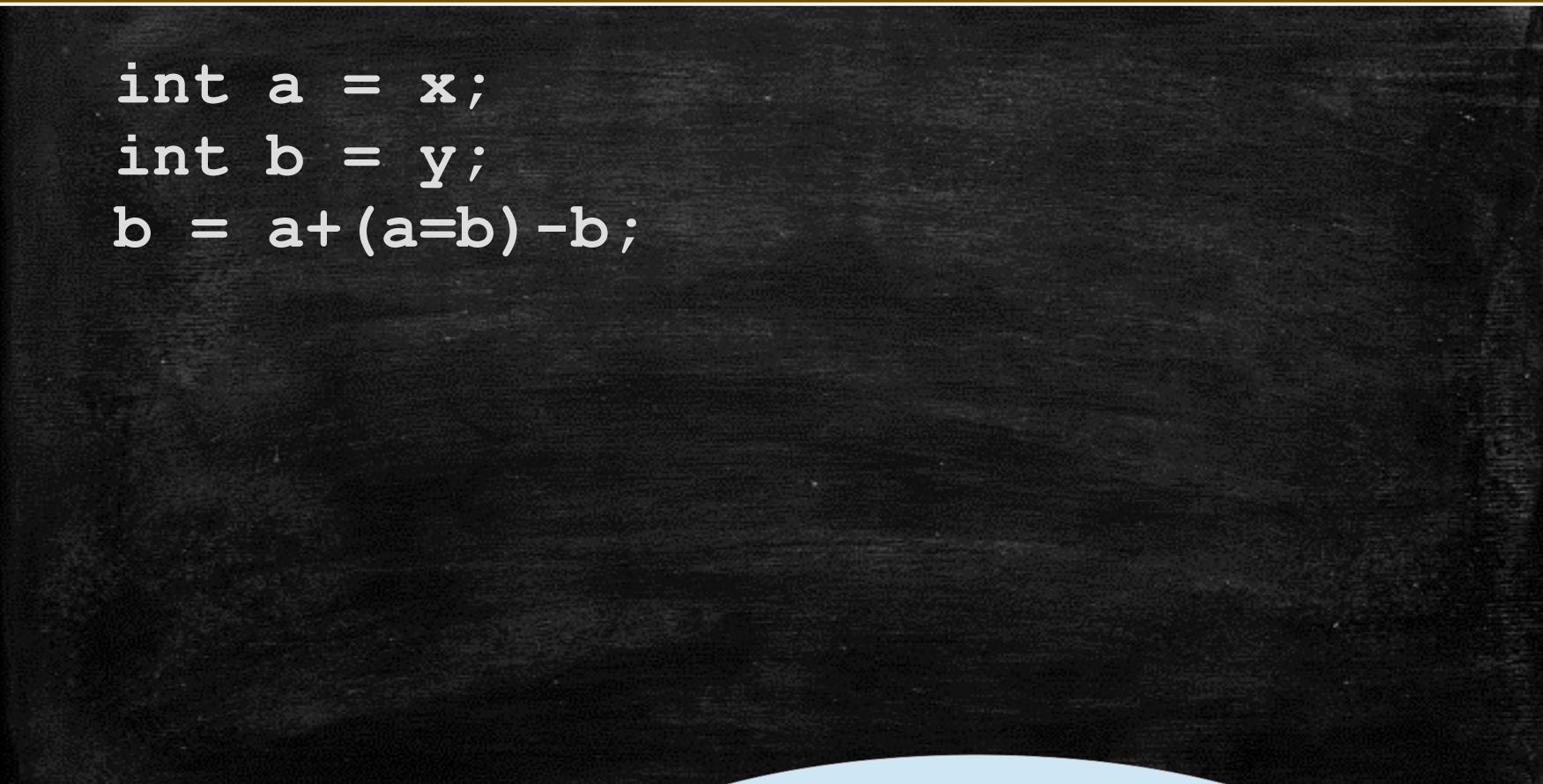
As he reads the neatly written words, a sense of dread washes over him.



$b = a + (a = b) - b;$

*Works in Java, but has
undefined behavior in C/C++*

```
int a = x;  
int b = y;  
b = a+(a=b)-b;
```



Is this what I think it is?



```
int a = x;  
int b = y;  
b = a+(a=b)-b;
```

Is this what I think it is?

Going step by step, using
x and y as symbolic values



```
int a = x;  
int b = y;  
b = x+(a=b)-b;
```

We evaluate a into x



```
int a = x;  
int b = y;  
b = x+(a=y)-b;
```

We evaluate the first b into y

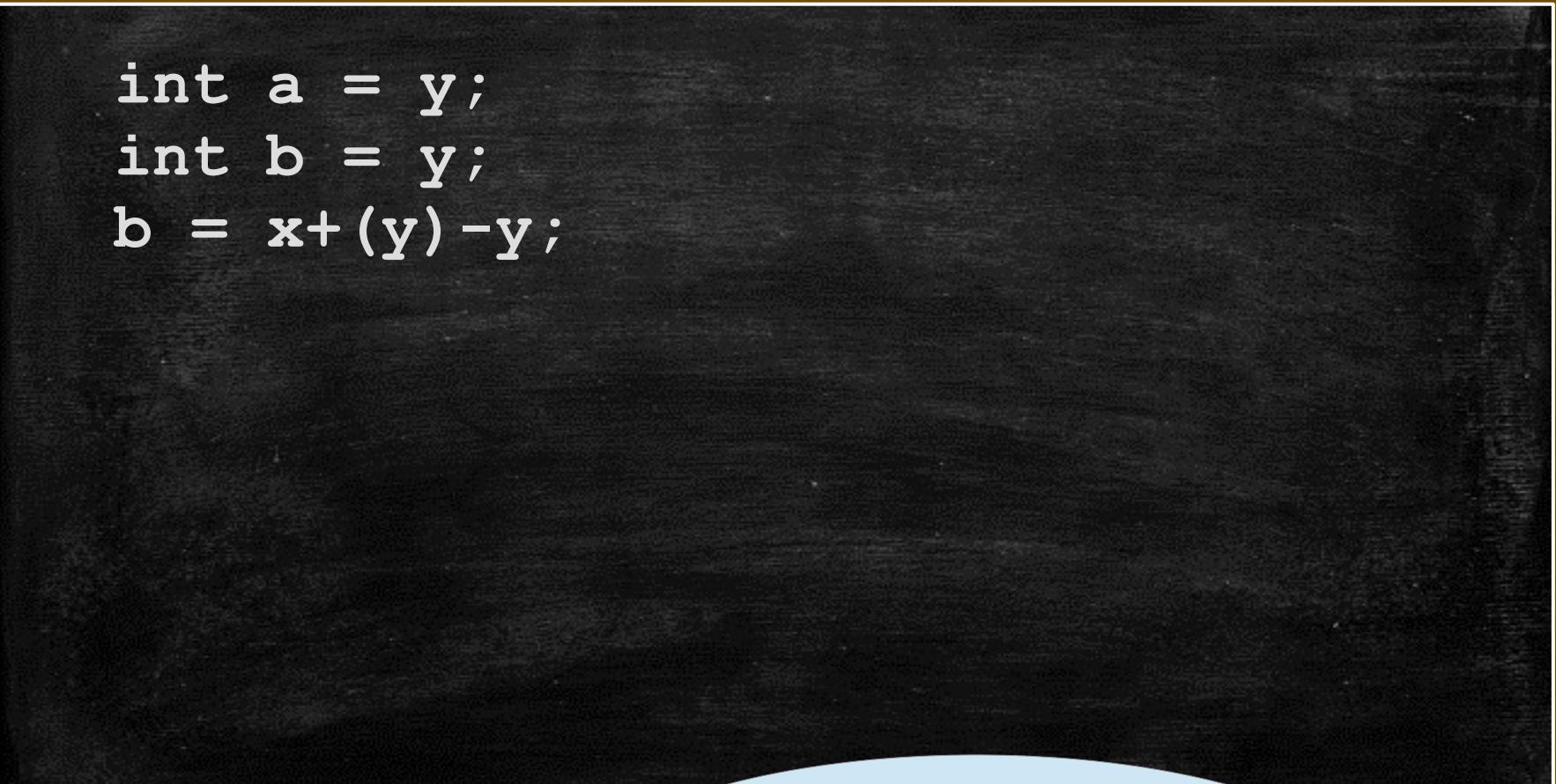


```
int a = y;  
int b = y;  
b = x+(y)-b;
```

We update a with y



```
int a = y;  
int b = y;  
b = x+(y)-y;
```



We evaluate the second b

And now it's just simple math



```
int a = y;  
int b = y;  
b = x;
```



Finally we update b with x

```
int a = y;  
int b = x;  
  
//swap two variables  
// b = a+(a=b)-b;  
//Works in Java, but has  
//undefined behavior in C/C++
```

And here it is



```
int a = y;  
int b = x;  
  
//swap two variables  
// b = a+(a=b)-b;  
//Works in Java, but has  
//undefined behavior in C/C++
```

And here it is

A one liner swapping
two variables in Java



```
int a = y;  
int b = x;  
  
//swap two variables  
// b = a+(a=b)-b;  
//Works in Java, but has  
//undefined behavior in C/C++
```

And here it is

A one liner swapping
two variables in Java

Also, it has undefined
behavior in C/C++?



```
int a = y;  
int b = x;  
  
//swap two variables  
// b = a+(a=b)-b;  
//Works in Java, but has  
//undefined behavior in C/C++
```

And here it is

A one liner swapping
two variables in Java

Also, it has undefined
behavior in C/C++?

So, different languages are different indeed;
even if they may look superficially similar



Then Dany turns the paper, and on the other side there is another shocking message!



*Every good coding practice
was originally just a trick.*



02: The sky is infinite





Credits

- Story: Marco
- Art: MidJourney, NijiJourney, Dall-E
- - Wording: Marco, chatGPT
- Composition: Marco
- Thanks to all my friends for providing great feedback!

Next episode on Feb 13 2023