


The stage is set for a new lecture to commence



Conditionals:

We can use ifs to choose between options



**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 holds
Thus we skip the body
and we get 0.**



**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.**

**Value equal input times ten
is an expression producing
the assigned value.**

Step by step

```
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

2 times 10 is 20

We update value to 20

value = 20 is now just '20'

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

with false, we skip the if body

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



As you can see

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



So, you, .. come here!



Do you mean... Me?!?

So, you, .. come here!



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 then res

So we update res to 10!

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

*does it get
un-updated?*

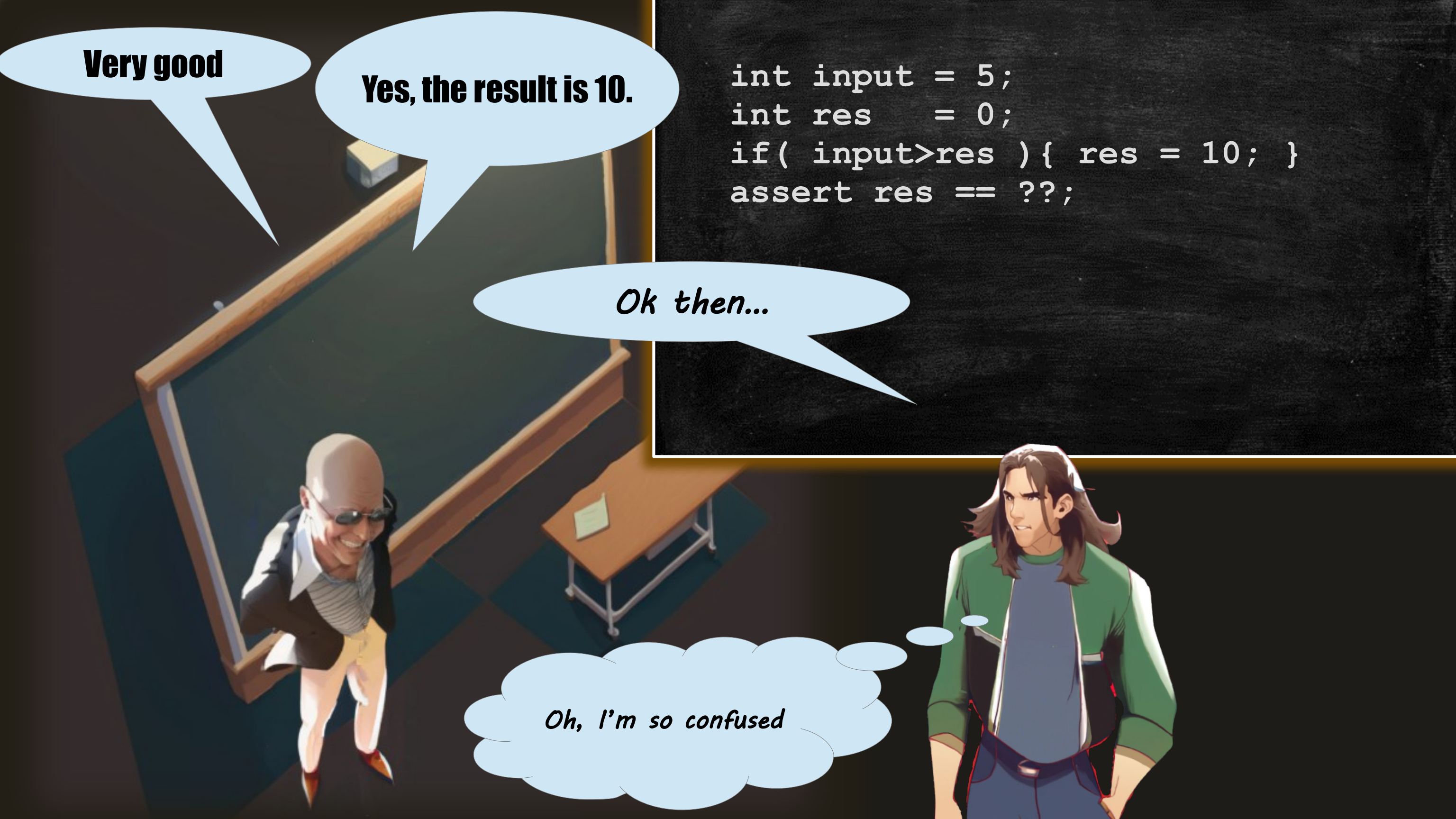
Very good


Yes, the result is 10.

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

Ok then...


Oh, I'm so confused





Me?!?

Next, you above!

A 3D-rendered lecture hall with tiered seating. In the foreground, a cartoon character with spiky blue hair, glasses, and a blue jacket is shown in a close-up, screaming with a wide-open mouth and sweat drops on his face. In the background, a man with a shaved head and a goatee, wearing a green jacket, sits in a row of seats, looking towards the character. A large, light blue thought bubble originates from the man, containing the text "This is it, he is going to take revenge!".

This is it,
he is going to take revenge!



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

Cool



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 equal equals



**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.



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' ?




**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.




A dark, stylized illustration of a man in a suit and tie, smiling and playing cards. He is holding a fan of cards in his left hand, which includes the Ace of Spades and several red cards. His right hand is also holding a fan of cards. A large, light blue thought bubble is positioned above his head, containing the text "Wait a minute, would he even ask the question if it was the same?". The background is dark and indistinct.


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

A man in a dark suit and tie is seated at a table, smiling. He is holding a fan of playing cards in his left hand, which includes several red-backed cards. His right hand is resting on the table. In the foreground, another hand is visible, holding a fan of cards that includes the King of Hearts and the Queen of Hearts. A large, white, cloud-like thought bubble is positioned above the man's head, containing the text "There must be a difference then!". The background is dark and indistinct.

There must be a difference then!

A man with a shaved head, wearing sunglasses and a dark suit with a striped tie, is seated at a table. He is holding a fan of playing cards in his right hand, showing the red patterned backs. His left hand is open and gesturing towards a fan of cards held by another person on the right. The cards on the right include the Ace of Spades, King of Spades, Queen of Spades, Jack of Spades, 10 of Spades, 9 of Spades, 8 of Spades, 7 of Spades, 6 of Spades, 5 of Spades, 4 of Spades, 3 of Spades, 2 of Spades, and the Ace of Hearts. A large, white, cloud-like thought bubble is positioned above the man's head, containing the text: "Or.. is he thinking that I will think this, and it is thus the same??" The background is dark and indistinct.

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


A man with a shaved head, wearing dark sunglasses, a white shirt, a striped tie, and a dark suit jacket, is smiling while holding three playing cards with red floral patterns. He is seated at a green poker table. In the foreground, another hand is visible, holding a fan of playing cards including the Ace of Spades, King of Hearts, Queen of Diamonds, Jack of Clubs, and 10 of Hearts. A thought bubble above the man's head contains the text "It is as if we were playing poker!". The background shows a dimly lit room with a large window or balcony railing.

It is 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;}
```


0

A man with a shaved head and sunglasses is smiling while holding a fan of playing cards. He is wearing a dark jacket over a light-colored shirt. The background is a stylized, colorful illustration of a casino or gaming area with a green felt table and a red and white striped wall. Two speech bubbles and a code snippet are overlaid on the image. The first speech bubble explains the initial state of a variable 'res'. The code snippet shows an if-statement that checks if 'res' is equal to the result of 'res=5'. The second speech bubble explains the result of the update operation.

The first one must be zero.
Because at the start res is zero.

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

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



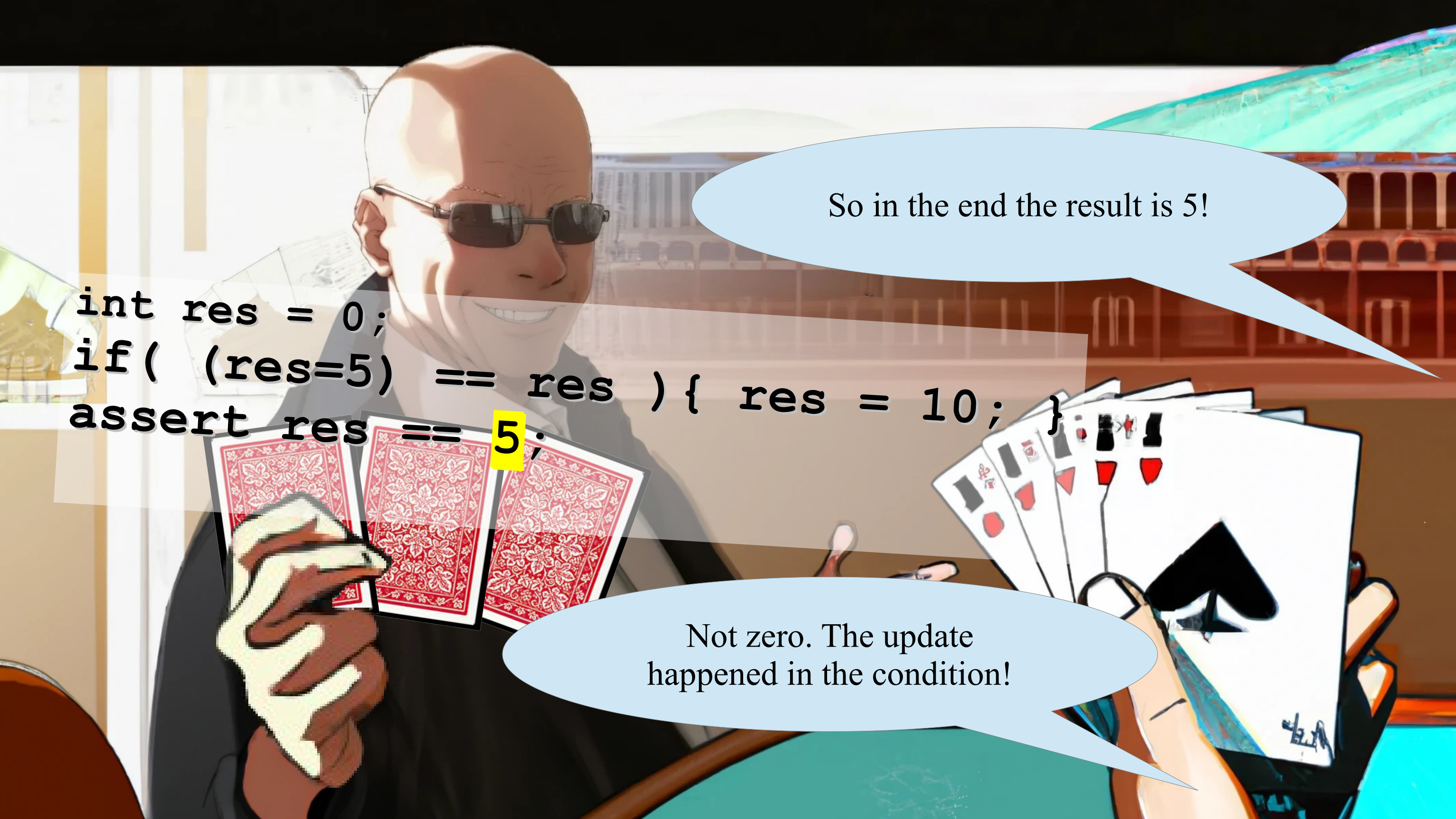
The first card is evaluated already,
so it is not changing

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

0 5

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

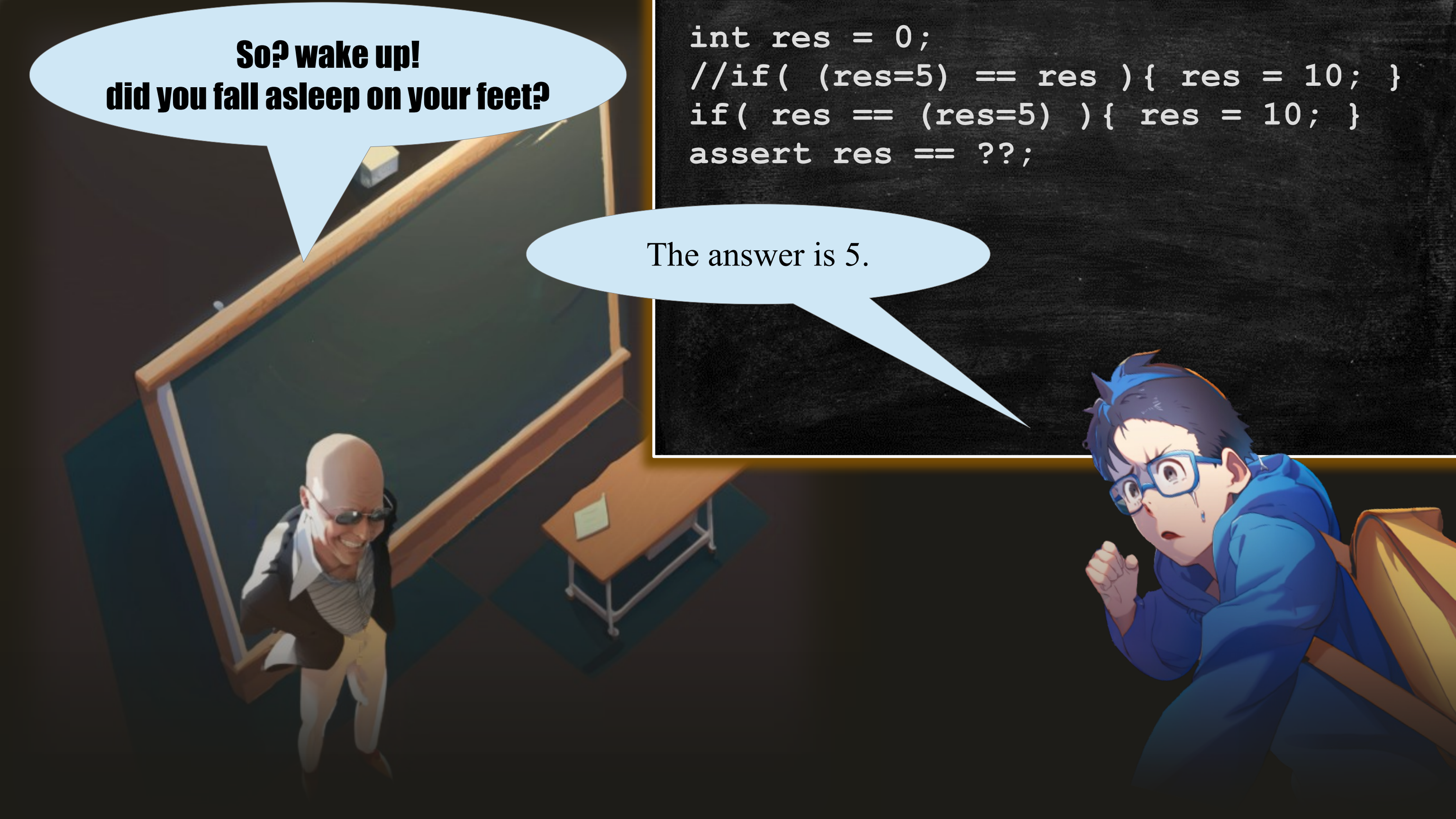
So the last card is not played!

A man with a bald head and sunglasses is smiling while holding a fan of playing cards. The cards are mostly red-backed, with some face cards visible. The background is a blurred indoor setting with a green table.

```
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!

A 3D-rendered classroom scene. On the left, a bald teacher with sunglasses and a black jacket over a striped shirt is smiling and pointing towards a student. On the right, a young man with blue hair and glasses, wearing a blue hoodie, is sitting at a desk, looking surprised with his mouth open. In the background, there is a large green chalkboard with a wooden frame and a small wooden desk with a green paper on it. A speech bubble from the teacher contains the text "So? wake up! did you fall asleep on your feet?". A speech bubble from the student contains the text "The answer is 5.". In the top right corner, there is a black box containing white C++ code.

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.


A classroom scene with a teacher and a student. The teacher, a bald man with glasses, is on the left, looking surprised. The student, a young man with glasses and a blue hoodie, is on the right, looking shocked. A chalkboard in the background displays a C code snippet. A speech bubble from the teacher asks a question, and another from the student asks for clarification.

**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.

What did you just say?

A cartoon illustration of a classroom. On the left, a bald professor with glasses and a striped shirt under a dark jacket stands next to a green chalkboard, smiling. On the right, a student with spiky blue hair and glasses, wearing a blue jacket, is shown in a dynamic, shouting pose with his fist clenched. A speech bubble from the student points towards the professor. In the top right corner, a black box contains white text of a Java code snippet. A speech bubble from the professor is at the bottom.

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

The answer is 5!

Apparently equality is
not commutative in Java.