

Up to now, we've been able to create a class when we created the class, we were able to define our attributes and methods, both in the definition of the class and the implementation of the class.

We have some attribute values that sat there empty, and these attributes are supposed to be used by our methods.

Now, what is a constructor?

Well, a constructor is a special type of method, and we have the option of when we are defining a class of defining a constructor method.

Now, a constructor method is automatically executed by the system when we instantiate a brand new object, so soon as we create an object, this method gets fired and any code within it gets run.

And when we create a constructor method, there are some fixed rules that we have to abide by. And these are a constructor has to have a fixed name of constructor.

We can't name anything else.

And the method must be public and can only have important parameters. And like I just mentioned a second ago, I just want to point out the constructors are optional, but you will find they are very useful and used in a lot of different classes.

So now let's ask the question, well, why do we need constructor's why you've seen the cluster, we're creating the car class we need to be able to do something with a constructor to fill our attributes of the class.

So here a list, but obviously it can be extended.

We can dynamically set the state of an object.

We can use them to initialize attributes for an object.

We can allocate resources for objects and we can use them to execute any code that we think needs to be executed, when an object is created, when it's instantiated.

Here we go, and what we're going to do is go back to the top of the class because just like a normal method, we have to include it in the definition section and then go down and implement some code for the constructor.

So what I'm going to do up here at the top and I tend to put my constructor at the top of my class definition,

I'll create a new line and we'll start with methods as usual.

But this time, remember, we have to use the word constructor.

Which means it will get executed for each instance of the class that is created.

And what we're going to do is using importing, remember, that's the only one we can use, we can't use exporting or anything like that.

So we fill the make model, the number of seats and the maximum speed.

And what I will do is method constructor.

And then specify our code to fill our class attributes, so we want our make attribute to equal the make that's going to be imported into our constructor model will equal the model.

Now seats will equal seats, the max speed or equal max speed.

And then if you remember, we have a Static attribute some cost data number of cars and what we will do every time an object is created from this class, we will increment this by one so we can add that into the constructor and the same time.

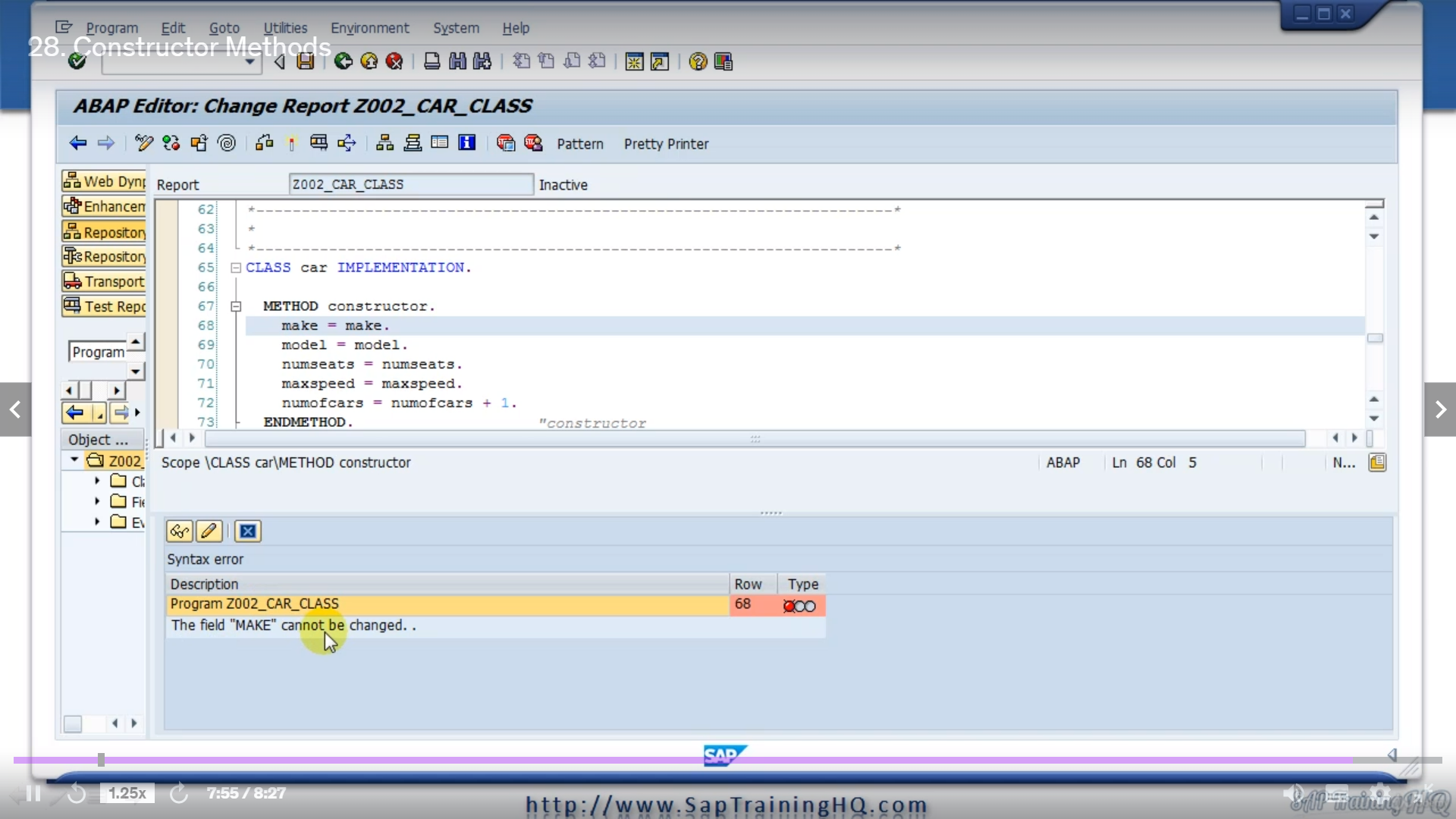
So we can just do a number of cars equals number of cars plus one.

here, let me go back up.

When I define this construct, you can see I'm saying I am importing make model NUM seats and maximum

speed, so I am defining full parameters here and giving them names are normally within a method.

This is fine.



It doesn't matter what we name them because they are local to the method.

But let's consider this when we are implementing the code.

What actually referencing attributes of the object that have the same names as the parameters specified in the constructor and well, I know that the ABAP system is going to complain of this.

So if I try and do a syntax check.

We can see here it's telling me the field of make cannot be changed and it's not very descriptive.

It's not really telling me what the issue is.

But I know that what's happening is the system doesn't know what Make referring to.

Is it the constructor?

Is it the attribute?

It doesn't know.

So we need to qualify this with something.

So in the next video, we will see how we can fix this syntax error by implementing a new way of referring to attributes within our class.