

Now, we have been through the description and use of compound interfaces and alias names.

Let's bring them together and discuss how we use alias names in compound interfaces in this lesson.

So as well as defining alias names in a class I have mentioned, you can also create them in an interface

as well, and the syntax for doing that is exactly the same as declaring them in a class.

But it's important to note, though, that in interfaces like we have here with Interface AI, which

is included in Interface B, which is included in Interface C, we have to code things a little bit

differently.

Now, as mentioned previously, interface components can only occur once in a compound interface, so

when we have this type of scenario where where nest in, let's say, three levels deep and we want to

create an alias name for method and one interface, see, we have to declare the alias name in Interface

B also, because without doing that, we wouldn't be able to make use of the alias name in Interface

C, it wouldn't work.

This is one of the specific rules for alias names in compound interfaces structured this way.

If you focus on the code, on the right of the screen, you can see a very simple class where we're

using interface, see, well, in order for the object in my class to properly reference method and

one, it has to use the alias AMTA, which is defined in interface C, AMTA is declared for I Am One,

which is also an alias, but defined in Interface B Nehalem.

One alias is defined for the actual method in Interface A.

So that's how a class can use an alias defined in its facey routine all the way back to interface a

four method M1.