Subject: Vector(const Vector& v, int)
Posted by Novo on Thu, 25 Aug 2016 22:03:33 GMT
View Forum Message <> Reply to Message

Hello,

What is a reason to keep the deep copy constructor Vector(const Vector& v, int)?
It prevents automatic generation of default copy constructor for classes using Vector. Default move constructor can be autogenerated, but copy constructor has to be coded manually.

Am I missing something?

TIA

Subject: Re: Vector(const Vector& v, int)
Posted by mirek on Fri, 26 Aug 2016 15:13:32 GMT
View Forum Message <> Reply to Message

With C++11, the reason is to make 'clone' explicit.

We want to prevent this:

Vector<int> a, b;

a = b;

and force the user to write either

a = pick(b);

or

a = clone(b);

(That said, maybe there could be a better way how to achieve this, maybe make copy-construtor explict would work too, but addition int parameter is sort of legacy).

Mirek

Subject: Re: Vector(const Vector& v, int)
Posted by Novo on Fri, 26 Aug 2016 21:47:08 GMT
View Forum Message <> Reply to Message

mirek wrote on Fri, 26 August 2016 11:13With C++11, the reason is to make 'clone' explicit.

(That said, maybe there could be a better way how to achieve this, maybe make copy-constructor explicit would work too, but addition int parameter is sort of legacy).

Thank you.
Explicit copy-constructor doesn't force to call "clone" for some reason.
Maybe, it makes sense to make pick/clone optional, and let compiler auto-generate default copy constructor? This legacy move semantics doesn't seem to have much sense when compiler already does good job moving temporary data for you. You still will be able to call "pick". Typing code which is not really necessary is very annoying ...

Thanks.

Subject: Re: Vector(const Vector& v, int)
Posted by mirek on Sat, 27 Aug 2016 04:52:03 GMT
View Forum Message <> Reply to Message

Novo wrote on Fri, 26 August 2016 23:47
Maybe, it makes sense to make pick/clone optional, and let compiler auto-generate default copy constructor? This legacy move semantics doesn't seem to have much sense when compiler already does good job moving temporary data for you. You still will be able to call "pick". Typing code which is not really necessary is very annoying ...

Well, the idea is to prevent accidental deep copy... In most cases, you want 'pick', but with implicit '=' for deep copy, it is easy to forget.

Mirek
