

Re: Correct C++ tutorial part 3 "Intro to loops" available (Windows, mingw/msvc/std)

Source: http://coding.derkeiler.com/Archive/C_CPP/comp.lang.cpp/2005-01/3468.html

From: Attila Feher (attila.feher_at_lmf.ericsson.se)

Date: 01/27/05

Date: Thu, 27 Jan 2005 15:28:45 +0200

Alf P. Steinbach wrote:

[SNIP]

>>> ** If you use integer update you should use integer type for the*

>>> *variable.*

>>

>> *Why? You have introduced your lines with the word reasoning, but I*

>> *see no reason told for this statement.*

>

> *It seems to boil down to two issues: why it's not a good idea to use a*

> *misleading type (maintainability), and whether a floating point type*

> *can always represent all values of an integral type exactly*

> *(technical).*

Why would a floating point type misleading if I need a floating point value?

Furthermore – on the same note – why would I care about it representing or not integer values, if I needed a floating point one in the first place?

[SNIP]

> *Yes, to the latter two sentences. I wonder whether you are happy with*

> *the resolution of that issue as it turned out in the rest of the*

> *thread? That is, adding the parenthetical expression "unless you*

> *really know what you're doing"?*

That is one way to put it. :-) But I guess there is another and probably more important reason/cause: do you need a floating point iteration or not?

:-) And then comes the know what you do, do what you know. :-)

[SNIP]

> *Iterators and pointers as loop control variables is a good point,*

> *thanks.*

No problemo. :-)

> *The discussion above was trapped in the implicit context of numeric*

> *loop control variables.*

comp.lang.c++. Re: Correct C++ tutorial part 3 "Intro to loops" available (Windows, mingw/msvc/std)

And it possibly should for that very–beginner part, but IMHO the rule itself may need to be worded differently. So that it suggests to the reader that the rule is supposed to say: do not use floating point unless you really need it (IMHO this goes not only for loops), so basically tell what is wrong, instead of identifying what is right. Because limiting the possible types (in this case/stage) will rule out valid cases, vali