

Re: derangement: code review request

Source: http://coding.derkeiler.com/Archive/C_CPP/comp.lang.c/2004-10/3361.html

From: Tim Rentsch (txr_at_alumnus.caltech.edu)

Date: 10/29/04

Date: 29 Oct 2004 10:17:19 -0700

rlb@hoekstra-uitgeverij.nl (Richard Bos) writes:

> *Tim Rentsch <txr@alumnus.caltech.edu> wrote:*
>
>> *rlb@hoekstra-uitgeverij.nl (Richard Bos) writes:*
>>
>>> *Tim Rentsch <txr@alumnus.caltech.edu> wrote:*
>>>
>>>> *A couple of the names use abbreviations as part of the name (eg,*
>>>> *'fam_size'). I recommend using the whole word, and always avoiding*
>>>> *abbreviations.*
>>>
>>>> *I disagree. Even for normal abbreviations, not just for acronyms,*
>>>> *industry-standard abbrevs are probably more legible than the whole word.*
>>>
>>>> *I mean, my main program deals with entering data for advertisement. Can*
>>>> *you imagine how much larger and less legible my code would've been if*
>>>> *I'd used advertisement_width_in_columns everywhere I know have adv_cols?*
>>>
>>>> *How about 'ad_columns'? My unabridged dictionary includes 'ad' as a*
>>>> *word. If 'ad_columns' is still too long for you, 'ad_width' has the*
>>>> *same number of letters as 'adv_cols'.*
>>>
>>>> *Whatever. It's still an abbreviation, and avoiding abbreviations would*
>>>> *have been counterproductive, in this case.*

Sorry, I thought the meaning was obvious – to use words, and avoid abbreviations that aren't words. Nothing wrong with using 'gas' (in the sense of "gasoline"), 'motel' (short for "motor hotel"), or 'transistor' (short for "transfer resistor"), since they are all used as words. Perhaps the phrasing should have been to avoid non-words, in particular abbreviations that aren't words.

>>>> *void*
>>>
>>>> *Ew, ganoo style.*
>>>
>>>> *It's hard to know how to respond to a statement that is clearly*

comp.lang.c: Re: derangement: code review request

> > *nothing more than name calling. If you have a comment about the usage*
> > *rather than just an objection to some people who happen to use it, how*
> > *about letting us know what that is?*
>
> *It's ugly.*

I might say the same thing about the all-on-one-line style. :)

Seriously though, how about a statement that is somewhat more objectively verifiable?

> *It splits the type of the function over two lines for no good*
> *reason [...],*

Just curious – how can you say that when you haven't heard what the reasons are? Or are you saying you've considered all possible reasons and determined that none of them are good?

Also – what is the metric for whether a reason is "good"?

> *and separates the return*
> *type from the identifier in a way that nobody would even consider using*
> *for objects. You don't write*
>
> *int*
> *i, j, counter;*
>
> *do you? Well, then, why do so for functions?*

Clearly the two cases are different, and so different reasoning applies:

1. Functions are different from objects; this difference is made evident in many places in the text of the standard.
2. Variable declarations (and definitions) usually can be written one per line, with variable names aligned on several lines; function definitions always have function bodies after them, so there isn't the same benefit of aligning function names in several successive functions.
3. Function definitions tend to use a lot of space after the function name (for parameter names and types), whereas variable declarations usually don't. So line breaking considerations are different for the two cases.

```
> > > > generate_random_derangement( unsigned a[], unsigned n ){  
> > > > unsigned i, j, t, r, r_limit;  
> > > >  
> > > > r_limit = RAND_MAX - RAND_MAX % n;  
> > > >
```

comp.lang.c: Re: derangement: code review request

```
> > > > for( i = 0; i < n; i++ ) a[i] = i;  
> > >  
> > > Ew ew ew! You just proved that whitespace isn't _always_ good (and  
> > > whitespace on the _inside_ or parens rarely is).  
> >  
> > If indeed something has been proved here, a better candidate is that  
> > the spacing style used above is something Richard Bos is unaccustomed  
> > to seeing.  
>  
> Thankfully, yes. It's been quite well proven (for centuries; spacing is  
> not something that is unique to source code) that both too few, but also  
> too much whitespace renders text less legible.  
> You do not write like this, do you?
```

I might, if I thought it would communicate more effectively.

But more to the point, the analogy text makes a weak and silly argument. Weak because the spacing in the prose text example is clearly more homogeneous and more random than the code text example; silly because it presumes that the same kind of reasoning that applies to prose text should apply to code text, whereas obviously that's not so. We don't write code text all left-justified; nor is code text filled the way prose text paragraphs are. The reading mechanisms for the two kinds of text are quite different; we would expect that what style choices communicate effectively would in many cases be different also.