

Re: C Programming

Source: <http://linux.derkeiler.com/Mailing-Lists/Fedora/2004-05/2011.html>

From: Wade Chandler (wchandler_at_redesetgrow.com)

Date: 05/11/04

Date: Mon, 10 May 2004 18:47:52 -0400

To: For users of Fedora Core releases <fedora-list@redhat.com>

fredex wrote:

> *On Sun, May 09, 2004 at 10:48:03PM -0500, Jeff Vian wrote:*

>

>>

>>*fredex wrote:*

>>

>>

>>>*On Sun, May 09, 2004 at 08:34:28AM -0500, Jeff Vian wrote:*

>>>

>>>

>>>

>>>>*fredex wrote:*

>>>>

>>>>

>>>>

>>>>>*On Sat, May 08, 2004 at 10:46:37PM -0400, Wade Chandler wrote:*

>>>>>

>>>>>

>>>>>

>>>>>>*Gene Heskett wrote:*

>>>>>>

>>>>>>

>>>>>>

>>>>>>>*On Saturday 08 May 2004 06:27, Trevor McNamara wrote:*

>>>>>>>

>>>>>>>

>>>>>>>

>>>>>>>>*Hello,*

>>>>>>>>

>>>>>>>>

>>

>><*snip*>

>>

>>>>>*Not to be a negative-sounding force here, but I would urge you to NOT*

>>>>>*ever use any of Herbert Schildt's books.*

>>>>>

Fedora: Re: C Programming

>>>>> *Mr. Schildt is a skillful writer, his writing is clear and lucid. He writes books on C and C++ that very easily and clearly lead you down the path to writing programs that are WRONG. he teaches bad practice in C and C++ in clear and easily understood ways, such that if you follow his advice you will learn his bad habits. one minor example is that he is one of the many people who incorrectly teach that it is permissible to declare main() as being of type void:*

```
>>>>> void main (void)
>>>>> {
>>>>> ...
>>>>> }
```

>>>>> *A little web searching, or spending a little time on groups.google.com reading comp.lang.c will find you plenty of examples of why his books are not a good way to learn C or C++. You can learn, in the same way, of many other books that do teach the language(s) correctly.*

>>>>>
>>>>>
>>>>>
>>>>>
>>>>>
>>>>> *Please explain why main cannot be of type void.*

>>>>>
>>>>>
>>>>>
>>>>>
>>>>> *1. Because the ANSI/ISO standard for the C language requires that main returns int.*

>>>>>
>>>>> *2. And if that isn't good enough for you, if you don't care about standards, look at the practical implications. When you write a function:*

>>>>>
>>>>>
>>>>>
>>>>>
>>>>> *I expected a reasoned discussion, not flaming.*

>
>
> *I didn't (and don't) view this as flaming. I merely state it that way because I've seen discussion on this for years from people who aren't interested in what the standard says (they say "Well, it works for me on my compiler, so why shouldn't I use it???"). So, I gave a practical reason for why it matters.*

>
>
>><snip>
>>
>>>>> *While I agree main is most often typed to int for the purpose of returning a value designating completion with no errors or an error, there is no *requirement* for that.*

>>>>>
>>>>>
>>>>>

Fedora: Re: C Programming

>>>ah, but there is. Again, the language standard requires that main return
>>>an integer.
>>>
>>>
>>>So "why", you ask, "does the standard require main to return int?".
>>>The original C standard was written to "codify existing practice". In many
>>>environments (Unix, MS-DOS, probably other platforms) the return value of
>>>main is (or can be) used by the invoking program as a success or fail
>>>status indicator. If the program doesn't return such a value the calling
>>>program has no way to find out if hte program ran successfully or not.
>>>
>>>
>>
>>>true
>>
>>
>>>True, you CAN write a broken program that doesn't return anything from
>>>main, but it's still going to hand back SOME value to the invoking program
>>>as its status. if you don't provide a value, the code that calls main is
>>>going to return something anyway, most likely some garbage/arbitrary
>>>value it picked up from the place main's return should have been, but
>>>wasn't (since main is expected to return something but didn't).
>>>
>>>Understand, main() is NOT the first code to execute when a C program is
>>>invoked. There's a chunk of "startup code", that is provided by the
>>>compiler vendor or a linkable library (invisible to the user) that
>>>actually is invoked first, before main(). This code may do little in some
>>>environments, or in other environments it may do a great deal of
>>>stuff. But whatever it does, it (among other things) hands main its two
>>>arguments, and receives an integer back from main. Changing the type of
>>>main in your program does not change this code.
>>>
>>>
>>>
>>>
>>>
>>>>If you can provide **definitive documentation** of your statement that his
>>>>teaching is wrong in the requirements for function main() I would like
>>>>to see it.
>>>>
>>>>
>>>
>>>See the language standard document.
>>>
>>>
>>
>>I did look at that, all 550 pages of it.
>>
>
> OK, then, you surely saw the place(s) where it discusses main().
>

Fedora: Re: C Programming

>
>>>*Or go ask this question in the newsgroup comp.lang.c where you'll either*
>>>*be ignored as a troll, or else will get flamed, depending on the mood*
>>>*this week.*
>>>
>>>*Mr. Schildt, for all his skill as a writer (and he is good, no denying*
>>>*that!), is one (of many) people who perpetuate the myth that it's OK*
>>>*to play fast and loose with the language standard. The result is that*
>>>*so many people think that it IS OK to do so. But some day, somewhere in*
>>>*the future, doing one of these things he teaches will nail your program*
>>>*to the wall and you won't have a clue why it went wrong.*
>>>
>>>
>>
>>*This as certainly not the first author of programming books to violate*
>>*the standards in his methods. I first learned C in the days of*
>
>
> *Did I say he was the first/only? No, I did not.*
>
>
>>*Borland's Turbo C compiler, (taught at Drexel University in*
>>*Philadelphia), and the text book used in that class also ignored the*
>>*return type of main(). (yes, I still have the text and verified this)*
>>
>>*Are you planning to be this hostile about ALL books that you don't*
>>*happen to like.?*
>
>
> *Again, it's not intended as hostility. It is intended as a warning to*
> *people who will take a book and learn from it, especially those books*
> *that teach incorrect programming. It's not the reader's fault that the*
> *book teaches them wrongly. However someone who does not already know C*
> *will not be able to discern those areas where the book teaches wrong*
> *practice. There are books around that carefully do not teach bad practice,*
> *many of them are recommended whenever this discussion comes up.*
>
> *Again, the reason why I warn people about the books of Mr. Schildt is*
> *that they too often teach bad practice, book after book. (no I haven't*
> *read them all, but I have read some of them, and have seen plenty of*
> *discussion from those who both know C and have read his books).*
>
>
> *I've said all I want to say on this. Please do not continue to*
> *misinterpret what I'm trying to say (which is intended as a service to*
> *those who might be lead astray by books that teach incorrect practice).*
>
>
>>*Your point would be much better made without hostility, but reasoned*
>>*argument instead. I noted where the standards define main and I am*
>>*satisfied with that. All I originally asked for was the location where*

>>it was defined that way, not a hostile diatribe.
>
>
> And you got a pointer to the standard (I don't happen to have a copy
> of it *HERE* right now to which I can refer so I could have given you
> a more detailed reference). What you also got was a lengthy discussion
> of why it matters, it was *NOT* a diatribe. sorry if you misinterpreted it.
>
> Goodbye.
>
>
>

Basically this thread has went nuts. I followed up on a post and I mentioned the wrong name of a book of Herberts I like, and I still do by the way. It isn't C++ Bible, but "The Complete Reference: C++". One, this book doesn't use void main(void). The only mention of void in main is int main(void) or int main() (perfectly legal and logical). int main() directly from "The C++ Programming Language" Stroustrup and int main(void) and even int main()...in "The C Programming Language" Kernighan/Ritchie.

Anyways, void main(void) is a bad example to use to make a point against an author without giving a book and page number. I have not seen where Schildt says to use or uses void main(void) in any book I've seen. In the complete reference I can turn to the index page 1003 look in M and find a reference to page 153 (main, return value from) (in the complete reference). Now, if I turn to page 153 I'll find (as I should):

"

What Does main() Return?

The main() function returns an integer to the calling process, which is generally the operating system. Returning a value from main() is the equivalent of calling exit() with the same value. If main() does not explicitly return a value, the value passed to the calling process is technically undefined. In practice, most C/C++ compilers automatically return 0, but do not rely on this if portability is a concern.

"

I guess he could have said one should always use return values, which would have been more correct according to many standards, however, I don't think this warrants an example of void main(void) in his name. Maybe you saw some typo or something that passed the editors, but I have never seen void main(void) in the book I'm speaking of nor any others of his...that I have personally seen.

I personally find "The Complete Reference: C++" to be a well written book, and it gets into details that plenty of books I have read do not.

I know plenty of convert C to C++ programmers who use static instead of an empty namespace for hiding from the global scope because they don't know you can...apparently. So instead of typing a few characters they have to type static multiple times.

Fedora: Re: C Programming

Contrast this book to "The C++ Programming Language". I happen to like Stroustrup's book, but it is a little hard to read at times. It isn't because the subject is very complicated either. It is just laid out a little strange (in my opinion), and isn't as nice to read as "The Complete Reference: C++". It just doesn't seem to flow as well as what I'm saying. This in no way means that Stroustrup's book isn't useful or doesn't have a lot of good info. I happen to like them both, but I don't believe Schildts writing is bad nor incorrect either.

Remember that I'm talking about "The Complete Reference: C++" as the example here.

Wade

```
--  
fedora-list mailing list  
fedora-list@redhat.com  
To unsubscribe: http://www.redhat.com/mailman/listinfo/fedora-list
```