

# What makes you a C programming expert? [closed]

Asked 16 years, 3 months ago   Modified 9 years, 8 months ago

Viewed 21k times



26



**Closed.** This question is [opinion-based](#). It is not currently accepting answers.



**Want to improve this question?** Update the question so it can be answered with facts and citations by [editing this post](#).

Closed 6 years ago.

[Improve this question](#)

I attended a job fair yesterday and a developer asked me how I would rank my proficiency in C. I then realized that this is incredibly arbitrary and almost impossible to nail down, so my question is what knowledge makes you an expert in programming C?

Edit: or what would the breakdown be? what makes you good, decent, proficient, etc.

Edit again: I was looking more for like a list of skills or some other constructive measure by which to judge one's own proficiency in C, as that's hard to do.

List so far:

- Experience in large projects
- Mastery of Pointers (and memory management, I'd assume)
- Mastery of a debugger (gdb, ...)
- Mastery of a profiler (gprof,...)
- Mastery of a memory profiler (valgrind, ...)
- Knowledge of the fundamental standards

c

Share

Improve this question

Follow

edited Jul 30, 2010 at 13:09



Bill the Lizard

405k ● 211 ● 572 ● 889

asked Sep 19, 2008 at 16:55



Alex Gartrell

2,544 ● 5 ● 22 ● 23

---

I've had this same question for C++ in interviews, especially for interns and recent graduates. The thing is often that they're trying to guess your character more than your proficiency in programming (in C++). If you give yourself a very high score, it means you're 1) very knowledgeable and very confident; or 2) fairly ignorant and trying to impress just to get the job. I even had one case where they had 2 different tests. If you answered 9 or 10, they'd give you the tougher

one to find out in which of those two categories you really were. – [André Caron](#) Jul 20, 2011 at 14:32

---

I would say knowing how to find which function caused an crash by using a map file. – [Rocky Pulley](#) Mar 30, 2015 at 17:56

---

## 17 Answers

Sorted by:

Highest score (default)



35



I think the trouble with this question is that the answer is kind of meaningless. I see people talking about experience, and that's good, and I see people talking about understanding the intricacies of the language, and that's good. However, if I were hiring someone to work on my C project, and I had a magic 8-ball that would give me an accurate answer to any one (and only one) question, I would never ask it, "Are they a C expert".

Why?

Just because someone is a C expert doesn't mean that they're a good software developer. Experience and language familiarity are good, but I think they are both trumped by that intangible, un-quantifiable property that makes someone a "good software developer". What I'm trying to say is, "What makes you a C programming expert?" is not a useful question, because there are more important questions. If someone is a **Good Programmer**, they will rise to the occasion.

As an example: You can be a C programming expert and be horrible on a team. You can be a C programming

expert and refuse to use version control. You can be a C programming expert without knowing how to actually **DO** anything with C.

The "without" clauses in those sentences are equally important questions: What makes you a good team programmer? What's the best way to use SCM x or y? How do you approach programming a client/server game, or billing application, or web browser, or operating system, or compiler, in C? If a candidate told me "No, I am not a C expert", but gave me great answers to these other questions, I would hire them in a heartbeat over the guy who the magic 8-ball said was a C expert, but doesn't know how to check his code into subversion and hasn't learned a new language in 12 years.

Share Improve this answer

answered Sep 19, 2008 at 17:10

Follow



Max Cantor

8,357 ● 7 ● 47 ● 59

---

Another good point is that C experts may try to program EVERYTHING in C, when it's blatantly unnecessary. I agree with what you're saying, but, all other things being equal, I'd argue knowledge of a technology is a good thing (when used only as necessary). – [Alex Gartrell](#) Sep 19, 2008 at 17:14

---

I didn't mean to suggest that actual skill was unimportant; just that it must be tempered with a sense of pragmatism and perhaps even humility. I do agree with you. – [Max Cantor](#) Sep 19, 2008 at 17:16

---

Also, being an expert developer goes beyond being an expert <your language here> programmer - an expert developer also needs to understand things like architecture, design,

security, etc, things that go far beyond the syntax of a language. – [Avid](#) Sep 21, 2008 at 0:07

---

@Avid I believe that's a software architect's job, and we don't all really stand up to that status. – [kellogs](#) Mar 22, 2012 at 10:43

---

@kellogs sure, building the architecture is the architect's job - but that doesn't mean a good developer does not need to *understand* it. Also, there are plenty of these other aspects - security, performance, UX, etc - that a good developer absolutely *must* be proficient in. – [Avid](#) Mar 22, 2012 at 11:44

---



21



When I interviewed with Google, the interviewer told me to think about it this way. On a scale of 1-10 for C proficiency, to say you're a "10" means you've written papers and/or books or been a speaker in a conference on programming in C. Based on this, very few people are 10s.

FWIW, I have been programming in C for 15 years. I consider myself very proficient. I'd perhaps give myself a solid 8 or 8.5.

Share Improve this answer

Follow

answered Sep 19, 2008 at 17:01



[EmmEff](#)

7,921 ● 2 ● 19 ● 19

---

2 I had a similar experience. I was told to use the scale where 1 = "Can write Hello, world!" and 10 = "Can write a functioning compiler." – [Bill the Lizard](#) Sep 19, 2008 at 17:04

---

I up-voted this for the humility. – [Alex Gartrell](#) Sep 19, 2008 at 17:07

---

12 The problem is that gap between 1 and 10 on this scale is not linear – [RomanM](#) Sep 19, 2008 at 17:12

---

1 By this type of scale people are imply they'd defer to someone with a 9/9.5/10. Herb. Schildt has written books, but I'd bet EmmEff shouldn't defer to him. A functioning (even very strict) compiler is different from knowing the std. or applying it well. I'd say google are too full of themselves. – [James Antill](#) Sep 19, 2008 at 17:14

---

One cannot argue that Google is full of themselves, but I it is an easy way for them to reposition the poseurs on the scale. And yes, I certainly agree the scale is non-linear. – [EmmEff](#) Sep 19, 2008 at 17:21

---



19

To someone less skilled than you, you're an expert.

To someone more skilled than you, you're a newbie.



Share Improve this answer

answered Sep 19, 2008 at 17:00

Follow



[Joseph Styons](#)

58.6k ● 64 ● 167 ● 237





16



Experience is key, knowing the "rules" and syntax of the of the language is of course a must, but it is only a base. Learning the common pitfalls and idioms for doing things right is key.

Knowing what if any resources exist to get help from while your programing, and of course, knowing you're tool chain. I've known many C++ "experts" who had never used a debugger, or a memory tracker. If you ask me, being an expert in something is different from being proficient in something in you knowing all aspects of it.

Share Improve this answer  
Follow

answered Sep 19, 2008 at 17:06



Adam

26.5k ● 23 ● 77 ● 87

---

15 I will add to this, that you are an expert in C when you have made all possible mistakes :P – [freespace](#) Sep 20, 2008 at 5:28

---



9



Everyone is an expert at a job fair

Share Improve this answer  
Follow

answered Sep 19, 2008 at 16:58



Kristian

6,603 ● 6 ● 29 ● 30



You're an expert in c if you can answer all the questions tagged "c" on stackoverflow.com without blinking.

6

Share Improve this answer

answered Sep 19, 2008 at 16:58



Follow



[Ian Terrell](#)

10.9k ● 12 ● 47 ● 67



---

1 I approve this message. Don't forget the blinking.

– [Pascal Paradis](#) Sep 19, 2008 at 16:58

---

... and when it comes to an area of algorithms that you are not really familiar with.. let's say DSP stuff for example? These guys do write most of their code in C, and they do magic not much people understand... – [Nils Pipenbrinck](#) Sep 19, 2008 at 17:43

---

2 then you are not a C expert :P – [Sekhat](#) Feb 12, 2009 at 0:17

---

If you achieve the super human task of not blinking for several minutes while you look up the answer and write the post, does that make you an expert C programmer?

– [André Caron](#) Jul 20, 2011 at 14:34

---



This doesn't directly answer your question (sorry), but it might help you decide how you classify yourself.

6

Instead of just "expert" and "clueless newbie" I prefer the three-level system of expertise used by the medieval guilds:







## Apprentice

- Still needs to RTFM.
- Getting to grips with the tools and techniques of his craft.
- Needs supervision.

## Journeyman

- Has Read The Effin' Manual.
- Competent with all the standard tools and techniques of his craft.
- Can work alone, and can supervise apprentices on routine jobs.

## Master

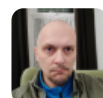
- Could have Written The Effin' Manual.
- Is developing or adopting new tools and techniques.
- Can oversee a major project that might never have been attempted before.

At a job fair? There are no experts: everybody's an expert. :)

Share Improve this answer

answered Sep 19, 2008 at 17:28

Follow



Pitarou

2,239 ● 18 ● 25



4

Some may disagree but I think experience is key to being an expert in any language. I know plenty of people who've past the certification test but couldn't apply their knowledge to anything practical in the real world.



So I think overall being an expert is a product of having enough knowledge on a given subject (C) and then having applied it to enough real world scenarios to make the mistakes that we all do and learn from them.

Share Improve this answer

answered Sep 19, 2008 at 16:59

Follow



Smallinov

1,272 ● 2 ● 9 ● 16



4

The answers to this question do make for some interesting reading - it seems that we can't get good convergence on what defines an expert here. What hope is there going to be in a broader forum like a jobs fair? :-)



But to put my own 2 cents in... I think there's two kinds of C expert.

- There's the expert in the academic sense (as in "could write their own compiler", "has written papers").
- There's the pragmatic expert. I would like to define this as "someone who can write elegant C code that *anyone* can understand".

I would take one of the latter over the former in a heartbeat. If you've got a chunk of code written by an expert that is so brittle that can only be read and understood by another expert then for all intents that code is unmaintainable. It's all very nice that the author of this code remembers the intricacies of type conversions in the middle of expressions, but it's much better if the code has been written so that it's completely unambiguous.

Projects usually have enough technical challenge without adding the need for all team members to have memorized the C'99 standard.

Share Improve this answer

Follow

answered Sep 21, 2008 at 0:00



[Andrew Edgecombe](#)

40.3k ● 3 ● 38 ● 63



You're an expert in C when you can write your own C compiler.

3



Share Improve this answer

Follow

answered Sep 19, 2008 at 17:13



[Steve Baker](#)

4,373 ● 1 ● 22 ● 15



Interview questions like this are always tough. You want to blow your own horn a little, but not sound like a

2

blowhard.



If you have done a lot with C (say, worked on open source projects in C), then I'd respond with that, but not just by pointing to the list of accomplishments on your resume, but by talking about one or more of them and what was particularly interesting or challenging about it (in regards to its use of C).



Share Improve this answer

answered Sep 19, 2008 at 17:03

Follow



nsayer

17k ● 4 ● 36 ● 52



How about having read "Expert C Programming" by Peter van der Linden and remembering everything he covered?

2

Share Improve this answer

answered Sep 19, 2008 at 17:03

Follow



Slapout

3,809 ● 5 ● 42 ● 62



3 I'd say reading ... and not finding anything new in it.  
– [Eli Bendersky](#) Sep 19, 2008 at 17:14

Understanding everything he covered. Not remembering, but understanding. – [Galaxy](#) Jun 14 at 21:34



Lacking a standard test there's really no way to decide what expert level is but here are a few of my litmus tests,

2

everyone's list is different, I'm sure.



Without looking at documentation:



- Know the precedence of the main operators so you don't have to litter your code with parens to avoid getting the wrong order of evaluation
- Be able to write a prototype for a simple function pointer
- Be comfortable with passing a pointer to a pointer
- Understand block, function, module scope

There are more items like this.

On the other hand, I don't think you have to be able to understand or be able to write out Duff's device or figure out obfuscated C contests in your head to consider yourself an expert.

Even if I considered myself an expert (not sure I do) I probably would never claim it in a job interview.

Andrew

[Share](#) [Improve this answer](#)

answered Sep 19, 2008 at 17:09

[Follow](#)



[Andrew Queisser](#)

1,175 ● 10 ● 20

---

as far as precedence of operations is concerned, doesn't omission of unnecessary parens sometimes come at the cost of code clarity? It's a lot easier to make it explicit through parens than to say "Oh, any future reader of this should be

familiar with how it works." – [Alex Gartrell](#) Sep 19, 2008 at 17:12

---

- 3 I consider knowing precedence completely unimportant. I've always liked the rule put forth by Steve Oualline in "Practical C": ----- 1) Multiplication and division come before addition and subtraction. 2) Put parentheses around everything else. ----- – [Michael Burr](#) Sep 19, 2008 at 17:17
- 



Mastery of pointers.

1

Share Improve this answer

answered Sep 19, 2008 at 17:00

Follow



[J Miller](#)

990 ● 1 ● 9 ● 13



- 
- 4 That's not expert knowledge, that's basic competence! – [Jim](#) Sep 19, 2008 at 17:02
- 

- 1 I cannot imagine anybody being able to even call themselves a C programmer (ie. rating of 1 in a scale of 1-10) without having mastery of pointers. – [EmmEff](#) Sep 19, 2008 at 17:02
- 

I mastered pointers long before I started learning C. I wrote assembly language code. – [Ferruccio](#) Sep 19, 2008 at 17:18

---

- 1 Mastery, people. Not competence. – [J Miller](#) Sep 19, 2008 at 17:24
-



1



I would say that for any given language, experience is the key thing. It just takes time to learn a language and learn the APIs and 'idioms' that the language uses. Whether someone is an expert in anything or not is something that should be asked of ones peers. To paraphrase Jeff Foxworthy, "If you answer more questions than you ask, you *might* be an expert."

Share Improve this answer

Follow

answered Sep 19, 2008 at 17:01



[SteveDonie](#)

8,996 ● 3 ● 45 ● 43



I think a fair answer would be understanding all of the intricacies of ISO C.

1



The reality, as any comp.lang.c regular will tell you, is that almost nothing that people need to do can be done in pure ISO C, as you generally need to interact with your environment in a more well defined way. That's where POSIX comes about.



I would not blink at anyone who self-ranked themselves as an "Expert" who had a solid understanding of the language of C, a decent understanding of what ISO C promises, and a working understanding of the POSIX functions.

Share Improve this answer

answered Sep 19, 2008 at 17:02

Follow



Daniel Papasian

16.4k ● 6 ● 31 ● 32



Being able to write papers/books doesn't necessarily make one an expert programmer. It takes plenty of hard work, practical experience and a good understanding of various C libraries.

1



Good luck!



Share Improve this answer

edited Jul 20, 2011 at 14:14

Follow

answered Nov 16, 2009 at 23:08





SteJava

495 ● 4 ● 11



**Highly active question.** Earn 10 reputation (not counting the [association bonus](#)) in order to answer this question. The reputation requirement helps protect this question from spam and non-answer activity.