

# What is the single most influential book every programmer should read? [closed]

Asked 16 years, 4 months ago   Modified 12 years, 10 months ago

Viewed 1.1m times

1434

votes



As it currently stands, this question is not a good fit for our Q&A format. We expect answers to be supported by facts, references, or expertise, but this question will likely solicit debate, arguments, polling, or extended discussion. If you feel that this question can be improved and possibly reopened, [visit the help center](#) for guidance.

Closed 12 years ago.



**Locked.** This question and its answers are [locked](#) because the question is off-topic but has historical significance. It is not currently accepting new answers or interactions.

If you could go back in time and tell yourself to read a specific book at the beginning of your career as a developer, which book would it be?

I expect this list to be varied and to cover a wide range of things.

**To search:** Use the search box in the upper-right corner. To search the answers of the current question, use

`inquestion:this` . For example:

```
inquestion:this "Code Complete"
```

resources

Share

edited Sep 26, 2011 at 15:39

community wiki

18 revs, 13 users 34%

Robert Harvey

---

8 Browsing this thread make me realize how ugly most programming related books are. Very good thread though!

– [Carl Bergquist](#) Aug 5, 2009 at 12:09 

---

23 Interesting this is, while the title reads "What is the single most influential book every programmer should read?", there are quite a few books suggested that deal with language specific topics. By definition, and by question as it was put, the books suggested here should deal with language agnostic topics, which proves most programmers have yet to learn how to read. – [Rook](#) Oct 2, 2009 at 19:54



19 If I could go back in time and tell myself to read something, it better be a newspaper or sports fact book that I carried with me. Anything else is a waste of good time travel. :-)  
– [jmucchiello](#) Nov 8, 2009 at 9:38

---

32 You know, if I wasn't worried about getting down voted a WHOLE lot I would trollishly go and suggest Twilight. "Its ALSO about people who are pale and avoid the sun!"  
– [Jacob Bellamy](#) Feb 12, 2010 at 0:20

---

3 Can someone cleanup answers by deleting repeated entries on books? Most of them area already down voted.  
– [rao](#) Mar 16, 2010 at 5:44

---

Comments disabled on deleted / locked posts / reviews |

214 Answers

Sorted by:

Highest score (default)



Prev

1

2

3

4

5

...

8

Next

7

votes



Kernighan & Plauger's [Elements of Programming Style](#). It illustrates the difference between gimmicky-clever and elegant-clever.

Share

edited Oct 2, 2009 at 19:51

community wiki

2 revs, 2 users 71%

[Peter Mortensen](#)

7

[The TCP/IP Guide](#), by Charles M. Kozierok

votes



Although it is described as an 'encyclopedic reference', it is *incredibly* readable as a narrative.



This author provides a very , very, very well written, comprehensive, introduction to networking and the infrastructure that underlies the web. Something all programmers ought to know.

For me it is the natural follow-on from Charles Petzold's '[Code](#)'. If "Code" explains to the layman how computers work, 'The TCP/IP Guide' explains how they connect together.

If you gave a 12 year old geek a copy 'Code' and a copy of 'The TCP/IP Guide' - they'd be building the next Google by the age of 17.

In other words, if I could go back in time and tell myself to read a specific book at the beginning of my career as a developer, this (plus *Code*) is up there in the top of my list.

Share

edited May 23, 2017 at 11:54

community wiki  
5 revs, 3 users 91%  
JW.

---

6

votes



I've been around a while, so most books that I have found influential don't necessarily apply today. I do believe it is universally important to understand the platform that you



are developing for (both hardware and OS). I also think it's important to learn from other peoples mistakes. So two books I would recommend are:

[Computing Calamities](#) and [In Search of Stupidity: Over Twenty Years of High Tech Marketing Disasters](#)

Share

answered [Aug 11, 2008 at 23:52](#)

community wiki  
[bruceatk](#)

---

3 Give me a break. Why in the world would you mark this down because I mentioned two books? – [bruceatk](#) Apr 6, 2010 at 17:03

---

6

votes



[The Pragmatic Programmer: From Journeyman to Master](#) without a doubt. The advice in it is so well presented, and simple, that it comes across as if it was 'The Common Sense Programmer'. Love it.

Share

edited [Aug 21, 2008 at 20:01](#)

community wiki  
[2 revs](#)  
[Nate Smith](#)

---

## 6 [Mastering Regular Expressions](#)

votes



Share

answered [Aug 21, 2008 at 20:02](#)



community wiki  
[Kibbee](#)

---

6

votes



In no particular order except how they're arranged on my bookshelf:



- *The Pragmatic Programmer*
- *Refactoring* by Fowler
- *Working Effectively with Legacy Code* by Feathers. This is practically a companion volume to *Refactoring*.
- *UML Distilled* by Fowler. Among its other virtues is brevity.
- *Debugging the Development Process* by Steve Maguire
- *Design Patterns* (aka "Gang of Four") by Gamma et al

Share

answered [Aug 21, 2008 at 21:19](#)

community wiki  
[Eric Pohl](#)

---

6

votes



I have a few good books that strongly influenced me that I've not seen on this list so far:

[\*The Psychology of Everyday Things\*](#) by Donald Norman.

The general principles of design for other people. This may seem to be mostly good for UI but if you think about it, it has applications almost anywhere there is an interface that someone besides the original developer has to work with; e. g. an API and designing the interface in such a way that other developers form the correct mental model and get appropriate feedback from the API itself.

[\*The Art of Software Testing\*](#) by Glen Myers. A good, general introduction to testing software; good for programmers to read to help them think like a tester i. e. think of what may go wrong and prepare for it.

By the way, I realize the question was the "Single Most Influential Book" but the discussion seems to have changed to listing good books for developers to read so I hope I can be forgiven for listing two good books rather than just one.

Share

answered [Aug 28, 2008 at 19:13](#)

community wiki

[Onorio Catenacci](#)

---

6

votes

Do users ever touch your code? If you're not doing solely back-end work, I recommend [\*About Face: The Essentials\*](#)



[of User Interface Design](#) — now in its third edition (linked). I used to think my users were stupid because they didn't "get" my interfaces. I was, of course, wrong. About Face turned me around.



Share

answered [Sep 11, 2008 at 3:15](#)

community wiki  
[Garth Kidd](#)

---

-1: Good book, but Duplicate, see earlier – [Ruben Bartelink](#) Jul 15, 2010 at 23:11

---

-1: dup, and Cooper's "Inmates" is a better start for the subject – [user23987](#) Sep 26, 2010 at 19:37

---

6

[Rapid Development](#) by McConnell

votes



Share

answered [Sep 16, 2008 at 14:55](#)



community wiki  
[MattH](#)

---


1 -1 Good book, but dup of [stackoverflow.com/questions/1711/...](http://stackoverflow.com/questions/1711/...) (even if it rambles, it has critical mass) – [Ruben Bartelink](#) Jul 15, 2010 at 23:18

---



+1 At least this one has a link. – [Isabelle Wedin](#) Feb 2, 2011 at 23:15

---



It may be a dup, but the other answer lists two books when the question is "single most". That's cheating. In addition, the other answer is a dup itself. Finally, mine has a link. :-) – [MattH](#) Feb 16, 2011 at 17:48 

---



- 6  
votes
-  The language is easy and also the subject matter is
-  interesting.

Share

answered [Nov 8, 2009 at 8:26](#)

community wiki  
[Steven Bose](#)

---

- 6  
votes
-  Share
- 

[edited Jan 30, 2010 at 0:26](#)

community wiki  
[2 revs, 2 users 67%](#)  
[Paul Tomblin](#)

---

- 6  
votes
- [alt text http://ecx.images-  
amazon.com/images/I/51fhwR6eb3L. \\_BO2,204,203,200\\_PI](#)



[sitb-sticker-arrow-](#)



[click,TopRight,35,-76\\_AA300\\_SH20\\_OU01\\_.jpg](#)

alt text [\[amazon.com/images/I/51PDNR3C40L\\\_SL500\\\_AA300\\\_.jpg\]\(http://amazon.com/images/I/51PDNR3C40L\_SL500\_AA300\_.jpg\)](http://ecx.images-</a></p></div><div data-bbox=)

Share

answered Jul 28, 2010 at 16:10

community wiki

[Display Name](#)

6

votes



As I started out developing in Java (and am still doing so to this very day) I'd have to recommend the outstanding work in the field: [Mr Bunny's Big Cup o' Java](#).



From the author's blurb:

There is simply no better way to learn Java than to have the pineal gland of an expert Java programmer surgically implanted in your brain. Sadly, most HMOs refuse to pay for this career saving procedure, deeming Java to be too experimental. At last there is an alternative treatment for those of us who cannot wait for sweeping health care reforms.

Mr. Bunny's Big Cup O' Java is recommended by n out of ten doctors, where n is any integer you wish to make up to impress an astoundingly gullible

public. The book begins with an overview of the book, and quickly expands into the book itself. Just look at the topics covered:

- Java

In short, MBBCOJ will teach you all you need to know for a successful career in today's rabbit development environments.

The insight into pixels alone would have cut years off my software developing life.

Share

edited Jun 20, 2020 at 9:12

community wiki  
2 revs, 2 users 97%  
Adrian

---

## 5 [Refactoring](#)

votes



Share

answered Aug 21, 2008 at 20:08



community wiki  
Mark Biek

---

5

votes



There are a lot of votes for Steve McConnell's Code Complete, but what about his [Software Project Survival Guide](#) book? I think they're both required reading but for different reasons.

Share

answered [Sep 1, 2008 at 8:14](#)

community wiki  
[christefano](#)

---

Can you add a pic - right now it scans as if its yet another parroting of the CC recommends that this entire thread seems to be about - [Ruben Bartelink](#) Mar 27, 2010 at 13:39

---

5

votes



Software Tools by by Brian W. Kernighan and P. J. Plauger

It had a profound influence on how I write software.

Share

answered [Sep 15, 2008 at 18:25](#)

community wiki  
[Bryan Oakley](#)

5

votes



[Facts and Fallacies of Software Engineering by Robert L. Glass](#) <http://www.codinghorror.com/blog/images/facts-and-fallacies-of-software-engineering.jpg>



[Facts and Fallacies of Software Engineering by Robert L. Glass](#) is a really excellent book. I had been a professional hacker for almost 10 years before I read it, and a I still learned a ton of stuff.

Share

answered [Oct 8, 2008 at 16:38](#)

community wiki  
[afternoon](#)

5

votes



Not the most influential, but worth a look is [Youth](#) by J.M.Coetzee.

The narrator of Youth, a student in the South Africa of the 1950s, has long been plotting an escape from his native country: from the stifling love of his mother, from a father whose failures haunt him, and from what he is sure is impending revolution. Studying mathematics, reading poetry, saving money, he tries to ensure that when he arrives in the real world, wherever that may be, he will be prepared to experience life to its full intensity, and transform it into art. Arriving at last in London, however, he finds neither poetry nor romance. Instead he succumbs to the monotony of life as a computer programmer, from which random, loveless affairs offer no relief. Devoid of inspiration, he stops writing. An awkward colonial, a constitutional outsider, he begins a dark pilgrimage in which he is continually tested and continually found wanting.

[youth cover](#)

<http://img440.imageshack.us/img440/5140/youthgd4.jpg>

Share

answered Jan 22, 2009 at 22:41

community wiki  
[Pete Kirkham](#)

---

## 5 [Perfect Software: And Other Illusions about Testing](#)

votes



[TITLE Cover http://ecx.images-  
amazon.com/images/I/51j3BSRspAL.\\_SL500\\_AA240\\_.jpg](#)



**Perfect Software: And Other Illusions about Testing** *by*  
*Gerald M. Weinberg*

ISBN-10: 0932633692

ISBN-13: 978-0932633699

Share

answered Aug 18, 2009 at 10:20

community wiki  
[Ryan Fernandes](#)

---

## 5 [Design Concepts in Programming Languages](#) by FA Turbak produces detailed implementations of many programming

votes





concepts and is very useful for understanding what's going on underneath the hood.

Share

answered [May 26, 2010 at 12:06](#)

community wiki  
[Dave Clarke](#)

---

5 [The Back of the Napkin](#), by [Dan Roam](#).

votes



[The Back of the Napkin](#)



<http://www.coverbrowser.com/image/bestsellers-2008/302-7.jpg>

A great book about visual thinking techniques. There is also an [expanded](#) edition now. I can't speak to that version, as I do not own it; yet.

Share

edited [Sep 5, 2010 at 18:22](#)

community wiki  
2 revs  
[George Marian](#)

---

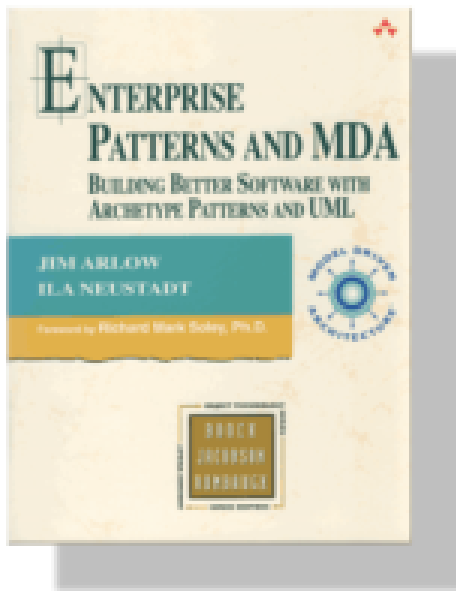
5 [Enterprise Patterns and MDA: Building Better Software with Archetype Patterns and UML](#)

votes





An excellent read for those looking to leverage ORM and UML



Share

answered [Nov 1, 2010 at 2:40](#)

community wiki  
[user219760](#)

5

votes



This one isnt really a book for the beginning programmer, but if you're looking for SOA design books, then [SOA in Practice: The Art of Distributed System Design](#) is for you.

Share

edited [Sep 2, 2011 at 6:45](#)


community wiki  
[2 revs, 2 users 82%](#)  
[Kilhoffer](#)



---

2 Too narrow to be the 1 book every programmer should read  
– user23987 Sep 26, 2010 at 19:31

---

4  
votes  


Code Complete is the number one choice, but I'd also cite Gang of Four's Design Patterns and Craig Larman's Applying UML and Patterns.



The Timeless Way of Building, by Christopher Alexander, is another great one. Even though it's about architecture, it's included in the bibliography of many great programming books I have already read.

Another one, from which I'm learning lots of new things, is Data Access Patterns, by Clifton Nock.

Share


answered [Aug 5, 2008 at 17:28](#)

community wiki  
[Mario Marinato](#)

---

more than one book per anser, and many if not all of these in the first 2 pages. – [xenoterracide](#) Jul 24, 2011 at 0:16

---

4  
votes  


I recently read [Dreaming in Code](#) and found it to be an interesting read. Perhaps more so since the day I started reading it Chandler 1.0 was released. Reading about the growing pains and mistakes of a project team of talented



people trying to "change the world" gives you a lot to learn from. Also Scott brings up a lot of programmer lore and wisdom in between that's just an entertaining read.

[Beautiful Code](#) had one or two things that made me think differently, particularly the chapter on top down operator precedence.

Share

answered [Sep 1, 2008 at 16:22](#)

community wiki  
[Patrick Hogan](#)

---

## 4 Programming Perl (O'Reilly)

votes



Share

answered [Sep 16, 2008 at 14:03](#)



community wiki  
[rikxik](#)

---

## 4 [Lean Software Development](#) by Mary and Tom

votes



Poppendieck is definitely one for every developers bookshelf



Share

answered [Sep 16, 2008 at 14:20](#)

## 4 Object-Oriented Software Construction by Bertrand Meyer

votes



Share

answered [Sep 16, 2008 at 16:33](#)



community wiki  
[user12933](#)

---

I found this book very useful in my early days - it introduced me to the art and science of of large scale framework design.  
Whatever happened to Eiffel? – [Andrew Matthews](#) Sep 8, 2010 at 4:07

---

4 Nobody seems to have mentioned Stroustup's [The C++ Programming Language](#) which is a great book that every C++ programmer should read.

votes



I also think that [Extreme Programming Explained: Embrace Change](#) should be read by every programmer and manager. Many of the ideas in the book are common knowledge now but the book gives an intelligent and inspiring account of the pursuit of quality in software engineering.

I would second the recommendations for Knuth and Gang of Four which are classics.

Share

answered Nov 4, 2008 at 22:40

community wiki  
[nlativy](#)

---

4 Three books come to mind for me.

votes



- The Art of Unix Programming by Eric S. Raymond.
- The Wizardry Compiled by Rick Cook.
- The Art of Computer Programming by Donald Knuth.

I also love the writing of [Paul Graham](#).

Share

answered Apr 27, 2009 at 13:02

community wiki  
[Levi Campbell](#)

---

Prev

1

2

3

4

5

...

8

Next