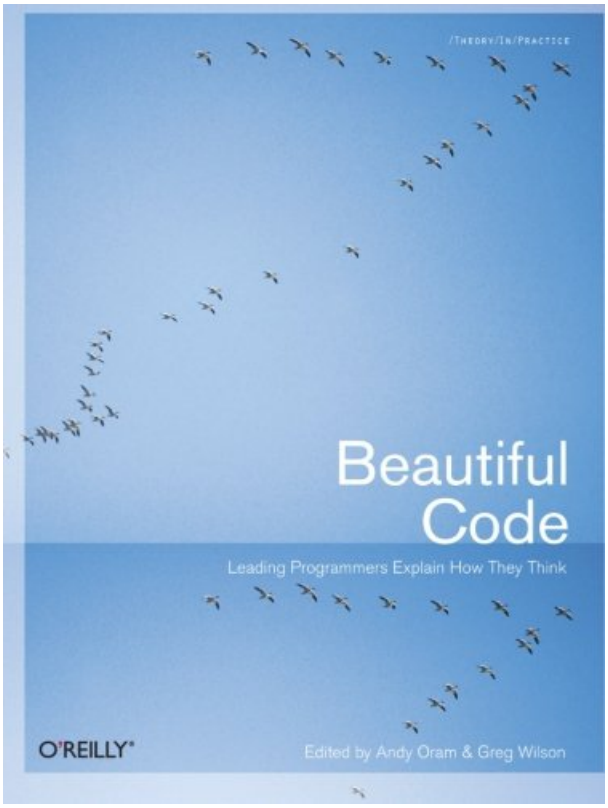
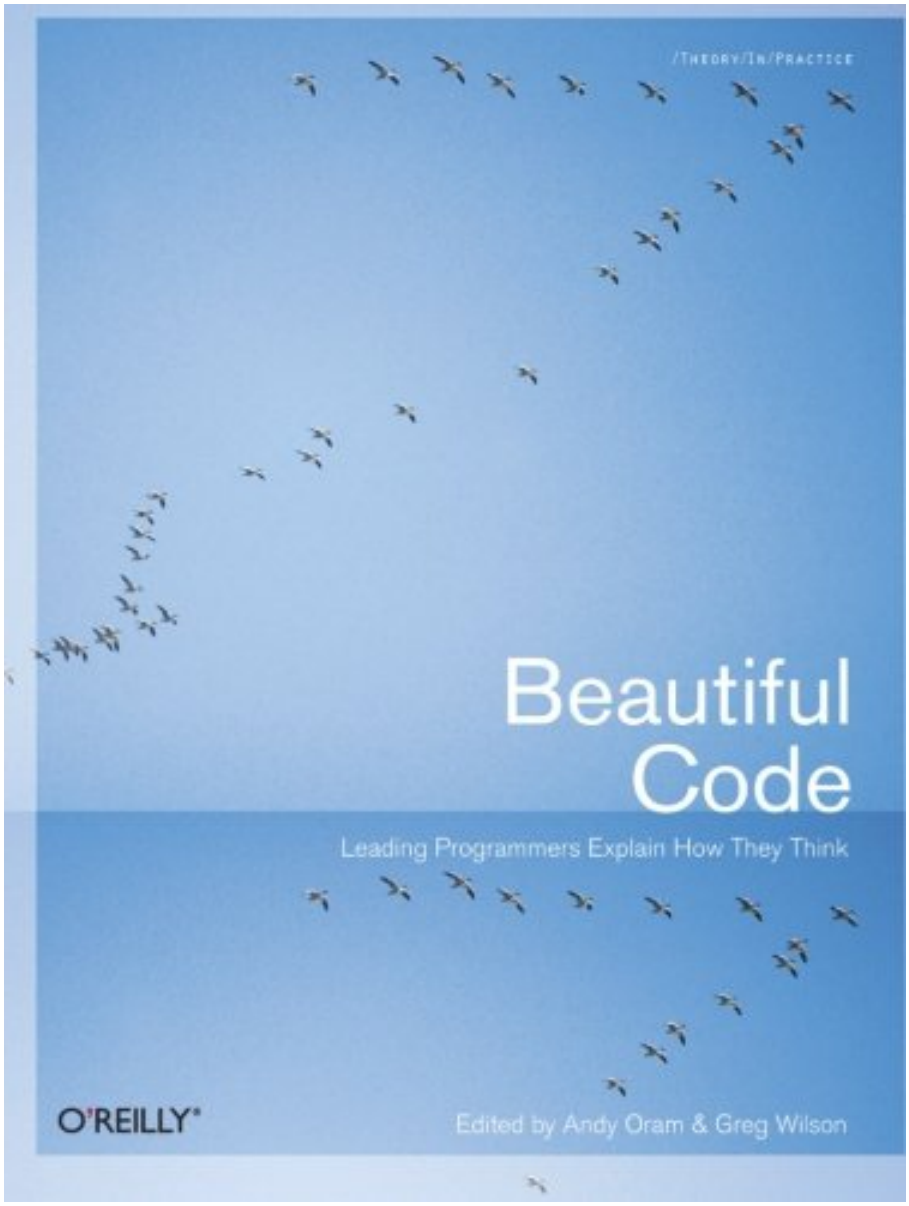
BEAUTIFUL CODE: LEADING PROGRAMMERS EXPLAIN HOW THEY THINK (THEORY IN PRACTICE (O'REILLY)) FROM ORAM, ANDY/ WILSON, GREG





BEAUTIFUL CODE: LEADING PROGRAMMERS EXPLAIN HOW THEY THINK (THEORY IN PRACTICE (O'REILLY)) FROM ORAM, ANDY/ WILSON, GREG PDF

For everyone, if you intend to begin accompanying others to read a book, this *Beautiful Code: Leading Programmers Explain How They Think (Theory In Practice (O'Reilly)) From Oram, Andy/ Wilson, Greg* is much suggested. As well as you should get the book Beautiful Code: Leading Programmers Explain How They Think (Theory In Practice (O'Reilly)) From Oram, Andy/ Wilson, Greg below, in the web link download that we supply. Why should be here? If you desire various other type of books, you will always discover them and also Beautiful Code: Leading Programmers Explain How They Think (Theory In Practice (O'Reilly)) From Oram, Andy/ Wilson, Greg Economics, national politics, social, sciences, religious beliefs, Fictions, as well as more publications are supplied. These available books are in the soft documents.

About the Author

Andy Oram is an editor at O'Reilly Media, a highly respected book publisher and technology information provider. An employee of the company since 1992, Andy currently specializes in free software and open source technologies. His work for O'Reilly includes the first books ever published commercially in the United States on Linux, and the 2001 title Peer-to-Peer. His modest programming and system administration skills are mostly self-taught.

Greg Wilson holds a Ph.D. in Computer Science from the University of Edinburgh, and has worked on high-performance scientific computing, data visualization, and computer security. He is the author of Data Crunching and Practical Parallel Programming (MIT Press, 1995), and is a contributing editor at Doctor Dobb's Journal, and an adjunct professor in Computer Science at the University of Toronto.

BEAUTIFUL CODE: LEADING PROGRAMMERS EXPLAIN HOW THEY THINK (THEORY IN PRACTICE (O'REILLY)) FROM ORAM, ANDY/ WILSON, GREG PDF

Pointer in picking the very best book **Beautiful Code: Leading Programmers Explain How They Think (Theory In Practice (O'Reilly)) From Oram, Andy/ Wilson, Greg** to read this day can be gotten by reading this web page. You could find the very best book Beautiful Code: Leading Programmers Explain How They Think (Theory In Practice (O'Reilly)) From Oram, Andy/ Wilson, Greg that is marketed in this globe. Not only had actually guides released from this country, but additionally the other countries. As well as now, we intend you to read Beautiful Code: Leading Programmers Explain How They Think (Theory In Practice (O'Reilly)) From Oram, Andy/ Wilson, Greg as one of the reading materials. This is only one of the best publications to accumulate in this website. Check out the page and also browse the books Beautiful Code: Leading Programmers Explain How They Think (Theory In Practice (O'Reilly)) From Oram, Andy/ Wilson, Greg You could discover bunches of titles of guides provided.

How can? Do you assume that you do not need adequate time to opt for purchasing e-book Beautiful Code: Leading Programmers Explain How They Think (Theory In Practice (O'Reilly)) From Oram, Andy/ Wilson, Greg Don't bother! Just rest on your seat. Open your gizmo or computer system and be on the internet. You can open or go to the link download that we provided to obtain this *Beautiful Code: Leading Programmers Explain How They Think (Theory In Practice (O'Reilly)) From Oram, Andy/ Wilson, Greg* By this means, you can obtain the on-line e-book Beautiful Code: Leading Programmers Explain How They Think (Theory In Practice (O'Reilly)) From Oram, Andy/ Wilson, Greg Reviewing the book Beautiful Code: Leading Programmers Explain How They Think (Theory In Practice (O'Reilly)) From Oram, Andy/ Wilson, Greg by on-line could be really done quickly by saving it in your computer system and also device. So, you could proceed every time you have spare time.

Reading the e-book Beautiful Code: Leading Programmers Explain How They Think (Theory In Practice (O'Reilly)) From Oram, Andy/ Wilson, Greg by online can be likewise done easily every where you are. It appears that waiting the bus on the shelter, hesitating the checklist for line up, or various other locations feasible. This Beautiful Code: Leading Programmers Explain How They Think (Theory In Practice (O'Reilly)) From Oram, Andy/ Wilson, Greg could accompany you in that time. It will not make you feel weary. Besides, through this will likewise enhance your life high quality.

BEAUTIFUL CODE: LEADING PROGRAMMERS EXPLAIN HOW THEY THINK (THEORY IN PRACTICE (O'REILLY)) FROM ORAM, ANDY/ WILSON, GREG PDF

How do the experts solve difficult problems in software development? In this unique and insightful book, leading computer scientists offer case studies that reveal how they found unusual, carefully designed solutions to high-profile projects. You will be able to look over the shoulder of major coding and design experts to see problems through their eyes. This is not simply another design patterns book, or another software engineering treatise on the right and wrong way to do things. The authors think aloud as they work through their project's architecture, the tradeoffs made in its construction, and when it was important to break rules. Beautiful Code is an opportunity for master coders to tell their story. All author royalties will be donated to Amnesty International. tion.

- Sales Rank: #79837 in Books
- Brand: Oram, Andy/ Wilson, Greg
- Published on: 2007-07-06
- Released on: 2007-07-06
- Original language: English
- Number of items: 1
- Dimensions: 9.19" h x 1.41" w x 7.00" l, 2.16 pounds
- Binding: Paperback
- 620 pages

Features

- Used Book in Good Condition

About the Author

Andy Oram is an editor at O'Reilly Media, a highly respected book publisher and technology information provider. An employee of the company since 1992, Andy currently specializes in free software and open source technologies. His work for O'Reilly includes the first books ever published commercially in the United States on Linux, and the 2001 title Peer-to-Peer. His modest programming and system administration skills are mostly self-taught.

Greg Wilson holds a Ph.D. in Computer Science from the University of Edinburgh, and has worked on high-performance scientific computing, data visualization, and computer security. He is the author of Data Crunching and Practical Parallel Programming (MIT Press, 1995), and is a contributing editor at Doctor Dobb's Journal, and an adjunct professor in Computer Science at the University of Toronto.

Most helpful customer reviews

95 of 95 people found the following review helpful.

great idea, mediocre execution
By Eli Bendersky
This book came to being from a very good idea. The editors decided to go around and ask renown programmers and designers about snippets of code, software architecture, design or anything related they found beautiful and see as an example of good design.

Indeed, the idea is terrific. After all, besides books describing specific technologies we read on a per-need basis, what books do programmers have to read for inspiration ? Consider artists and architects, for example. They have peer art and work to study and be inspired by. Sure, code reading is highly recommended, but wouldn't it be great if someone had already collected all the good bits ? Wouldn't it be sweet for Brian Kernighan and Yuhikiro Matsumoto to tell you what they've found beautiful ?

Unfortunately, this books doesn't fulfill the high expectations I had from it. It's not bad, no, but it isn't as good as I hoped it to be. There are two main reasons for this:

1. Many of the authors forgot that they're writing for a paper book, and not an online article / blog entry. When reading a paper book, you can't just click on links to find out more information. Therefore, I'd expect many chapters to be more complete. The authors could have spent a few extra lines to explain a concept instead of referencing it to some online resource or (worse) a paid-subscription-access paper at ACM. This is a paper book - I want to read it on the bus to work. Had I wanted to read an online article jumping around links, I would just do that.
2. A few of the chapters in the book are just way too specific. How many people would understand a chapter about LINPAK - a Fortran library for linear algebra manipulation, especially when the author is very parsimonious in explaining the concepts and sends you to linear algebra tomes instead (see complaint #1).

In general, I think that to better execute the idea of such a book, a panel of experts has to be assembled and scrutinize each and every article. I would be much happier to read a book of 10 great articles than a book of 33, of which 10 are great. Who said that each and every programming book should be more than 600 pages long ?

However, I want to finish on a positive note, since as I stated in the beginning, the book is not bad. Here's a list of articles I found really good and interesting. I guess that just for them it was worth to read:

1. Chapter 1, A Regular Expression Matcher, by Brian Kernighan
2. Chapter 2, Subversion's Delta Editor: Interface as Ontology, by Karl Fogel
3. Chapter 3, The Most Beautiful Code I Never Wrote, by Jon Bentley
4. Chapter 8, On-the-Fly Code Generation for Image Processing, by Charles Petzold
5. Chapter 10, The Quest for an Accelerated Population Count, by Henry S. Warren, Jr.
6. Chapter 16, The Linux Kernel Driver Model: The Benefits of Working Together, by Greg Kroah-Hartman
7. Chapter 18, Python's Dictionary Implementation: Being All Things to All People, by Andrew Kuchling
8. Chapter 23, Distributed Programming with MapReduce, by Jeff Dean and Sanjay Ghemawat
9. Chapter 28, Beautiful Debugging, by Andreas Zeller
10. Chapter 33, Writing Programs for "The Book," by Brian Hayes

476 of 515 people found the following review helpful.
It's beautiful, see ? SEE ???
By Dmitry Dvoinikov
The idea of this book is that thirty software developers and/or researchers (respectable ones, no doubt there), had to find the most beautiful piece of code and present its study. Each of them then writes a chapter and

there you have it - a volume of "beautiful code" ! Simple as that.

If there was somebody to fully support the idea of such book, it would be me - I believe that the software industry already spent too much time and effort neglecting the art-and-craft in programming, pretending that it all can be reduced to hard math. Didn't work so far, did it ? Then I very welcome books like this one. But not exactly the one.

Let me put it this way - I couldn't say anything good about this book except that I adore the concept and found may be ten of thirty three chapters interesting (not necessarily beautiful). Beauty is in the eye of the beholder they say, but this lame excuse is the last good thing I could say for this book.

It was supposed to be pedagogical. Did not happen. Rather than making it timeless reference for the readers, the book made a tribune for the authors to talk about, uhm, just about anything. We know how programmers love to talk about what they do, and it's ok. But we also know that they often mumble instead of talking and it's very difficult for us to understand one another, no matter friendly or hostile. This is not to mention that there are no commonality in topics or style or language (programming or English) or anything. The editor had simply glued it together.

Not so bad you say, a good assortment is fine you say ? Let me tell you more, and it's all downhill.

It's as though you expected an album of paintings but instead got a book of random excerpts from chemical specifications for producing paints.

Exemplary conventional antimicrobial, antimildew, or antialgae agent includes 3-iodo-2-propynyl butylcarbamate, diiodomethyl-p-tolylsulfone, 1,2-benzoisothiazolin-3-one, 2-methylthio-4-tert-butylamino-6-cyclopropylamino-s-triazine, 2-(4-thiocyanomethylthio) benzothiazole and the mixtures thereof.

See how beautiful it is that can be painted with that ?

If you ask me, a book like this ought to have structure. Remember the classic one by Gamma et al - they also presented abstract things from different areas or levels, but they kept the information stylistically uniform and structured against a clear taxonomy. Not the case here.

Each chapter is about different matter, presented in a different way. One author presents a performance hack in which he compiles code on the fly. The chapter will then contain several pages of dynamic assembly. The other will show an interesting approach to syntax parsing. This one will have 50 short snippets of something JavaScript-like. Yet another will tell you how to automate debugging by automatically mutating the application. This one won't have code at all. Yet another will show a slick algorithm for counting bits in a word. This one will have a lot of bitwise arithmetic.

And I just loved the one that has NASA in it's title. There - "A Highly Reliable Enterprise System For NASA's Mars Rover Mission". Wow ! How promising ! Want to know what it says ? It says - "In NASA they love their software reliable, even a web-based file server, and so we present you a web-based file server built with JavaBeans in three-tier architecture". Ahem, Mars Rover anyone ?

Don't get me wrong, some of the chapters are reasonably interesting. Interesting ! Not beautiful !

With a little exception, the authors don't even mention the word "beautiful" in their texts. They allure with "There, we have this system, it works like this..." . What exactly the author finds beatiful about it and why -

remains secret.

The most impressive standout was the chapter written by Yukihiro Matsumoto, the creator of Ruby. Three pages in which he simply speaks about what he believes a beautiful code is. He explains to you his understanding of a beautiful code. This is what the book is all about !

Instead, many chapters just demonstrate a few pages (!) of code and conclude - it is beautiful, see !

Many times I wasn't unable to grasp the problem - what was it that required that so called beauty to emerge ? I couldn't see the whole picture, but the authors sort of presume I do and so my possible appreciation of beauty requires deep understanding. What if I show you a magnified fragment of Mona Lisa's background, some 3x3 blackish pixels ? No doubt, Leonardo had to paint them too. But what was that beauty again ?

Only a few authors were wise enough to use a pseudocode. Something that anyone can read, no matter from which camp. Otherwise it's just weird when the authors present their beatiful code in Ruby or Perl or LISP. Look, I didn't touch Ruby yet, I hate Perl and I can't imagine using LISP in practice. Nevertheless the authors repeatedly say something like "It's easy, I'll show you, this bracket does this and that character does something else. Now you see how beautiful it is ?". They literally show you a piece of poetry in foreign language and ask you to appreciate it.

A classical example of awful poetry in Russian is (transliterated)

Ya poet, zovus' Neznajka,
ot menya vam balalajka.

Can you tell whether it's good or bad and why ? What if I told you it's beatiful ? Would you believe ? Does it appeal to your sense of beauty ? Same thing about this entire book.

Awful implementation of an idea that I fully adore. In fact, implementations like this undermine the idea, that's why I rate this book so low and put it away with disgust.

107 of 117 people found the following review helpful.
Beautiful Code
By Joshua Benuck
I am always looking to for new ways to look at programming problems. I love studying new programming languages in order to bend my mind in new, uncomfortable ways. Both of these are reasons I enjoyed Beautiful Code.

Beautiful Code is a collection of essays from some well known software engineers. That said, I didn't immediately recognize many of their names (this is probably an indication of my lack of exposure in their fields of expertise). If you are like me, there is an alphabetical list of short biographical entries in the back of the book you can use to acquaint yourself with who wrote each chapter.

There are chapters from people in the Perl, Python, Ruby, Google, Scheme, and Haskell communities (among others).

I especially enjoyed reading about Google's MapReduce algorithm, Haskell's Software Transactional Memory, and Scheme's syntax-case macro system. These are subjects I have previously tried to tackle, but the explanations written in this book have helped me approach understanding far better than the academic

papers on these subjects I have tried to read.

You'll have to put forth effort to follow the explanations in the chapters as the authors walk you through how they tackle a given problem. This leads eventually to the solution, but may involve many twists and turns along the way. These twists and turns show how the authors think and grants us as the readers insight into how they approach the problems at hand. It's the journey to the desination that sometimes matters more than the destination.

For example, I've long wondered abut the difference between hygenic and non-hygenic macros. Various descriptions on the web have given me some clue, but chapter 25 shows examples and explains the problem very clearly. It then goes about discussing various solutions that have been devised over the years before going into the details of the current solution that is in use today. I've seen the end result before, but knowing what motivated the solution gives me a much greater appreciation for and understanding of it.

The effort required for some chapters may be over your head as they are for me, but those are the chapters where I find the rewards to be the greatest as they force me to look at things in new ways. Once I do achieve understanding I'm able to apply the new found ways of thinking about problems to the situations I face at work and elsewhere which has led to unique and compelling solutions that I would not have thought of before.

I've long been on the search for beauty in the code I write. I have found that as I read and take the time to understand what others see as beautiful, even when I do not see beauty in it at first, I gain greater insights into my craft. I am glad that O'Reilly has taken the time to solicit responses from the authors in this book as it has given us a wealth of experience and expertise that we all can benefit from as we seek to gain greater insights into the various facets of beauty and elegance in code.

See all 53 customer reviews...

BEAUTIFUL CODE: LEADING PROGRAMMERS EXPLAIN HOW THEY THINK (THEORY IN PRACTICE (O'REILLY)) FROM ORAM, ANDY/ WILSON, GREG PDF

So, just be below, find the book Beautiful Code: Leading Programmers Explain How They Think (Theory In Practice (O'Reilly)) From Oram, Andy/ Wilson, Greg now and also check out that quickly. Be the very first to read this book Beautiful Code: Leading Programmers Explain How They Think (Theory In Practice (O'Reilly)) From Oram, Andy/ Wilson, Greg by downloading in the web link. We have a few other e-books to check out in this site. So, you can locate them also easily. Well, now we have done to offer you the ideal e-book to check out today, this Beautiful Code: Leading Programmers Explain How They Think (Theory In Practice (O'Reilly)) From Oram, Andy/ Wilson, Greg is actually appropriate for you. Never ever dismiss that you require this book Beautiful Code: Leading Programmers Explain How They Think (Theory In Practice (O'Reilly)) From Oram, Andy/ Wilson, Greg to make better life. On the internet e-book **Beautiful Code: Leading Programmers Explain How They Think (Theory In Practice (O'Reilly)) From Oram, Andy/ Wilson, Greg** will really provide simple of everything to read and take the perks.

About the Author

Andy Oram is an editor at O'Reilly Media, a highly respected book publisher and technology information provider. An employee of the company since 1992, Andy currently specializes in free software and open source technologies. His work for O'Reilly includes the first books ever published commercially in the United States on Linux, and the 2001 title Peer-to-Peer. His modest programming and system administration skills are mostly self-taught.

Greg Wilson holds a Ph.D. in Computer Science from the University of Edinburgh, and has worked on high-performance scientific computing, data visualization, and computer security. He is the author of Data Crunching and Practical Parallel Programming (MIT Press, 1995), and is a contributing editor at Doctor Dobb's Journal, and an adjunct professor in Computer Science at the University of Toronto.

For everyone, if you intend to begin accompanying others to read a book, this *Beautiful Code: Leading Programmers Explain How They Think (Theory In Practice (O'Reilly)) From Oram, Andy/ Wilson, Greg* is much suggested. As well as you should get the book Beautiful Code: Leading Programmers Explain How They Think (Theory In Practice (O'Reilly)) From Oram, Andy/ Wilson, Greg below, in the web link download that we supply. Why should be here? If you desire various other type of books, you will always discover them and also Beautiful Code: Leading Programmers Explain How They Think (Theory In Practice (O'Reilly)) From Oram, Andy/ Wilson, Greg Economics, national politics, social, sciences, religious beliefs, Fictions, as well as more publications are supplied. These available books are in the soft documents.