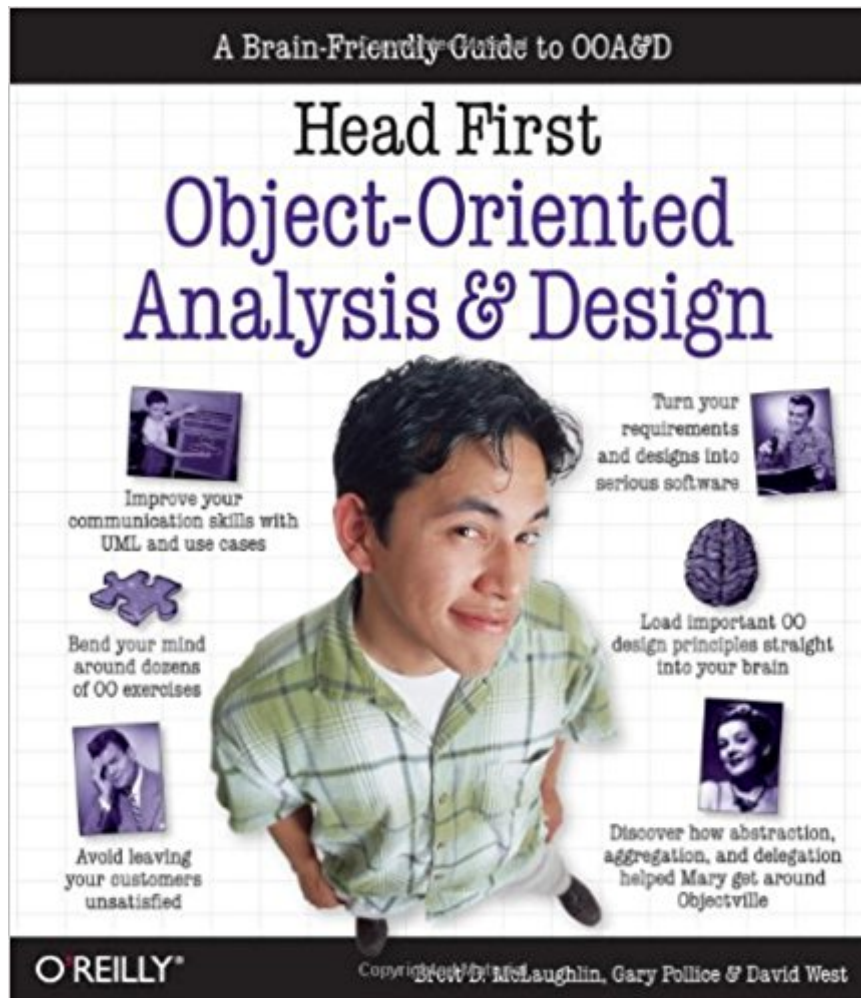




The book was found

# Head First Object-Oriented Analysis And Design



## Synopsis

"Head First Object Oriented Analysis and Design is a refreshing look at subject of OOAD. What sets this book apart is its focus on learning. The authors have made the content of OOAD accessible and usable for the practitioner." --Ivar Jacobson, Ivar Jacobson Consulting "I just finished reading HF OOA&D and I loved it! The thing I liked most about this book was its focus on why we do OOA&D-to write great software!" --Kyle Brown, Distinguished Engineer, IBM "Hidden behind the funny pictures and crazy fonts is a serious, intelligent, extremely well-crafted presentation of OO Analysis and Design. As I read the book, I felt like I was looking over the shoulder of an expert designer who was explaining to me what issues were important at each step, and why." --Edward Sciore, Associate Professor, Computer Science Department, Boston College Tired of reading Object Oriented Analysis and Design books that only makes sense after you're an expert? You've heard OOA&D can help you write great software every time-software that makes your boss happy, your customers satisfied and gives you more time to do what makes you happy. But how? Head First Object-Oriented Analysis & Design shows you how to analyze, design, and write serious object-oriented software: software that's easy to reuse, maintain, and extend; software that doesn't hurt your head; software that lets you add new features without breaking the old ones. Inside you will learn how to: Use OO principles like encapsulation and delegation to build applications that are flexible Apply the Open-Closed Principle (OCP) and the Single Responsibility Principle (SRP) to promote reuse of your code Leverage the power of design patterns to solve your problems more efficiently Use UML, use cases, and diagrams to ensure that all stakeholders are communicating clearly to help you deliver the right software that meets everyone's needs. By exploiting how your brain works, Head First Object-Oriented Analysis & Design compresses the time it takes to learn and retain complex information. Expect to have fun, expect to learn, expect to be writing great software consistently by the time you're finished reading this!

## Book Information

Series: Head First

Paperback: 636 pages

Publisher: O'Reilly Media; 1 edition (December 7, 2006)

Language: English

ISBN-10: 0596008678

ISBN-13: 978-0596008673

Product Dimensions: 8 x 1.4 x 9.2 inches

Shipping Weight: 2.6 pounds (View shipping rates and policies)

Average Customer Review: 3.8 out of 5 stars 79 customer reviews

Best Sellers Rank: #99,946 in Books (See Top 100 in Books) #7 in [Books > Computers & Technology > Programming > Software Design, Testing & Engineering > UML](#) #44 in [Books > Textbooks > Computer Science > Object-Oriented Software Design](#) #60 in [Books > Computers & Technology > Computer Science > Systems Analysis & Design](#)

## Customer Reviews

About 'Head First' Books We think of a Head First Reader as a Learner Learning isn't something that just happens to you. It's something you do. You can't learn without pumping some neurons. Learning means building more mental pathways, bridging connections between new and pre-existing knowledge, recognizing patterns, and turning facts and information into knowledge (and ultimately, wisdom). Based on the latest research in cognitive science, neurobiology, and educational psychology, Head First books get your brain into learning mode. Here's how we help you do that: We tell stories using casual language, instead of lecturing. We don't take ourselves too seriously. Which would you pay more attention to: a stimulating dinner party companion, or a lecture? We make it visual. Images are far more memorable than words alone, and make learning much more effective. They also make things more fun. We use attention-grabbing tactics. Learning a new, tough, technical topic doesn't have to be boring. The graphics are often surprising, oversized, humorous, sarcastic, or edgy. The page layout is dynamic: no two pages are the same, and each one has a mix of text and images. Metacognition: thinking about thinking If you really want to learn, and you want to learn more quickly and more deeply, pay attention to how you pay attention. Think about how you think. The trick is to get your brain to see the new material you're learning as Really Important. Crucial to your well-being. Otherwise, you're in for a constant battle, with your brain doing its best to keep the new content from sticking.

Here's what we do: We use pictures, because your brain is tuned for visuals, not text. As far as your brain's concerned, a picture really is worth a thousand words. And when text and pictures work together, we embedded the text in the pictures because your brain works more effectively when the text is within the thing the text refers to, as opposed to in a caption or buried in the text somewhere. We use redundancy, saying the same thing in different ways and with different media types, and multiple senses, to increase the chance that the content gets coded into

more than one area of your brain. We use concepts and pictures in unexpected ways because your brain is tuned for novelty, and we use pictures and ideas with at least some emotional content, because your brain is more likely to remember when you feel something. We use a personalized, conversational style, because your brain is tuned to pay more attention when it believes you're in a conversation than if it thinks you're passively listening to a presentation. We include many activities, because your brain is tuned to learn and remember more when you do things than when you read about things. And we make the exercises challenging-yet-do-able, because that's what most people prefer. We use multiple learning styles, because you might prefer step-by-step procedures, while someone else wants to understand the big picture first, and someone else just wants to see an example. But regardless of your own learning preference, everyone benefits from seeing the same content represented in multiple ways. We include content for both sides of your brain, because the more of your brain you engage, the more likely you are to learn and remember, and the longer you can stay focused. Since working one side of the brain often means giving the other side a chance to rest, you can be more productive at learning for a longer period of time. We include challenges by asking questions that don't always have a straight answer, because your brain is tuned to learn and remember when it has to work at something. Finally, we use people in our stories, examples, and pictures, because, well, you're a person. Your brain pays more attention to people than to things.

## A Brain-Friendly Guide to OOA&D

Brett McLaughlin is a bestselling and award-winning non-fiction author. His books on computer programming, home theater, and analysis and design have sold in excess of 100,000 copies. He has been writing, editing, and producing technical books for nearly a decade, and is as comfortable in front of a word processor as he is behind a guitar, chasing his two sons and his daughter around the house, or laughing at reruns of Arrested Development with his wife. Brett spends most of his time these days on cognitive theory, codifying and expanding on the learning principles that shaped the Head First series into a bestselling phenomenon. He's curious about how humans best learn, why Star Wars was so formulaic and still so successful, and is adamant that a good video game is the most effective learning paradigm we have. Gary Pollice is a self-labeled curmudgeon (that's a crusty, ill-tempered, usually old man) who spent over 35 years in industry trying to figure out what he wanted to be when he grew up. Even though he hasn't grown up yet, he did make the move in 2003 to the hallowed halls of academia where he has been corrupting the minds of the next generation of

software developers with radical ideas like, "develop software for your customer, learn how to work as part of a team, design and code quality and elegance and correctness counts, and it's okay to be a nerd as long as you are a great one." Gary is also a co-author of Head First Object-Oriented Analysis and Design. Gary is a Professor of Practice (meaning he had a real job before becoming a professor) at Worcester Polytechnic Institute. He went to WPI because he was so impressed with the WPI graduates that he's worked with over the years. He lives in central Massachusetts with his wife, Vikki, and their two dogs, Aloysius and Ignatius. When not working on geeky things he ... well he's always working on geeky things. You can see what he's up to by visiting his WPI home page at <http://web.cs.wpi.edu/~gpollice/>. Feel free to drop him a note and complain or cheer about the book. Dave West would like to describe himself as sheik geek. Unfortunately no one else would describe him in that way. They would say he is a professional Englishman who likes to talk about software development best practices with the passion and energy of an evangelical preacher. Recently Dave has moved to Ivar Jacobson Consulting, where he runs the Americas and can combine his desire to talk about software development and spread the word on rugby and football, and argue that cricket is more exciting than baseball. Before running the Americas for Ivar Jacobson Consulting, Dave worked for a number of years at Rational Software (now a part of IBM). Dave held many positions at Rational and then IBM, including Product Manager for RUP where he introduced the idea of process plug-ins and agility to RUP. Dave still laments the days when he used to sit in a cube and write software in the city of London. This is where he believes he cut his teeth writing big insurance systems with nothing but a green screen and a process flow chart. Dave can be contacted at [dwest@ivarjacobson.com](mailto:dwest@ivarjacobson.com), and if he is not with customers or drinking warm beer with his friends in Boston, he will email you back.

Simply put the book has no valuable content at all. Once you get past the Head First presentation format, which I liked before. But for this book it is just an excuse to filling page with irrelevant chit chat. The book also use Java to demonstrate examples. Unfortunately, too much Java, you will find many exercises where you fill-in the blanks in Java code. This book should be about analysis and design, not about coding. The book does mention some important concepts, but does not explain them at all. Many inaccuracies and misleading information make this book useless. Regardless of your level I would not recommend this book for anyone. After reading over 400 pages, I simply decided to stop wasting my time. There is absolutely nothing to be gained from that book. The trade-in value of this book is less than \$4.00, that speaks for itself.

It is actually a decent book. The reason that I ended up not liking a lot is that all the examples are Java. After a couple of chapters, it seemed to me more like I was reading a book about beginning to program with Java than a book about object based analysis and design. In general, I have been a fan of the Head First series, but this one was an exception for me. I was hoping for Craig Larman's Applying UML and Patterns (the textbook for object design basically) in the friendly, head first fashion instead of so academic, but not so much. So if you're a Java programmer, it's probably a worthwhile add. If you're working in C#, I'd look elsewhere.

Since I'm a software dev that hasn't really taken anything outside of a data structures class as far as computer science stuff goes (I was in electrical engineering for BS and MS), I thought maybe I should buy a couple books and see where I stand. I liked head first ajax, thought I'd give this one and the design patterns one a try, this one was first. The short conclusion is that I think this book, though a bit outdated with java 1.5 or whatever, does get a lot of good points across as far as how to consider design of classes, program layout, and whatever. At the same time, I didn't find it particularly enlightening, like a lot of it just seemed like common sense when it asked me to solve some things, and some things I don't even completely agree with but arguably that's part of the book's ideas it is trying to get across to the reader. It may be too simple for anybody who's done a bit of professional software work (like maybe they'd pick it up on the job or something), but I think it's a constructive book for anybody who maybe can't quite wrap their head around why their code is so spaghetti like or why their objects have serious copy/paste issues or poor reusability/encapsulation or something. I think I'd recommend it to anybody taking computer science in high school level? but then again I didn't take any CS classes in high school so that may be a reach also.

This is one of the texts we used in an introductory course on OOA & D. As with other Head First books, I found this book to be an easy to read and I liked the humor too. This book is pretty much aimed at beginners, so there are no prerequisites really. All examples are presented in Java but those with a different programming background should be able to follow it easily. If you are looking to a short and sweet introduction into OO land, this would be it. We also used Head First Design Patterns text for the latter part of our class. That book in combination with this book will cover enough ground for anyone to start developing software that is not only Object Oriented but also in line with best practices and methodologies. UML is used in this text but the authors don't go too crazy over it - they mention that folks may use UML to the extent they need. Towards the end of the text, the authors have couple of chapters that show the software lifecycle in action and put together

a relatively simple example that shows all the steps and they explain how it all speeds up development and delivery life cycles. All in all, it is a text worth reading but certainly not a reference.

Good book, paired with the Design Patterns book, these two can really turn a Jr developer into a Sr developer in a week or three.

Best book on object oriented analysis EVER! And I'm old enough to know! You will be entertained and educated too.

This is a very good introduction to object oriented design. It goes over some topics that even my object oriented computer science classes didn't cover. If you need an alternate resource to better understand the concepts of object oriented analysis and design, this is the book to give that to you.

It is a great book for everyone who wants to understand how to build great software. It explains all the process since the initial specs until the delivery of the system with the best quality for the customer and for the developers that will have to change the code, too. Every step in this process is well explained and helped a lot in my daily routine as a programmer. This book really teaches practical concepts about Object Oriented Analysis And Design.

[Download to continue reading...](#)

Head First Object-Oriented Analysis and Design Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development (3rd Edition) Systems Analysis and Design: An Object-Oriented Approach with UML Object-Oriented Analysis and Design with Applications (3rd Edition) Object Oriented Systems Analysis and Design Object-Oriented Analysis and Design with the Unified Process (Available Titles CengageNOW) An Object-Oriented Approach to Programming Logic and Design Object-Oriented Modeling and Design with UML (2nd Edition) Tools For Structured and Object-Oriented Design (7th Edition) Object-Oriented Design and Patterns Design Patterns: Elements of Reusable Object-Oriented Software Practical Object-Oriented Design in Ruby: An Agile Primer (Addison-Wesley Professional Ruby) Object Lessons for a Year: 52 Talks for the Children's Sermon Time (Object Lesson Series) C++ and Object-Oriented Numeric Computing for Scientists and Engineers Introduction to Programming with Greenfoot: Object-Oriented Programming in Java with Games and Simulations (2nd Edition) Object-Oriented and Classical Software Engineering The Object-Oriented Thought Process (4th Edition) (Developer's Library) Java Methods: An Introduction to Object Oriented Programming

[Contact Us](#)

[DMCA](#)

[Privacy](#)

[FAQ & Help](#)