

## Re: Basic books on OOA&D

**Source:** <http://coding.derkeiler.com/Archive/General/comp.object/2003-10/0048.html>

---

**From:** Shane Mingins (*shanemingins\_at\_yahoo.com.clothes*)

**Date:** 10/16/03

Date: Fri, 17 Oct 2003 10:17:22 +1300

Hi

I can only recommend books I have read ;-)

I found Agile Software Development (Principles, Patterns, and Practices) – Robert C. Martin great! I am constantly referring to it. Code examples are in a mixture of Java & C++ If one of your requirements is wanting further explanation on the GOF patterns then grab this book.

I found OOSC – Bertrand Meyer a difficult read. It was the first OO book I read. I am not sure (after a year of other reading) if I would find it easier now or not. This is obviously only my experience.

Object Design – Wirfs-Brock/McKean was excellent for looking at designing objects with specific roles and responsibilities. This book is lanaguage independent but where needed, Java code is used in some places to illustrate points.

I have just "skim-read" Applying UML and Patterns (An Introduction to Object-Oriented Analysis and Design and the Unified Process) – Craig Larman. This book is Martin Fowler's first choice as a book introducing OO design. I skimmed the chapters that were explaining UML notation and UP specific but enjoyed the chapters that detail an approach to OO design.

He uses the GRASP patterns which I had not heard of. Quote – "The GRASP patterns are a learning aid to help one understand essential object design, and apply design reasoning in a methodical, rational, explainable way. This approach to understanding and using design principles is based on patterns of assigning responsibilities." Again lanaguage independent with Java is used to illustrate points.

Refactoring – Martin Fowler is a "must have" book at some stage. It details refactorings which are techiques for improving the design of your code without changing it's external behaviour. Java code used for all examples.

A book that looks like it will be really good reading (when published) is Refactoring To Patterns – Joshua Kerievsk. A draft (work-in-progress) is available here <http://www.industriallogic.com/xp/refactoring/>

comp.object: Re: Basic books on OOA&D

Another two I would recommend on Test Driven Development (TDD) are Test Driven Development by Example – Kent Beck and Test Driven Development: A Practical Guide – David Astels.

Currently I am reading Patterns of Enterprise Application Architecture – Martin Fowler which at first glance looks filled with pearls of wisdom specific to Enterprise Applications if that ever is a field you end up in :-)

Now having read all these books I am still struggling daily with how to put some of all this together. Luckily there is this newsgroup and lots of helpful experts providing answers and opinions to my basic questions. If you can at all hook up with someone who is prepared to mentor you grab that opportunity!

HTH  
Shane

BTW, on a recent discussion on comp.software.extreme-programming Robert C. Martin had this to say ---

"IMHO TDD is a discipline that should be taught to programmers before their first programming language. It should be ingrained in each of us. We should cringe at the prospect of writing code without a failing test case, and without an acceptance test. For me, it is a matter of professional honor."

As a student you have an opportunity to begin this practice now rather than later :-)

In a recent article by Scott Ambler  
<http://www.sdmagazine.com/documents/s=8939/sdm0311f/sdm0311f.html> the following research was mentioned:

"Hakan Erdogmus of the National Research Council Canada described the work that he did with Maurizio Morisio, Marco Torchiano and Andrea Capiluppi of Politecnico di Torino in Italy comparing test-first and "test-last" development. It found that the test-first group, once again students, was 70 percent more productive than the group following a traditional approach. The test-first group also wrote 60 percent more tests than the test-last group, implying that the test-first mindset instilled greater discipline and thus led to greater productivity. Interestingly, on the final exam, the students reverted to a test-last approach, indicating that the culture shift required to adopt agile approaches is longer than a single school term."

--

shanemingins@yahoo.com.clothes  
remove clothes before replying

"It is not the strongest of the species that survive, nor the most intelligent, but the one most responsive to change." --- Charles Darwin  
"Jacob Atzen" <jacob@aub.dk> wrote in message  
news:877k35favp.fsf@morpheus.aub.dk...

Re: Basic books on OOA&D

## comp.object: Re: Basic books on OOA&D

> Hi,  
>  
> I'm a comp. sci. student with a bit of real world experience on  
> top. I've been programming PHP the last couple of years in a more or  
> less object oriented manner. I know the basics, classes, interfaces,  
> inheritance, polymorphism, etc.  
>  
> All the time I've felt like I'm not doing this whole OOP thing  
> right. Yes, I got objects, inheritance and the likes. But my OOA&D  
> skills are getting too limited when dealing with larger projects.  
>  
> Recently I read the GOF book. It was a real eye opener to me even  
> though I haven't begun really using the patterns yet.  
>  
> I've also been reading random articles around the net on OOA&D but it  
> seems too fragmented and I would like a more thorough walkthrough on  
> the subject.  
>  
> So the question is: Where should I start?  
>  
> As inspiring as the GOF book is, I feel I need a more high level  
> reference of the analysis and design process of small- to mediumsized  
> projects. I've browsed Amazon and came across Agile Software  
> Development which seems to match my wishes and also Meyer's OO  
> Software Construction. Would these be a good starting point or should I  
> look somewhere else?  
>  
> --  
> Thank you,  
> - Jacob Atzen