

*Reading List for Object Oriented Technology*

1. *Developments in Design Methodology*, p. 357, John Wiley & Sons (Out of Print), New York, 1984.  
This is a collection of papers on design methodology. It includes three articles with Christopher Alexander as at least one author. My favorite paper in the collection (I've not read them all) is "Planning Problems are Wicked Problems" by Horst Rittel and Melvin Webber. We should all understand the meaning of "Wicked Problems" -- I think if you look, you'll find "Wicked" was a term current at the time in describing what are now called "Chaotic" systems. Herbert Simon's paper "The Structure of Ill-structured Problems" is also included.
2. Alexander, Christopher, *The Timeless Way of Building*, Oxford University Press, New York, 1979. ISBN 0-19-502402-8  
Category: Patterns
3. Alexander, Christopher, *Notes on the Synthesis of Form*, p. 216, Harvard University Press, Cambridge, MA and London, England, 1964, Thirteenth Printing: 1994. ISBN 0-674-62751-2 (paper)  
This is the first of Alexander's books I knew, and seems to be from his PhD thesis. Here he describes a view of architecture that resonates with many software designers. The description of what he's trying to do seems particularly useful and certainly changed the way I thought when I encountered this book in grad school. Later works define a different approach to design, Patterns, but the philosophy here, in the Preface to the Paperback Edition, Part I, and the epilogue to Part II, is very important reading for designers. Take note, too, of the way the diagrams in Appendix I compose -- These are the original "sketches" that are so important in Patterns, and differ significantly from those typically seen in current software patterns. These sketches could not usefully be done in UML -- (And, I would argue, neither can those that usually are.)  
Category: Patterns
4. Alexander, Christopher, Sara Ishikawa, Murray Silverstein, Max Johnson, Ingrid Fiksdahl-King, and Shlomo Angel, *A Pattern Language*, Oxford University Press, New York, 1997. ISBN 0-19-501919-9  
Category: Patterns
5. Andleigh, P.K. and M.R. Gretzinger, *Distributed Object-Oriented Data-Systems Design*, Prentice Hall, Englewood Cliffs, 1992.  
Provides some good design concepts but the structure is sometimes confusing.
6. Association for Computing Machinery, "Conference Proceedings on: Object-Oriented Programming," *SIGPLAN Notices*. <http://www.acm.org/pubs/contents/proceedings/oops/>
7. Barton, John J. and Lee R. Nackman, *Scientific and Engineering C++*, p. 671, Addison-Wesley, Reading, MA, 1994. ISBN 0-201-53393-6  
This book illustrates the use of C++ to build matrix-like classes to solve scientific and numerical problems. The book shows many advanced idioms that use C++ parameterized types. Category: Language.
8. Beck, Ken and Ward Cunningham, "A Laboratory for Teaching Object-Oriented Thinking," *Proceedings of the Conference on Object-oriented Programming Systems, Languages and Applications (OOPSLA '89)*, ACM Press, Addison-Wesley. Special Issue of SIGPLAN Notices  
Original description of Class-Responsibilities-Collaborations (CRC Cards) technique.
9. Bentley, Jon, *Writing Efficient Programs*, Prentice-Hall, Englewood Cliffs, NJ, 1982.
10. Bentley, Jon, *More Programming Pearls*, Addison-Wesley, Reading, MA, 1988.
11. Bentley, Jon, *Programming Pearls*, Addison-Wesley, Reading, MA, 1989.

12. Booch, Grady, *The Booch Method: Process and Pragmatics*, Rational Inc, Santa Clara, 1992.
13. Booch, Grady, "The Booch Method: Notation, Part I," *Computer Language*, pp. 36-54, October, 1992.  
Summary of Booch's notation
14. Booch, Grady, "The Booch Method: Notation, Part I," *Computer Language*, pp. 47-70, September, 1992.  
Summary of Booch's notation
15. Booch, Grady, *Object-oriented Analysis and Design with Applications*, p. 589, Benjamin Cummings, Redwood City, CA, 1994. ISBN 0-8053-5340-2  
Often cited work on object-oriented design, particularly for the notation. This book describes the Booch methodology, which is one of the more popular object oriented analysis and design methods. Category: Techniques.
16. Booch, Grady, *Object Oriented Design with Applications. 2d ed*, Benjamin/Cummings, Redwood City, CA, 1994.  
This is one of the best books on object-oriented design. Mr Booch thoroughly covers the ins and outs of it. If you read nothing else, read this book
17. Booch, Grady, *Object Solutions: Managing the Object-Oriented Project*, Addison-Wesley, Reading, MA, 1996. ISBN 0-8053-0594-7  
This book is filled with practical rules and guidelines for object oriented project management. Booch talks a lot about the use of iterative and incremental software development to minimize the risks in a software development project. Category: Management.
18. Buschmann, Frank, Regine Meunier, Hans Johnert, Peter Sommerlad, and Michael Stal, *Pattern-Oriented Software Architecture: A System of Patterns*, p. 457, Wiley, Chichester, England, 1996. ISBN 0-471-95869-7  
This book discusses architecture patterns (such as Pipes and Filters, Blackboard, and Model-View-Controller) and design patterns (such as Proxy and Publisher-Subscriber). This book extends some of the original design pattern work contained in the book by Erich Gamma, et. al. Category: Patterns
19. C++ Report, *Recommended reading for C++ Programmers*, Addison Wesley, Reading, MA, 1998. ISBN 0-201-43290-0  
C++ professionals from a variety of backgrounds recommend their favorite "must have" C++ books from all publishers.
20. Cargill, Tom, *C++ Programming Style*, Addison-Wesley, Reading, MA, 1992.  
This book contains that rare and useful information--examples of bad code with analyses. Like the original Elements of Programming Style, this book is quite helpful because it shows what not to do
21. Carroll, Martin, "Design of the USL Standard Components," *C++ Report*, vol. 5, no. 5, June 1993.
22. Carroll, Martin D. and Margaret A. Ellis, *Designing and Coding Reusable C++*, p. 317, Addison-Wesley, Reading, MA, 1995. ISBN 0-201-51284-X  
This book presents many issues in the creation of reusable C++ class libraries: correct use of multiple inheritance, source file organization, error handling, initialization of objects, and other technical C++ topics. Category: Language.
23. Coad, P. and E. Yourdon, *Object-Oriented Analysis, Second Edition*, Yourdon Press, Prentice Hall, Englewood Cliffs, 1991. ISBN 0-13-629981-4  
An introduction to object-orientation. It discusses how analysis can be improved through the use of object-oriented techniques describes an object-oriented analysis method, and provides some thoughts on how to get started.

24. Coad, P. and E. Yourdon, *Object-Oriented Design*, Yourdon Press, Prentice Hall, Englewood Cliffs, 1991. ISBN 0-13-630070-7  
A follow on to the analysis book. It discusses how design can be improved through the use of object-oriented techniques, describes an object-oriented design method, and provides some thoughts on how to get started.
25. Coad, P. and J. Nicola, *Object-Oriented Programming*, Yourdon Press, Prentice Hall, Englewood Cliffs, 1993. ISBN 0-13-032616-X  
The third book in the series. It completes the discussion of Coad and Yourdon's object-oriented approach.
26. Cockburn, Alistair, *Surviving Object-Oriented Projects: a Manager's Guide*, p. 250, Addison-Wesley, Reading, MA, 1998. ISBN 0-201-49834-0  
This is an excellent practical manager's guide to the introduction of object oriented technology. This book gives a bunch of practical advice to managers on how to select the right languages and tools, how to do iterative and incremental project scheduling, how to bring in training, and how to measure your success. Category: Management.
27. Coleman, Derek, Patrick Arnold, Stephanie Bodoff, Chris Dolling, Helena Gilchrist, Fiona Hayes, and Paul Jeremaes, *Object-Oriented Development: The Fusion Method*, Prentice-Hall, Englewood Cliffs, NJ, 1994. ISBN 0-13-338823-9  
This book describes the HP Fusion methodology, which was originally developed at HP Labs. It is a simple methodology that is based on ideas taken from the Booch, Shlaer-Mellor, and Jacobson OOSE methodologies. Said to be a next generation of object-oriented methods incorporating the best features of other object-oriented methods, including "use cases" and CRC Cards. Category: Techniques.
28. Coplien, James O, *Advanced C++ Programming Styles and Idioms*, Addison-Wesley, Reading, MA, 1992.  
Lots and lots of useful techniques. However, it can be fairly dense at points, it isn't well organized, and some of the examples run counter to rules in this and other books.
29. Cornell, Gary and Cay S. Horstmann, *Core Java, second edition*, Prentice-Hall, Upper Saddle River, NJ, 1997. ISBN 0-13-596891-7  
This book is a great introduction to Java for C and C++ programmers. It gives excellent examples of good Java usage. The book is much less negative about other programming languages than most of the books written by people from Sun Microsystems. Category: Language.
30. Ellis, Margaret and Bjarne Stroustrup, *The Annotated C++ Reference Manual*, Addison-Wesley, Reading, MA, 1990.  
This book has been supplanted by the current working paper for the draft ANSI C++ standard. It reflects the state of the language at the start of the standardization process a few years ago. Neither this book (abbreviated as the ARM) nor the working paper are light bedtime reading. However, it's essential to have one of them close at hand if you hope to make optimum use of C++. If you can possibly stay awake, read one all the way through. If you can get a copy of the working paper, that is preferable to using the ARM.
31. Embley, D.W., B.D. Kurtz, and S.N. Woodfield, *Object-Oriented Systems Analysis, A Model Driven Approach*, Yourdon Press, Prentice Hall, Englewood Cliffs, 1992.
32. Flanagan, David, *Java in a Nutshell, second edition*, p. 610, O'Reilly & Associates, Sebastopol, CA, 1997. ISBN 1-56592-262-X  
This book is the best concise reference book on the Java programming language. It contains a complete reference to the Java 1.1 standard library classes. The example programs are very useful -- they illustrate many of the common uses of Java in building real applications. Category: Language.

33. Gabriel, Richard, *Patterns of Software*, p. 235, Oxford University Press, New York, 1996.  
A collection of essays on Software development. Good discussion of what Patterns are and how they relate to software design, also a good discussion of some issues in software design, particularly how they relate to people. I particularly recommend his final essay "Money Through Innovation Reconsidered" which looks at development in the real world, where "Worse is Better".
34. Gamma, Erich, Richard Helm, Ralph Johnson, and John Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, p. 395, Addison-Wesley, Reading, MA, 1995. ISBN 0-201-63361-2  
This is the famous "Gang of Four" design patterns book. An excellent source of design patterns. Useful for application architects. It contains descriptions of 23 design patterns and shows examples of their use in many different kinds of applications. Category: Patterns
35. Goldberg, Adele and Kenneth S. Rubin, *Succeeding with Objects: Decision Frameworks for Project Management*, Addison-Wesley, Reading, MA, 1995. ISBN 0-201-62878-3.  
This book is a good overview of object oriented project management issues, with some discussion of a good set of real-world case studies. Category: Management.
36. Goldstein, Neil and Jeff Alger, *Developing Object-Oriented Software for the Macintosh*, Addison-Wesley, Reading, MA, 1992.  
This book provides another excellent perspective on object-oriented design. It explodes some common myths. Don't let the title fool you--it's really not that specific to the Macintosh.
37. Hansen, Tony, *The C++ Answer Book*, Addison-Wesley, Reading, MA, 1990.
38. Harel, D., *Statecharts: A Visual Formalism for Complex Systems*, Science of Computer Programming, 8, 1987.  
Description of Harel's state transition diagrams, developed for use in large systems, especially responsive systems (real-time systems).
39. Harmon, P. and D. Taylor, *Objects in Action: Commercial Applications of Object-Oriented Technologies*, Addison Wesley, Reading, 1993. ISBN 0-201-63336-1  
Results of real-world experience with object-oriented projects as submitted to OMG. The book is organized into sections for industry segments. Reasonable material but almost all the projects are successful and very few problems are cited in the lessons learned section.
40. IBM Object Oriented Technology Center, *Developing Object-Oriented Software: an Experience-Based Approach*, p. 636, Prentice-Hall, 1997. ISBN 0-13-737248-5.  
This is a very detailed handbook of object oriented techniques from IBM's Object Oriented Technology Center. They give good descriptions of the myriad software work products that have been used in various parts of IBM, without coming out as a firm advocate of any one analysis and design methodology. Category: Techniques.
41. ISO/ANSI C++ Standardization Committee, *Working Paper for Draft Proposed American International Standard for Information Systems--Programming Language C++..*  
This is essentially the same as the ANSI X3/J16 working paper for programming language C++.
42. ISO/IEC, *ISO/IEC 9899:1990, International Standard for Information Systems -- Programming Language C.*  
This is essentially the same as ANSI C, X3J11/88-159
43. Jacobson, Ivar, Magnus Christerson, Patrik Jonsson, and Gunnar Overgaard, *Object Oriented Software Engineering - A Use Case Driven Approach, (revised)*, ACM Press, Addison Wesley, Reading, 1994. ISBN 0-201-55435-0  
The original textbook for scenario, or use-case, based design. Category: Techniques.

44. Khoshafian, S., *Object-Oriented Databases*, John Wiley & Sons, Inc, New York, 1993. ISBN 0-471-57058-3  
Good treatment of object database issues, features, and implementations. Somewhat academic in places.
45. Kolewe, R., "Metrics in Object-Oriented Design and Programming," *Software Development*, vol. 1, no. 4, pp. 53-62, October, 1993.  
Brief article outlining some complexity metrics for object-oriented projects.
46. Lakos, John, *Large Scale C++ Software Design*, p. 846, Addison-Wesley, Reading, MA, 1996. ISBN 0-201-63362-0  
This book discusses large scale software development issues in C++: how to design software to minimize recompilation and relinking time when changes are made in the middle of a design, building layers of classes to reduce circular dependencies in compilation and testing, and component design principles. Lakos summarizes his experiences in using C++ into a well-defined set of design rules, guidelines, and principles.  
Category: Language.
47. Lakos, John, *Large-Scale C++ Software Design*, Addison-Wesley professional computing series, p. 846, Addison-Wesley Pub. Co., Reading, Mass., 1996. ISBN 0-201-63362-0  
Category: Languages, C++
48. Lampson, Butler, "Hints for Computer System Design," *Operating System Review*, vol. 17, no. 1, pp. 33-48., 1983. Proceedings of the Ninth ACM Symposium on Operating System Principles  
It's got lots of good observations by someone who's been through the wringer a few times.
49. Lea, Doug, *Concurrent Programming in Java*, p. 339, Addison-Wesley, Reading, MA, 1997. ISBN 0-201-69581-2  
This book gives a good introduction to the problems posed by the concurrent programming facilities that are built into the Java programming language. Doug explains many important tips to follow to write good multithreaded Java applications. Category: Language.
50. Lee, Richard C. and William M. Tepfenhart, *UML and C++*, p. 446, Prentice-Hall, Upper Saddle River, NJ, 1997. ISBN 0-13-619719-1  
This book describes general object oriented modeling techniques, with some cross-references to C++ implementation of object oriented models. The book uses UML notation throughout, but it doesn't follow any of the "standard" object oriented analysis and design methodologies. Category: Techniques.
51. Lieberherr, K., I. Holland, and A. Riel, "Object-Oriented Programming: An Objective Sense of Style," *Proceedings of the Conference on Object-oriented Programming Systems, Languages and Applications (OOPSLA '88)*, pp. 323-334, ACM Press, Addison-Wesley.  
There are other papers on this as well. Outlines the "Law of Demeter." which is a set of rules for structuring objects so that the resulting implementations are easy to maintain and support re-use. All are available on the Internet.
52. Lippman, Stanley and Josee Lajoie, *The C++ Primer, 3rd ed*, p. 1264, Addison-Wesley, Reading, MA, 1998. ISBN 0-201-82470-1  
A gentler tutorial for C++ than *The C++ Programming Language*, 2nd Edition. The second edition of Stroustrup's book is much better organized and is more approachable, lessening the need for this book. Also, this book does not cover classes until the later chapters.
53. Love, T., *Object Lessons*, SIGS Publications, New York, 1993. ISBN 0-9627477-3-4  
Real world experiences with implementing systems using an object-oriented approach.

54. Martin, J. and J. Odell, *Object Oriented Analysis and Design*, Prentice Hall, Englewood Cliffs, 1992. ISBN 0-13-630245-9  
Another object-oriented approach. Seems to spend a lot of time in terminology. Relates OO to Information Engineering approach.
55. Martin, J. and J. Odell, *Object Oriented Methods: A Foundation*, Prentice Hall, Englewood Cliffs, 1995.
56. Martin, Robert C., *Developing Object-Oriented C++ Applications using Booch*, p. 528, Prentice Hall, Englewood Cliffs, N.J. , 1995 . ISBN 0-13-203837-4  
"An Alan R. Apt Book." Includes index.
57. Meyer, B., *Object-Oriented Software Construction*, Prentice Hall, Englewood Cliffs, 1988.  
The original source for information on the contracting approach to design. The contracting approach provides precision of specifications and a way to trace through from analysis to design to construction. The book uses the Eiffel language for examples.
58. Meyers, Scott, *More Effective C++*, p. 336, Addison-Wesley, Reading, MA, 1996. ISBN 0-201-63371-X  
This is an advanced-level follow-on to Scott Meyers' Effective C++ book. It covers several advanced features of C++, including exception handling, efficiency issues for multiple inheritance and runtime type identification, and abstract classes. Category: Language.
59. Meyers, Scott, *Effective C++, 2nd ed*, p. 288, Addison-Wesley, Reading, MA, 1998. ISBN 0-201-92488-9  
This is the most important intermediate-level C++ book for anyone who plans to do object oriented programming in C++. Scott covers all of the important topics in C++ as a series of 50 short guidelines. Category: Language.
60. Meyers, Scott, *Effective C++, 2nd ed.*, p. 288, Addison-Wesley, Reading, MA, 1998. ISBN 0-201-92488-9  
This book presents 50 good rules for designing and writing with C++ The intent of Meyers is not unlike that of Taligent's Guide to Designing Programs.
61. Murray, Robert B, *C++ Strategies and Tactics*, Addison-Wesley, Reading, MA, 1992.  
Many useful tips and techniques for using C++. Gives more detailed examples of some of the techniques discussed in this document.
62. Murray, Robert B., *C++ Strategies and Tactics*, p. 282, Addison-Wesley, Reading, MA, 1993. ISBN 0-201-56382-7  
This is a useful intermediate C++ book that illustrates many important C++ class design issues. Rob Murray gives many examples of good library class design throughout. Category: Language.
63. Musser, David R. and Atul Saini, *STL Tutorial and Reference Guide*, p. 432, Addison-Wesley, Reading, MA, 1996. ISBN 0-201-63398-1  
This book describes the Standard Template Library (STL), a set of C++ container classes that have been added to the ANSI C++ standard. This book shows how to use the STL classes and functions, and it also shows how to extend them to new container types and algorithms. Category: Language.
64. Pirsig, Robert M., "Zen and the Art of Motorcycle Maintenance," (0688052304 (paperback) 0688052307 (hardback)), p. 412, William Morrow & Co. (Also a Bantam Book), May, 1974.  
If you're a designer and you've not read this, you probably should. To me, it's all about the Quality Without A Name, others see is as about motorcycles and/or mental illness (depends on what your focus is). If you'd like to see an amazing collection of reactions to the book head for [www.amazon.com](http://www.amazon.com) and look it up there -- they have reviews/comments from many people -- some love it, some hate it. I suspect it's related to personality type, and my prejudice is the designer type is generally in the "I love it" camp.

65. Riel, Arthur J., *Object-Oriented Design Heuristics*, p. 379, Addison-Wesley, Reading, MA, 1996. ISBN 0-201-63385-X  
This book discusses a series of 61 object oriented design heuristics. It explains the most important design tradeoffs to consider when doing object oriented design. Category: Techniques.
66. Rumbaugh, James, Michael Blaha, William Premerlani, Frederick Eddy, and William Lorensen, *Object-Oriented Modeling and Design*, p. 500, Prentice-Hall, Englewood Cliffs, NJ, 1991. ISBN 0-13-629841-9  
A difficult book to read but thorough. Rumbaugh's model is widely supported with tools. A criticism of the method is that the model components are not well integrated. This book describes Object Modeling Technique (OMT), an object oriented analysis and design methodology first developed at General Electric. This method is supported by many CASE tools and is the subject of many commercial training courses. Category: Techniques.
67. Rumbaugh, James, *OMT Insights*, SIGS Books, New York, 1996. ISBN 0-13-846965-2.  
This book is a compendium of articles by one of the original creators of the OMT methodology. Rumbaugh collects here a series of articles that he wrote on OMT for the Journal of Object Oriented Programming, which describe some of the newer items that have been added to the OMT methodology, such as use cases. Category: Techniques.
68. Selic, Bran, Garth Gullekson, and Paul T. Ward, *Real-Time Object-Oriented Modeling*, p. 525, Wiley, New York, 1994. ISBN 0-471-59917-4  
This book describes the Real-Time Object Oriented Modeling (ROOM) methodology, which is supported by the commercial CASE tool ObjecTime. Category: Techniques.
69. Shlaer, S. and S.J. Mellor, *Object Oriented Systems Analysis: Modeling the World in Data*, Yourdon Press, Prentice Hall, Englewood Cliffs, 1988. ISBN 0-13-629023-X  
Well presented and easy to read. Material will be easily understood by those with a data modelling background.
70. Shlaer, S. and S.J. Mellor, *Object Lifecycles: Modeling the World in States*, Yourdon Press, Prentice Hall, Englewood Cliffs, 1992. ISBN 0-13-629940-7  
Well presented, easy to read. This small book provides a methodology from initial discussion to the beginning of physical design.
71. Shlaer, S. and S.J. Mellor, *Real-Time Recursive Design*, Project Technology, Berkeley, 1992.  
A paper giving an overview of a transformational method of physical design. Physical design is to be the topic of the third book in Schlaer/Mellor's series.
72. Stroustrup, Bjarne, "The C++ Programming Language. 3rd ed," (ISBN 0-201-88954-4), Addison-Wesley, Reading, MA, 1991.  
In addition to being the best introduction and reference to the language (excepting the ANSI draft specification), it includes excellent chapters on object-oriented design and management of object-oriented projects. These chapters are full of experience from the man who has seen more C++ projects than anyone else.  
Of special note in the 3rd edition are chapters on *Design Using C++*: this part presents C++ and the techniques it supports in the larger picture of software development. The focus is on design and the effective realization of design in terms of language constructs. Category: languages.
73. Taylor, D., *Object-Oriented Technology: A Manager's Guide*, Servio Corporation, Alameda, 1990.  
Overview material, reasonable for concepts

74. USENIX Association, *Usenix C++ Conference Proceedings*, Berkeley CA, 1987.
75. USENIX Association, *Usenix C++ Conference Proceedings*, Berkeley CA, 1988.
76. USENIX Association, *Usenix C++ Conference Proceedings*, Berkeley CA, 1989.
77. USENIX Association, *Usenix C++ Conference Proceedings*, Berkeley CA, 1990.
78. USENIX Association, *Usenix C++ Conference Proceedings*, Berkeley CA, 1991.
79. USENIX Association, *Usenix C++ Conference Proceedings*, Berkeley CA, 1992.
80. USENIX Association, *USENIX Sixth C++ Technical Conference* , Cambridge, MA, April 11 - 14, 1994. <http://www.usenix.org/publications/library/proceedings/c++94/index.html>
81. USENIX Association, *USENIX Conference on Object-Oriented Technologies (COOTS), Monterey, California*, 1995. <http://www.usenix.org/publications/library/proceedings/coots95/technical.html>
82. USENIX Association, *Second USENIX Conference on Object-Oriented Technologies (COOTS)*, 1996. <http://www.usenix.org/publications/library/proceedings/coots96/technical.html>
83. USENIX Association, *Third USENIX Conference on Object-Oriented Technologies (COOTS)*, 1997. <http://www.usenix.org/publications/library/proceedings/coots97/technical.html>
84. USENIX Association, *Fourth USENIX Object-Oriented Technologies and Systems (COOTS)*, 1998. <http://www.usenix.org/publications/library/proceedings/coots98/technical.html>
85. Walden, K. and J-M. Nerson, *Seamless Object-Oriented Software Architecture*, Prentice Hall, Englewood Cliffs, 1995.  

Presents a full analysis and design methodology using the contracting approach.
86. Webster, Bruce F., *Pitfalls of Object-Oriented Development*, p. 256, M&T Books, New York , 1995 . ISBN 1-558-51397-3  

Includes bibliographical references (p. 241-250) and index.
87. Weinberg, Gerald, *Psychology of Computer Programming*, Van Nostrand Reinhold, New York, NY, 1971.  

This is a timeless book on the activities of programming. It is useful to read this periodically. Note especially the parable of the Coke machine. Note also the parable of the special I/O system. Some choice bits about what drives programmers to worry about efficiency. Category: General
88. Wilkinson, Nancy, *Using CRC Cards*, p. 226, SIGS Books, New York, 1995. ISBN 0-13-374679-8.  

This book is an easy introduction to the CRC card process. It works through one example from early requirements to C++ code, with lots of false starts and wrong turns. It is very easy to read, but it also explains a lot of subtle points in the use of CRC cards. Category: Techniques.
89. Wirfs-Brock, Rebecca, Brian Wilkerson, and Lauren Wiener, *Designing Object-Oriented Software*, p. 341, Prentice-Hall, Englewood Cliffs, NJ, 1990. ISBN 0-13-629825-7  

Well presented, easy to read book on object-oriented design. This is the original textbook for responsibility driven design. This book is on the Responsibility Driven Design (RDD) methodology. This method focuses on assigning responsibilities to classes before creating a complex data model. Category: Techniques.