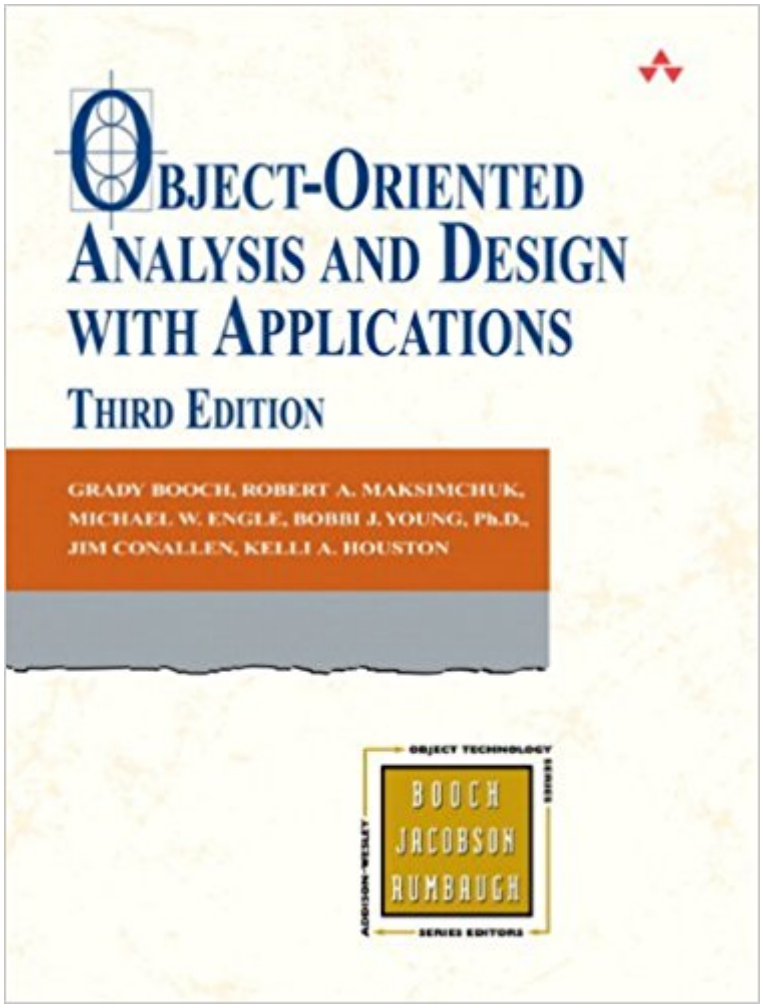
# Object-Oriented Analysis And Design With Applications (3rd Edition)



## Synopsis

Object-Oriented Design with Applications has long been the essential reference to object-oriented technology, which, in turn, has evolved to join the mainstream of industrial-strength software development. In this third edition--the first revision in 13 years--readers can learn to apply object-oriented methods using new paradigms such as Java, the Unified Modeling Language (UML) 2.0, and .NET. The authors draw upon their rich and varied experience to offer improved methods for object development and numerous examples that tackle the complex problems faced by software engineers, including systems architecture, data acquisition, cryptoanalysis, control systems, and Web development. They illustrate essential concepts, explain the method, and show successful applications in a variety of fields. You'll also find pragmatic advice on a host of issues, including classification, implementation strategies, and cost-effective project management. New to this new edition are  An introduction to the new UML 2.0, from the notation's most fundamental and advanced elements with an emphasis on key changes New domains and contexts A greatly enhanced focus on modeling--as eagerly requested by readers--with five chapters that each delve into one phase of the overall development lifecycle. Fresh approaches to reasoning about complex systems An examination of the conceptual foundation of the widely misunderstood fundamental elements of the object model, such as abstraction, encapsulation, modularity, and hierarchy How to allocate the resources of a team of developers and mange the risks associated with developing complex software systems An appendix on object-oriented programming languages  This is the seminal text for anyone who wishes to use object-oriented technology to manage the complexity inherent in many kinds of systems.  SidebarsÃ Â Ã Â  PrefaceÃ Â  Acknowledgments Ã Â Ã Â  About the Authors Ã Â Ã Â  Section I: ConceptsÃ Â Ã Â  Chapter 1: ComplexityÃ Â Ã Â Ã Â  Chapter 2: The Object Model Ã Â Ã Â  Chapter 3: Classes and Objects Ã Â Ã Â  Chapter 4: Classification Ã Â Ã Â  Section II: Method Ã Â  Chapter 5: Notation Ã Â Ã Â  Chapter 6: Process  Chapter 7: PragmaticsÃ Â Ã Â Ã Â  Chapter 8: System Architecture: Satellite-Based Navigation Ã Â Ã Â  Chapter 9: Control System: Traffic Management Ã Â Ã Â  Chapter 10: Artificial Intelligence: Cryptanalysis Ã Â Ã Â  Chapter 11: Data Acquisition: Weather Monitoring Station Ã Â  Chapter 12: Web Application: Vacation Tracking System Ã Â Ã Â Ã Â  Appendix A: Object-Oriented Programming LanguagesÃ Â  Appendix B: Further Reading Ã Â Ã Â  Notes Ã Â Ã Â  Glossary Ã Â Ã Â  Classified Bibliography Ã Â Ã Â  Index Ã Â Ã Â

## Book Information

Hardcover: 720 pages

Publisher: Addison-Wesley Professional; 3 edition (April 30, 2007)

Language: English

ISBN-10: 020189551X

ISBN-13: 978-0201895513

Product Dimensions: 7.1 x 1.7 x 9.4 inches

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

Average Customer Review: 4.1 out of 5 stars 17 customer reviews

Best Sellers Rank: #173,626 in Books (See Top 100 in Books) #75 inÃ Â Books > Textbooks > Computer Science > Object-Oriented Software Design #266 inÃ Â Books > Computers & Technology > Programming > Software Design, Testing & Engineering > Object-Oriented Design #1196 inÃ Â Books > Computers & Technology > Programming > Languages & Tools

## Customer Reviews

Object-Oriented Design with Applications has long been the essential reference to object-oriented technology, which, in turn, has evolved to join the mainstream of industrial-strength software development. In this third edition--the first revision in 13 years--readers can learn to apply object-oriented methods using new paradigms such as Java, the Unified Modeling Language (UML) 2.0, and .NET.The authors draw upon their rich and varied experience to offer improved methods for object development and numerous examples that tackle the complex problems faced by software engineers, including systems architecture, data acquisition, cryptoanalysis, control systems, and Web development. They illustrate essential concepts, explain the method, and show successful applications in a variety of fields. You'll also find pragmatic advice on a host of issues, including classification, implementation strategies, and cost-effective project management.New to this new edition areAn introduction to the new UML 2.0, from the notation's most fundamental and advanced elements with an emphasis on key changesNew domains and contextsA greatly enhanced focus on modeling--as eagerly requested by readers--with five chapters that each delve into one phase of the overall development lifecycle.Fresh approaches to reasoning about complex systemsAn examination of the conceptual foundation of the widely misunderstood fundamental elements of the object model, such as abstraction, encapsulation, modularity, and hierarchyHow to allocate the resources of a team of developers and mange the risks associated with developing complex software systemsAn appendix on object-oriented programming languagesThis is the seminal text for anyone who wishes to use object-oriented technology to manage the complexity inherent in many

kinds of systems. "Sidebars Preface Acknowledgments About the Authors " Section I: Concepts Chapter 1: Complexity  Chapter 2: The Object Model  Chapter 3: Classes and Objects  Chapter 4: Classification  Section II: Method  Chapter 5: Notation  Chapter 6: Process  Chapter 7: Pragmatics Chapter 8: System Architecture: Satellite-Based Navigation  Chapter 9: Control System: Traffic Management  Chapter 10: Artificial Intelligence: Cryptanalysis  Chapter 11: Data Acquisition: Weather Monitoring Station  Chapter 12: Web Application: Vacation Tracking System  Appendix A: Object-Oriented Programming Languages  Appendix B: Further Reading " Notes Glossary Classified Bibliography Index """"

Grady Booch is an IBM fellow and author of six best-selling books on object-oriented programming. He is world-reknowned as an originator of OO and founder of UML.  Robert A. Maksimchuk, as Research Director in the Unisys CTO Office, focuses on emerging modeling technologies to advance the strategic direction of the Unisys 3D-Visual Enterprise modeling framework. Bob brings an abundance of systems engineering, modeling, and object-oriented analysis and design expertise, in numerous industries, to this mission. He is the coauthor of the books UML for Mere Mortals and UML for Database Design, has written various articles, has traveled worldwide as a featured speaker in numerous technology forums, and led workshops and seminars on UML and object-oriented development.  Michael W. Engle is a principal member of the engineering staff with the Lockheed Martin Corporation. He has extensive technical and management experience across the complete system development lifecycle, from project initiation through deployment and support in a variety of application domains. As a systems architect, Mike employs object-oriented analysis nad design techniques in complex systems development.  Dr. Bobbi Young is a Director of Research for the Unisys Chief Technology Office. She has many years of experience in the IT industry working with commercial companies and Department of Defense contractors. Dr. Young has been a consultant mentoring in program management, enterprise architecture, systems engineering, and object-oriented analysis and design. Throughout her career, she has focused on system lifecycle processes and methodologies, and enterprise architecture.  Jim Conallen is a software engineer in IBM Rational's Model Driven Development Strategy team, where he is actively involved in applying the Object Management Group's (OMG) Model Driven Architecture (MDA) initiative to IBM Rational's model tooling.  Kelli A. Houston is a Consulting IT Specialist at IBM Rational. She is the method architect for IBM's internal method authoring method and is part of the team responsible for integrating IBM's methods.

I love this book. And this is my opinion: I bought it the first time back in the late 90s when I was in college and recently I got the latest edition for kindle format so I can access it from my phone. This is not a book that if you read it, you will be an OOP expert, but it helps to understand many concepts. In my case it is part of the foundation of my knowledge as an IT professional. It is so clear and well explained the concept of Class; Object; Polymorphism; Encapsulation, Abstraction, Inheritance, etc..

Only about halfway through but as many say this guy has a genius for explaining the subject that makes it crystal clear. A pleasure to read, nothing dry and stuffy and the analogies are spot-on so that even an old structured programming guy can understand and appreciate OO. I think it will continue to be the go-to reference for a long time to come.

This 3rd edition is the eagerly and long awaited update to the 2nd edition which was published in 1994. It provides thorough and practical coverage of concepts, techniques, notations and examples for modern object-oriented analysis and design. The material cov-ered draws upon a solid foundation of theoretical work but is con-sistently pragmatic in approach. This book provides an essential body of knowledge for professionals responsible for the analysis and design of complex systems.As with the second edition, the book is organized into three major sections - Concepts, Method and Applications. Concepts intro-duces the fundamental principles of object-oriented analysis and design (OOAD) such as creating abstractions, objects and classes, and how to address the complexities found in a variety of systems. The Method section focuses on how to analyze and design com-plex systems with an emphasis on using UML 2.This edition follows a similar format and addresses many of the same topics as its predecessor but varies in several areas. Most noticeably, the famous "clouds" and other Booch notations used in the 2nd edition have all been replaced with UML. The UML dia-grams also make frequent use of the newer UML 2 notations such as frames on sequence diagrams and ports on component diagrams.As a whole, the new set of applications nicely cover a variety of challenges found in modern systems design. There are also few-er code examples in this edition. However, as the frequent use of Courier font suggests, the text still sits conceptually just slightly above code level when that is necessary.This book is very well organized, written and edited. For example, in the Methods section, the chapter on Notation doesn’t merely plod through the syntax of various shapes and line styles but ex-plains each diagram set with regard to intended use and contribu-tion to object-oriented models. There are also clear and informative distinctions between essential techniques and more advanced concepts.

Though some concepts clearly build upon each other, the reader is not forced to read the material in a certain order - e.g., references to material in other chapters are clearly marked and summarized.Diagram styles vary somewhat from chapter to chapter but, as ex-plained in the preface, this is deliberate in order to familiarize the reader with the output of commonly used tools. More extensive or sharply defined distinctions between what is common practice as compared to alternative approaches, including the risks or benefits of either, would have been nice to have but this omission doesn't detract from what is a great book overall.Though intended primarily for developers and architects of soft-ware systems, the material presented would also be highly valuable to analysts in non-engineering roles such as business systems ana-lysts. It is also a worthwhile read for those working on systems without a software emphasis.Analysts, designers and architects of complex systems, will find this text provides broad and deep coverage in the current practice of OOAD. As a result, it should regarded as mandatory reading for professionals in those fields.[...]

I was reviewing the Gang of Four's book on Design Patterns and read Grady Booch's Forward. I decided to buy and read his Object-Oriented Analysis and Design with Applications book. I am very glad that I did. Booch does a great job of presenting the foundations of OOD, connecting all of the dots, and demonstrating how to apply OOD to various applications. I think this is a must-read for any professional OOP designer/developer.

Topics are already seen!

This book's reputation as one of the bibles of OOAD is probably deserved because (to someone relatively new to it) the essentials seem to be thoroughly covered. It just seems too much like wading through muck to find them.The problem begins at the very beginning; on the first pages of the preface. In describing changes between publication of the second edition and this third edition, the author lists "robots are cruising on the surface of Mars" and "Personal hovercraft are available." Tongue-in-cheek?Unfortunately, no, unless it's firmly planted there. As the book continues, the reader all too often wants to start skimming as paragraph after paragraph, sometimes page after page, of non-essential prattle clouds the essentials. For journeyman designers and developers, sections on the topology of old-fashioned procedural languages, on the importance of documentation, task planning, release planning (twice!) and more may be frustrating drags on learning the essentials of thinking through a good design and taking it to the doorstep of

implementation.A highly-simplified greenhouse application is used for examples throughout the first part of the book, leaving too many more-common scenarios unexplored and occasionally trapping skimmers who have not captured every concept in the design of that application along the way.Late chapters illustrate some concepts with (finally!) other applications including an all-important (for many of us) web application as well as applications for satellite tracking, data aquisition for a weather station, artificial intelligence, and a control system for traffic management. Interesting, but again wordy and by the time you get there you're exhausted!I did learn from this book, but I'm still looking for The Book that efficiently teaches OOAD, and I've read four or five already. So far I've learned more from a couple of implementation-level books: Martin Fowler's superb book Refactoring: Improving the Design of Existing Code, and his UML Distilled. These have been very instructive in part because Fowler's style is lean and very clear, un-clouded by distracting non-essentials. I've just ordered Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development by Craig Larman. Fingers are crossed, maybe that will be The Book.

Author described the OOAD sens, paradigm and advantages. All part of text about OOAD are illustrated in UML 2.0 diagrams with comments. Second half of the book include some examples with explanation.

I was expecting the material to be dated, but it was quite relevant. It gives a good background for how we got where we are.

Process (4th Edition) (Developer’s Library) Object-Oriented Programming in C++ (4th Edition) Object-Oriented Programming in Java: A Graphical Approach, Preliminary Edition Object Oriented Software Development Using Java (2nd Edition) 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