

Uml

More than 300,000 developers have benefited from past editions of UML Distilled . This third edition is the best resource for quick, no-nonsense insights into understanding and using UML 2.0 and prior versions of the UML. Some readers will want to quickly get up to speed with the UML 2.0 and learn the essentials of the UML. Others will use this book as a handy, quick reference to the most common parts of the UML. The author delivers on both of these promises in a short, concise, and focused presentation. This book describes all the major UML diagram types, what they're used for, and the basic notation involved in creating and deciphering them. These diagrams include class, sequence, object, package, deployment, use case, state machine, activity, communication, composite structure, component, interaction overview, and timing diagrams. The examples are clear and the explanations cut to the fundamental design logic. Includes a quick reference to the most useful parts of the UML notation and a useful summary of diagram types that were added to the UML 2.0. If you are like most developers, you don't have time to keep up with all the new innovations in software engineering. This new edition of Fowler's classic work gets you acquainted with some of the best thinking about efficient object-oriented software design using the UML--in a convenient format that will be essential to anyone who designs software professionally.

Diagramming and process are important topics in today's software development world, as the UML diagramming language has come to be almost universally accepted. Yet process is necessary; by themselves, diagrams are of little use. Use Case Driven Object Modeling with UML - Theory and Practice combines the notation of UML with a lightweight but effective process - the ICONIX process - for designing and developing software systems. ICONIX has developed a growing following over the years. Sitting between the free-for-all of Extreme Programming and overly rigid processes such as RUP, ICONIX offers just enough structure to be successful.

This book presents a variant of UML that is especially suitable for agile development of high-quality software. It adjusts the language UML profile, called UML/P, for optimal assistance for the design, implementation, and agile evolution to facilitate its use especially in agile, yet model based development methods for data intensive or control driven systems. After a general introduction to UML and the choices made in the development of UML/P in Chapter 1, Chapter 2 includes a definition of the language elements of class diagrams and their forms of use as views and representations. Next, Chapter 3 introduces the design and semantic facets of the Object Constraint Language (OCL), which is conceptually improved and syntactically adjusted to Java for better comfort. Subsequently, Chapter 4 introduces object diagrams as an independent, exemplary notation in UML/P, and Chapter 5 offers a detailed introduction to UML/P Statecharts. Lastly, Chapter 6 presents a simplified form of sequence diagrams for exemplary descriptions of object interactions. For completeness, appendixes A–C describe the full syntax of UML/P, and appendix D explains a sample application from the E-commerce domain, which is used in all chapters. This book is ideal for introductory courses for students and practitioners alike.

Globe-trotting travelers have long resorted to handy, pocket-size dictionaries as an aid to communicating across the language barrier. Dan Pilone's UML 2.0 Pocket Reference is just such an aid for on-the-go developers who need to converse in the Unified Modeling Language (UML). Use this book to decipher the many UML diagrams you'll encounter on the path to delivering a modern software system. Updated to cover the very latest in UML, you'll find coverage of the following UML 2.0 diagram types: Class diagrams Component diagrams* Sequence diagrams* Communication diagrams* Timing diagrams* Interaction Overview diagrams* Package diagrams* Deployment diagrams* Use case

diagrams Composite structure diagrams* Activity diagrams* Statechart diagrams* * New or expanded coverage in this edition Also new in this edition is coverage of UML's Object Constraint Language (OCL). Using OCL, you can specify more narrowly the functionality described in a given diagram by recording limits that are the result of business rules and other factors. The UML 2.0 Pocket Reference travels well to meetings and fits nicely into your laptop bag. It's near impossible to memorize all aspects of UML, and with this book along, you won't have to.

"This book manages to convey the practical use of UML 2 in clear and understandable terms with many examples and guidelines. Even for people not working with the Unified Process, the book is still of great use. UML 2 and the Unified Process, Second Edition is a must-read for every UML 2 beginner and a helpful guide and reference for the experienced practitioner." --Roland Leibundgut, Technical Director, Zuehlke Engineering Ltd. "This book is a good starting point for organizations and individuals who are adopting UP and need to understand how to provide visualization of the different aspects needed to satisfy it. " --Eric Naiburg, Market Manager, Desktop Products, IBM Rational Software

This thoroughly revised edition provides an indispensable and practical guide to the complex process of object-oriented analysis and design using UML 2. It describes how the process of OO analysis and design fits into the software development lifecycle as defined by the Unified Process (UP). UML 2 and the Unified Process contains a wealth of practical, powerful, and useful techniques that you can apply immediately. As you progress through the text, you will learn OO analysis and design techniques, UML syntax and semantics, and the relevant aspects of the UP. The book provides you with an accurate and succinct summary of both UML and UP from the point of view of the OO analyst and designer. This book provides Chapter roadmaps, detailed diagrams, and margin notes allowing you to focus on your needs Outline summaries for each chapter, making it ideal for revision, and a comprehensive index that can be used as a reference New to this edition: Completely revised and updated for UML 2 syntax Easy to understand explanations of the new UML 2 semantics More real-world examples A new section on the Object Constraint Language (OCL) Introductory material on the OMG's Model Driven Architecture (MDA) The accompanying website provides A complete example of a simple e-commerce system Open source tools for requirements engineering and use case modeling Industrial-strength UML course materials based on the book

This comprehensive guide has been fully revised to cover UML 2.0, today's standard method for modelling software systems. Filled with concise information, it's been crafted to help IT professionals read, create, and understand system artefacts expressed using UML. Includes an example-rich tutorial for those who need familiarizing with the system.

This book covers all you need to know to model and design software applications from use cases to software architectures in UML and shows how to apply the COMET UML-based modeling and design method to real-world problems. The author describes architectural patterns for various architectures, such as broker, discovery, and transaction patterns for service-oriented architectures, and addresses software quality attributes including maintainability, modifiability, testability, traceability, scalability, reusability, performance, availability, and security. Complete case studies illustrate design issues for different software architectures: a banking system for client/server architecture, an online shopping system for service-oriented architecture, an emergency monitoring system for component-based software architecture, and an automated guided vehicle for real-time software architecture. Organized as an introduction followed by several short, self-contained chapters, the book is perfect for senior undergraduate or graduate courses in software engineering and design, and for experienced software engineers wanting a quick reference at each stage of the analysis, design, and development of large-scale software systems.

CD-ROM contains: Java and XML implementations of ideas and models described in the appendix.

This book constitutes the thoroughly refereed post-workshop proceedings of the International Workshop on the Unified Modeling Language, '98: Beyond the Notation, held in Mulhouse, France in June 1998. The 33 revised full papers presented were carefully selected by the editorial committee after two rounds of reviewing. The volume presents state-of-the-art R&D results as well as advanced applications; it is indispensable reading for anybody seriously interested in UML.

The Unified Modeling Language has become the industry standard for the expression of software designs. The Java programming language continues to grow in popularity as the language of choice for the serious application developer. Using UML and Java together would appear to be a natural marriage, one that can produce considerable benefit. However, there are nuances that the seasoned developer needs to keep in mind when using UML and Java together. Software expert Robert Martin presents a concise guide, with numerous examples, that will help the programmer leverage the power of both development concepts. The author ignores features of UML that do not apply to Java programmers, saving the reader time and effort. He provides direct guidance and points the reader to real-world usage scenarios. The overall practical approach of this book brings key information related to Java to the many presentations. The result is an highly practical guide to using the UML with Java.

This textbook mainly addresses beginners and readers with a basic knowledge of object-oriented programming languages like Java or C#, but with little or no modeling or software engineering experience – thus reflecting the majority of students in introductory courses at universities. Using UML, it introduces basic modeling concepts in a highly precise manner, while refraining from the interpretation of rare special cases. After a brief explanation of why modeling is an indispensable part of software development, the authors introduce the individual diagram types of UML (the class and object diagram, the sequence diagram, the state machine diagram, the activity diagram, and the use case diagram), as well as their interrelationships, in a step-by-step manner. The topics covered include not only the syntax and the semantics of the individual language elements, but also pragmatic aspects, i.e., how to use them wisely at various stages in the software development process. To this end, the work is complemented with examples that were carefully selected for their educational and illustrative value. Overall, the book provides a solid foundation and deeper understanding of the most important object-oriented modeling concepts and their application in software development. An additional website offers a complete set of slides to aid in teaching the contents of the book, exercises and further e-learning material.

This book presents the analysis, design, documentation, and quality of software solutions based on the OMG UML v2.5. Notably it covers 14 different modelling constructs including use case diagrams, activity diagrams, business-level class diagrams, corresponding interaction diagrams and state machine diagrams. It presents the use of UML in creating a Model of the Problem Space (MOPS), Model of the Solution Space (MOSS) and Model of the Architectural Space (MOAS). The book touches important areas of contemporary software engineering ranging from how a software engineer needs to invariably work in an Agile development environment through to the techniques to model a Cloud-based solution.

UML Distilled: A Brief Guide to the Standard Object Modeling Language Addison-Wesley Professional

Typically, analysis, development, and database teams work for different business units, and use different design notations. With UML and the Rational Unified Process (RUP), however, they can unify their efforts -- eliminating time-consuming, error-prone translations, and accelerating software to market. In this book, two data modeling specialists from Rational Software Corporation show exactly how to model data with UML and RUP, presenting proven processes and start-to-finish case studies. The book utilizes a running case study to bring together the entire

process of data modeling with UML. Each chapter dissects a different stage of the data modeling process, from requirements through implementation. For each stage, the authors cover workflow and participants' roles, key concepts, proven approach, practical design techniques, and more. Along the way, the authors demonstrate how integrating data modeling into a unified software design process not only saves time and money, but gives all team members a far clearer understanding of the impact of potential changes. The book includes a detailed glossary, as well as appendices that present essential Use Case Models and descriptions. For all software team members: managers, team leaders, systems and data analysts, architects, developers, database designers, and others involved in building database applications for the enterprise.

Unified Modeling Language (UML) is a general-purpose notation language for specifying and visualizing complex software, especially large, object-oriented projects. Object-oriented programming is when a programmer defines not only the data type of a data structure, but also the types of operations/functions that can be applied to the data structure. Applying UML addresses the practical issues faced by users in adopting UML. As the title suggests, it helps the reader in actually applying UML to real life situations, rather than just in learning the language. The book covers in depth detail of UML, including notation on profiles and extensions. The scope of the book assumes prior experience in software engineering and/or business modeling, an understanding of object-oriented concepts and a basic knowledge of UML.

* Case study driven approach covering a wide range of issues * Contains advanced tutorial material to aid learning * Focuses on practical issues in the application of UML

Overviews the process of building and compiling executable UML models for software development. The book focuses on the BridgePoint tool suite and object action language developed by Project Technology. The authors discuss identifying system requirements, diagramming classes and attributes, constraints on the class diagram, ways of building sets of communicating statechart diagrams, and model verification. Annotation copyrighted by Book News, Inc., Portland, OR.

This book constitutes the thoroughly refereed joint postproceedings of the satellite activities held at the 7th International Conference on the Unified Modeling Language, UML 2004, in Lisbon, Portugal in October 2004 complementing the main conference track. The book presents reports on the 10 workshops held at UML and covers a broad range of topics around systems modelling; these reports are compiled by the respective workshop organizers. Furthermore 12 revised reviewed papers from the industry track are included as well as 11 short papers corresponding to selected poster/demo presentations and a summary on the UML tools exhibition.

The UML was conceived and first implemented as a language for describing the design of object-oriented programs. Its widespread adoption and inherent flexibility has, inevitably, led to its use in other areas, including the design of component-based systems. While it is not a perfect fit for component-based development, this book describes how best to use UML 1.3 in the specification and design of medium to large systems that utilize server-side component technologies.

This new book is the definitive primer for UML, and starts with the foundational concepts of object-orientation in order to provide the proper context for explaining UML.

The Unified Modeling Language (UML) is one of the most important languages for anyone in the software industry to know. The UML is a visual language enabling architects, designers, and developers to communicate about design. Seemingly simple on the surface, the UML is a rich and expressive language, with many visual syntactical elements. It's next to impossible to memorize all aspects of the UML. Just as a writer might require a dictionary to work with the spoken word, so too do UML practitioners require a dictionary of sorts. In this book, you'll

find information on UML usage, and also on the symbols, line-endings, and syntax used for the following diagram types: Class diagrams
Component diagrams Behavioral diagrams Sequence diagrams Statechart diagrams Object diagrams Deployment diagrams Use case
diagrams Collaboration diagrams Activity diagrams Let this book be your UML dictionary. It's clear, concise, and small. Keep this book at
hand, and never again be stymied by an unfamiliar UML symbol, a line-ending you don't recognize, or the use of an unfamiliar diagram type.
O'Reilly's Pocket References have become a favorite among programmers everywhere. By providing a wealth of important details in a
concise, well-organized format, these handy books deliver just what you need to complete the task at hand. When you need to get to a
solution quickly, the new UML Pocket Reference is the book you'll want to have.

ABOUT THE TECHNOLOGY What it is: UML (Unified Modeling Language) is a graphical modeling language used to specify, visualize,
construct, and document applications and software systems, which are implemented with components and object-oriented programming
languages, such as Java, C++, and Visual Basic. UML incorporates the object-oriented community's consensus on core modeling concepts
and provides a standard way for developers to communicate the details of system design and development. In addition to object-oriented
modeling of applications, UML is also used for business-process modeling, data modeling, and XML modeling. Purpose of modeling: Models
for software systems are as important as having a blueprint for a large building, or an outline for a book. Good models enhance
communication among project teams and assure architectural soundness. The more complex the software system, the more important it is to
have models that accurately describe the system and can be understood by everyone. UML helps provide this via a standard for graphical
diagrams. Just like an architect can understand the notations on any blueprint, UML enables software engineers and business managers to
understand the design of any software system, even if the original designers have long left the company. Organization behind it: Object
Management Group (OMG) (www.omg.org). (UML Resource Page at OMG Web site is www.omg.org/uml.) The OMG produces and
maintains the UML standard, an internationally recognized standard. The OMG is an open membership, not-for-profit consortium that
produces and maintains computer industry specifications for interoperable enterprise applications. Its membership roster (about 800) includes
just about every large company in the computer industry and hundreds of smaller ones. Most of the companies that shape enterprise and
Internet computing are represented on the OMG's Board of Directors. Companies that contributed to the UML Standard: Realizing that UML
would be strategic to their business, the following companies contributed their ideas to the first UML standard: Digital Equipment Corp, HP, i-
Logix, IntelliCorp, IBM, ICON Computing, MCI, Microsoft, Oracle, Rational Rose, TI, and Unisys. Companies that use UML: It is safe to say
that all Fortune 1000 companies are currently using UML, or are moving toward UML to model and design their applications and systems.
This includes companies from all vertical industries, from Coca Cola to Warner Brothers, from CVS Pharmacy to Lockheed Martin Aerospace.
You name the company - if they have an IT department, they are using UML.

System developers have used modeling languages for decades to specify, visualize, construct, and document systems. The Unified Modeling
Language (UML) is one of those languages. UML makes it possible for team members to collaborate by providing a common language that
applies to a multitude of different systems. Essentially, it enables you to communicate solutions in a consistent, tool-supported
language. Today, UML has become the standard method for modeling software systems, which means you're probably confronting this rich
and expressive language more than ever before. And even though you may not write UML diagrams yourself, you'll still need to interpret
diagrams written by others. UML 2.0 in a Nutshell from O'Reilly feels your pain. It's been crafted for professionals like you who must read,
create, and understand system artifacts expressed using UML. Furthermore, it's been fully revised to cover version 2.0 of the language. This

comprehensive new edition not only provides a quick-reference to all UML 2.0 diagram types, it also explains key concepts in a way that appeals to readers already familiar with UML or object-oriented programming concepts. Topics include: The role and value of UML in projects The object-oriented paradigm and its relation to the UML An integrated approach to UML diagrams Class and Object, Use Case, Sequence, Collaboration, Statechart, Activity, Component, and Deployment Diagrams Extension Mechanisms The Object Constraint Language (OCL) If you're new to UML, a tutorial with realistic examples has even been included to help you quickly familiarize yourself with the system. Presents a step-by-step process to master object-oriented analysis and design, from requirements gathering all the way to code generation, using the latest version of the industry standard modeling language. Original. (Advanced)

Topological UML Modeling: An Improved Approach for Domain Modeling and Software Development presents a specification for Topological UML® that combines the formalism of the Topological Functioning Model (TFM) mathematical topology with a specified software analysis and design method. The analysis of problem domain and design of desired solutions within software development processes has a major impact on the achieved result – developed software. While there are many tools and different techniques to create detailed specifications of the solution, the proper analysis of problem domain functioning is ignored or covered insufficiently. The design of object-oriented software has been led for many years by the Unified Modeling Language (UML®), an approved industry standard modeling notation for visualizing, specifying, constructing, and documenting the artifacts of a software-intensive system, and this comprehensive book shines new light on the many advances in the field. Presents an approach to formally define, analyze, and verify functionality of existing processes and desired processes to track incomplete or incorrect functional requirements Describes the path from functional and nonfunctional requirements specification to software design with step-by-step creation and transformation of diagrams and models with very early capturing of security requirements for software systems. Defines all modeling constructs as extensions to UML®, thus creating a new UML® profile which can be implemented in existing UML® modeling tools and toolsets

How to use UML to model Enterprise JavaBeans, Swing components, CORBA, and other popular technologies Enterprise Java with UML is the first comprehensive guide on using UML (Unified Modeling Language) to model Java applications. Written by three well-known members of the UML and Java community, the book presents strategies for developing enterprise systems using Java and related technologies -- XML, Servlets, Enterprise JavaBeans, Swing Components, CORBA, RMI, and others. The authors explain how UML is used as a modeling tool for object-oriented computer systems in the real world, break down common situations that development teams encounter, and discuss the tradeoffs of using different technologies in different combinations. They also explore different products, looking closely at their strengths and weaknesses. Four in-depth studies complete the presentation, showing readers how to make the right decision for their project through examples of both successes and failures. UML is an industry standard specification for modelling, visualizing, and documenting software projects. This title covers all aspects of the UML including the use of the UML, diagramming notation, the object constraint language (OCL), and profiles. Concise and easy-to-understand guidelines and standards for creating UML 2.0 diagrams.

The Unified Modeling Language (UML), for the first time in the history of systems engineering, gives practitioners a common language. This concise quick reference explains how to use each component of the language, including its extension mechanisms

and the Object Constraint Language (OCL). A tutorial with realistic examples brings those new to the UML quickly up to speed. Uses friendly, easy-to-understand For Dummies style to help readers learn to model systems with the latest version of UML, the modeling language used by companies throughout the world to develop blueprints for complex computer systems. Guides programmers, architects, and business analysts through applying UML to design large, complex enterprise applications that enable scalability, security, and robust execution. Illustrates concepts with mini-cases from different business domains and provides practical advice and examples. Covers critical topics for users of UML, including object modeling, case modeling, advanced dynamic and functional modeling, and component and deployment modeling.

This fifth edition continues to build upon previous issues with its hands-on approach to systems analysis and design with an even more in-depth focus on the core set of skills that all analysts must possess. Dennis continues to capture the experience of developing and analysing systems in a way that readers can understand and apply and develop a rich foundation of skills as a systems analyst.

"...(an) exceptionally balanced and informative text." --Rich Dragan

The Unified Modeling Language (UML) is a third generation method for specifying, visualizing, and documenting an object-oriented system under development. It unifies the three leading object-oriented methods and others to serve as the basis for a common, stable, and expressive object-oriented development notation. As the complexity of software applications increases, so does the developer's need to design and analyze applications before developing them. This practical introduction to UML provides software developers with an overview of this powerful new design notation, and teaches Java programmers to analyse and design object-oriented applications using the UML notation.

- + Apply the basics of UML to your applications immediately, without having to wade through voluminous documentation
- + Use the simple Internet example as a prototype for developing object-oriented applications of your own
- + Follow a real example of an Intranet sales reporting system written in Java that is used to drive explanations throughout the book
- + Learn from an example application modeled both by hand and with the use of Popkin Software's SA/Object Architect O-O visual modeling tool.

Here you will learn how to develop an attractive, easily readable, conceptual, business-oriented entity/relationship model, using a variation on the UML Class Model notation. This book has two audiences:

- Data modelers (both analysts and database designers) who are convinced that UML has nothing to do with them;
- and • UML experts who don't realize that architectural data modeling really is different from object modeling (and that the differences are important).

David Hay's objective is to finally bring these two groups together in peace. Here all modelers will receive guidance on how to produce a high quality (that is, readable) entity/relationship model to describe the data architecture of an organization. The notation involved happens to be the one for class models in the Unified Modeling Language, even though UML was originally developed to support object-oriented design. Designers have a different view of the world from those who develop business-oriented conceptual data models, which means that to use UML for architectural modeling requires some adjustments. These adjustments are described in this book. David Hay is the author of *Enterprise Model Patterns: Describing the World*, a comprehensive model of a generic enterprise. The diagrams were at

various levels of abstraction, and they were all rendered in the slightly modified version of UML Class Diagrams presented here. This book is a handbook to describe how to build models such as these. By way of background, an appendix provides a history of the two groups, revealing the sources of their different attitudes towards the system development process. If you are an old-school ER modeler and now find yourself having to come up to speed on UML to get that next job (or keep the current one), this is your guidebook to success. If you are a long time object oriented programmer who has to interact with data modelers, this book is for you too. David has done the hard work of mapping out how to do a logical entity relationship model using standard (and accepted) UML diagram components. This book shows you step-by-step, with ample examples, how to get from here to there with the least pain possible for all concerned. Kent Graziano Certified Data Vault Master and Oracle ACE Past-President of ODTUG & RMOUG Brilliantly organized: three books hidden in one cohesive work. Notwithstanding the tremendous value provided by cross-training data architects/modelers and object modelers/architects, making each better at what they do, Appendix B presents an absolutely awesome concise, yet detailed, history of modeling objects and data that clearly documents the differences in the approaches over the years and helps bring it all into perspective. This book is packed with useful information. Even the footnotes add clarity and offer interesting and often humorous editorial insight making it a fun read. Whatever viewpoint the reader is coming from this book has something to offer as long as the reader maintains an open mind. Roland Berg Senior Architect Diligent Consulting, Inc. San Antonio, Texas

With nearly 250,000 sold, Harvey and Paul Deitel's C++ How to Program is the world's best-selling introduction to C++ programming. Now, this classic has been thoroughly updated! The authors have given this edition a general tune-up of object-oriented programming presentation. The new Fourth Edition has a new code-highlighting style that uses an alternate background color to focus the reader on new code elements in a program. The Deitels' C++ How to Program is the most comprehensive, practical introduction to C++ ever published -- with hundreds of hands-on exercises, roughly 250 complete programs written and documented for easy learning, and exceptional insight into good programming practices, maximizing performance, avoiding errors, debugging, and testing. This new Fourth Edition has an upgraded OOD/UML case to latest UML standard, as well as significant improvements to exception handling and operator overloading chapters. Features enhanced treatment of strings and arrays as objects earlier in the book using standard C++ classes, string and vector. The Fourth Edition retains every key concept and technique ANSI C++ developers need to master: control structures, functions, arrays, pointers and strings, classes and data abstraction, operator overloading, inheritance, virtual functions, polymorphism, I/O, templates, exception handling, file processing, data structures, and more. It also includes a detailed introduction to Standard Template Library (STL) containers, container adapters, algorithms, and iterators. The accompanying CD-ROM includes all the code from the book as well as essential software for learning C++. For anyone who wants to learn C++, improve their existing C++ skills, and master object-oriented development with C++.

Fundamentals of Object-Oriented Design in UML shows aspiring and experienced programmers alike how to apply design concepts, the UML, and the best practices in OO development to improve both their code and their success rates with object-based projects. Essential skills for first-time programmers! This easy-to-use book explains the fundamentals of UML. You'll learn to read, draw, and use this visual modeling language to create clear and effective blueprints for software development projects. The modular approach of this

series--including drills, sample projects, and mastery checks--makes it easy to learn to use this powerful programming language at your own pace.

This multi-function volume starts off as an ideal basic textbook for teaching object modeling, fundamental concepts learning and system designing with thirteen UML diagrams. But it also contains a whole section devoted to advanced research topics, samples and case studies. It is an essential work for any system developer or graduate student in a discipline that requires the power of object modeling as part of a development methodology.

There's no easier, faster, or more practical way to learn the really tough subjects UML Demystified explains how to read, model, and use UML to create well-structured, stable software products. This self-teaching guide comes complete with key points, background information, quizzes at the end of each chapter, and even a final exam. Simple enough for beginners but challenging enough for advanced students, this is a lively and entertaining brush-up, introductory text, or classroom supplement.

Object-Oriented Design with UML and Java provides an integrated introduction to object-oriented design with the Unified Modelling Language (UML) and the Java programming language. The book demonstrates how Java applications, no matter how small, can benefit from some design during their construction. Fully road-tested by students on the authors' own courses, the book shows how these complementary technologies can be used effectively to create quality software. It requires no prior knowledge of object orientation, though readers must have some experience of Java or other high level programming language. This book covers object technology; object-oriented analysis and design; and implementation of objects with Java. It includes two case studies dealing with library applications. The UML has been incorporated into a graphical design tool called ROME, which can be downloaded from the book's website. This object modelling environment allows readers to prepare and edit various UML diagrams. ROME can be used alongside a Java compiler to generate Java code from a UML class diagram then compile and run the resulting application for hands-on learning. This text would be a valuable resource for undergraduate students taking courses on O-O analysis and design, O-O modelling, Java programming, and modelling with UML. * Integrates design and implementation, using Java and UML * Includes case studies and exercises * Bridges the gap between programming texts and high level analysis books on design

With its clear introduction to the Unified Modeling Language (UML) 2.0, this tutorial offers a solid understanding of each topic, covering foundational concepts of object-orientation and an introduction to each of the UML diagram types.

[Copyright: 597cdf202c41b592966f2855d85225f1](#)