

Uml Distilled 4th Edition Pearson

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++.

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)

Written primarily for undergraduates in CIS and MIS programs, 'Essentials of Systems Analysis and Design' is appropriate for SAD courses where a streamlined approach is necessary due to lab assignments, projects, contact time, and/or outside reading requirements.

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

A guide to using UML describes major UML diagrams, their creation, and how to decipher them.

This is the eagerly-anticipated revision to one of the seminal books in the field of software architecture which clearly defines and explains the topic.

The Object-Oriented Thought Process Third Edition Matt Weisfeld An introduction to object-oriented concepts for developers looking to master modern application practices. Object-oriented programming (OOP) is the foundation of modern programming languages, including C++, Java, C#, and Visual Basic .NET. By designing with objects rather than treating the code and data as separate entities, OOP allows objects to fully utilize other objects' services as well as inherit their functionality. OOP promotes code portability and reuse, but requires a shift in thinking to be fully understood. Before jumping into the world of object-oriented programming languages, you must first master The Object-Oriented Thought Process. Written by a developer for developers who want to make the leap to object-oriented technologies as well as managers who simply want to understand what they are managing, The Object-Oriented Thought Process provides a solution-oriented approach to object-oriented programming. Readers will learn to understand object-oriented design with inheritance or composition, object aggregation and association, and the difference between interfaces and implementations. Readers will also become more efficient and better thinkers in terms of object-oriented development. This revised edition focuses on interoperability across various technologies, primarily using XML as the communication mechanism. A more detailed focus is placed on how business objects operate over networks, including client/server architectures and web services. "Programmers who aim to create high quality software—as all programmers should—must learn the varied subtleties of the familiar yet not so familiar beasts called objects and classes. Doing so entails careful study of books such as Matt Weisfeld's The Object-Oriented Thought Process." —Bill McCarty, author of Java Distributed Objects, and Object-Oriented Design in Java Matt Weisfeld is an associate professor in business and technology at Cuyahoga Community College in Cleveland, Ohio. He has more than 20 years of experience as a professional software developer, project manager, and corporate trainer using C++, Smalltalk, .NET, and Java. He holds a BS in systems analysis, an MS in computer science, and an MBA in project management. Weisfeld has published many articles in major computer trade magazines and professional journals.

Information system design and development is of interest and importance to researchers and practitioners, as advances in this discipline impact a number of other related fields and help to guide future research. Theoretical and Practical Advances in Information Systems Development: Emerging Trends and Approaches contains fundamental concepts, emerging theories, and practical applications in database management, systems analysis and design, and software engineering. Contributions present critical findings in information resources management that inform and advance the field.

BeagleBone Black is a low-cost, open hardware computer uniquely suited to interact with sensors and actuators directly and over the Web. Introduced in April 2013 by BeagleBoard.org, a community of developers first established in early 2008, BeagleBone Black is used frequently to build vision-enabled robots, home automation systems, artistic lighting systems, and countless other do-it-yourself and professional projects. BeagleBone variants include the original BeagleBone and the newer BeagleBone Black, both hosting a powerful 32-bit, super-scalar ARM Cortex A8 processor capable of running numerous mobile and desktop-capable operating systems, typically variants of Linux including Debian, Android, and Ubuntu. Yet, BeagleBone is small enough to fit in a small mint tin box. The "Bone" may be used in a wide variety of projects from middle school science fair projects to senior design projects to first prototypes of very complex systems. Novice users may access the power of the Bone through the user-friendly BoneScript software, experienced through a Web browser in most major operating systems, including Microsoft Windows, Apple Mac OS X, or the Linux operating systems. Seasoned users may take full advantage of the Bone's power using the underlying Linux-based operating system, a host of feature extension boards (Capes) and a wide variety of Linux community open source libraries. This book provides an introduction to this powerful computer and has been designed for a wide variety of users including the first time novice through the seasoned embedded system design professional. The book contains background theory on system operation coupled with many well-documented, illustrative examples. Examples for novice users are centered on motivational, fun robot projects while advanced projects follow the theme of assistive technology and image-processing applications.

This volume examines all aspects of using agent or individual-based simulation. This approach represents systems as individual elements having their own set of differing states and internal processes. The interactions between elements in the simulation represent interactions in the target systems. What makes this "social" is that it can represent an observed society. Social systems include all those systems where the components have individual agency but also interact with each other. This includes human societies and groups, but also increasingly socio-technical systems where the internet-based devices form the substrate for interaction. These systems are central to our lives, but are among the most complex known. This poses particular problems for those who wish to understand them. The complexity often makes analytic approaches infeasible but, on the other hand, natural language approaches are also inadequate for relating intricate cause and effect. This is why individual and agent-based computational approaches hold out the possibility of new and deeper understanding of such systems. This handbook marks the maturation of this new field. It brings together summaries of the best thinking and practices in this area from leading researchers in the field and constitutes a reference point for standards against which future methodological advances can be judged. This second edition adds new chapters on different

modelling purposes and applying software engineering methods to simulation development. Revised existing content will keep the book up-to-date with recent developments. This volume will help those new to the field avoid "reinventing the wheel" each time, and give them a solid and wide grounding in the essential issues. It will also help those already in the field by providing accessible overviews of current thought. The material is divided into four sections: Introduction, Methodology, Mechanisms, and Applications. Each chapter starts with a very brief section called 'Why read this chapter?' followed by an abstract, which summarizes the content of the chapter. Each chapter also ends with a section on 'Further Reading'. Whilst sometimes covering technical aspects, this second edition of Simulating Social Complexity is designed to be accessible to a wide range of researchers, including both those from the social sciences as well as those with a more formal background. It will be of use as a standard reference text in the field and also be suitable for graduate level courses.

A new edition of this title is available, ISBN-10: 0672330164 ISBN-13: 9780672330162 The Object-Oriented Thought Process, Second Edition will lay the foundation in object-oriented concepts and then explain how various object technologies are used. Author Matt Weisfeld introduces object-oriented concepts, then covers abstraction, public and private classes, reusing code, and developing frameworks. Later chapters cover building objects that work with XML, databases, and distributed systems (including EJBs, .NET, Web Services and more). Throughout the book Matt uses UML, the standard language for modeling objects, to provide illustration and examples of each concept.

A clear, student-friendly and engaging introduction to how information technology is used in business. Featuring several case studies, video interviews, thorough pedagogy and completely up-to-date chapters, this textbook will be a core resource for undergraduate students of Business Information Systems, a compulsory module in business degrees.

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.

Second Edition of the UML video course based on the book Applying UML and Patterns. This VTC will focus on object-oriented analysis and design, not just drawing UML.

The most comprehensive, up-to-date introduction to the theory and practice of artificial intelligence. The long-anticipated revision of Artificial Intelligence: A Modern Approach explores the full breadth and depth of the field of artificial intelligence (AI). The 4th Edition brings readers up to date on the latest technologies, presents concepts in a more unified manner, and offers new or expanded coverage of machine learning, deep learning, transfer learning, multiagent systems, robotics, natural language processing, causality, probabilistic programming, privacy, fairness, and safe AI.

This book combines elementary theory from computer science with real-world challenges in global geodetic observation, based on examples from the Geodetic Observatory Wettzell, Germany. It starts with a step-by-step introduction to developing stable and safe scientific software to run successful software projects. The use of software toolboxes is another essential aspect that leads to the application of generative programming. An example is a generative network middleware that simplifies communication. One of the book's main focuses is on explaining a potential strategy involving autonomous production cells for space geodetic techniques. The complete software design of a satellite laser ranging system is taken as an example. Such automated systems are then combined for global interaction using secure communication tunnels for remote access. The network of radio telescopes is used as a reference. Combined observatories form coordinated multi-agent systems and offer solutions for operational aspects of the Global Geodetic Observing System (GGOS) with regard to "Industry 4.0".

Larman covers how to investigate requirements, create solutions and then translate designs into code, showing developers how to make practical use of the most significant recent developments. A summary of UML notation is included

The need to handle increasingly larger data volumes is one factor driving the adoption of a new class of nonrelational "NoSQL" databases. Advocates of NoSQL databases claim they can be used to build systems that are more performant, scale better, and are easier to program. NoSQL Distilled is a concise but thorough introduction to this rapidly emerging technology. Pramod J. Sadalage and Martin Fowler explain how NoSQL databases work and the ways that they may be a superior alternative to a traditional RDBMS. The authors provide a fast-paced guide to the concepts you need to know in order to evaluate whether NoSQL databases are right for your needs and, if so, which technologies you should explore further. The first part of the book concentrates on core concepts, including schemaless data models, aggregates, new distribution models, the CAP theorem, and map-reduce. In the second part, the authors explore architectural and design issues associated with implementing NoSQL. They also present realistic use cases that demonstrate NoSQL databases at work and feature representative examples using Riak, MongoDB, Cassandra, and Neo4j. In addition, by drawing on Pramod Sadalage's pioneering work, NoSQL Distilled shows how to implement evolutionary design with schema migration: an essential technique for applying NoSQL databases. The book concludes by describing how NoSQL is ushering in a new age of Polyglot Persistence, where multiple data-storage worlds coexist, and architects can choose the technology best optimized for each type of data access.

The book has been written according to the syllabus prescribed by the Directorate General of Employment and Training for the Craftsman Training Scheme and the Apprenticeship Training Scheme for the Electrical Trades (Electrician, Wireman and Lineman). The first volume covers what should be taught in the first year. The language is very simple and the concepts are explained with the help of clear illustrations. The theory is supported by practical applications of the concepts. A number of solved examples have been provided. At each chapter end is a set of unsolved numerical problems and review questions. Answers to these have been provided. These review questions are taken from the examination papers of the National Council for Vocational trades and from the All India Skill Competitions. This book will help trainees and apprentices prepare themselves for the final examination and for the job interviews. Key features Software estimation, software quality, software project management, risk management, COCOMO II model covered in detail. Discussions on software engineering tools, user interface issues, ISO 9001, and CMM. Cases and Term Projects. A case for study and analysis with questions for discussion related to the topics learnt at the end of each part. An integrated solution to the case using both the approaches-System and Object-Oriented-given at the end of the text. Three cases are given at the end of Part V, for the students to analyze and submit as term project. Joined-up healthcare makes information available when and where it is needed to improve safety, efficiency and effectiveness. Politicians may take interoperability between healthcare computer systems for granted, but it is non-trivial. Healthcare integration projects are notoriously under-estimated and come in over-budget and over-time. Joined-up healthcare depends on

standards. The two leading standards are the SNOMED CT, which is a clinical terminology (semantics) and HL7 Version 3, which is a specialised healthcare interoperability language (syntax). Both are new, complex and fit for purpose. Tim Benson believes there is an unmet need for a book on Healthcare Integration. Some health informatics textbooks include chapters on HL7 and/or SNOMED, but these are usually quite short and cannot provide even an adequate introduction. There is little of much value on the Internet, or in journals or conference proceedings. This guide will help readers learn how to employ the significant power of use cases to their software development efforts. It provides a practical methodology, presenting key use case concepts.

For nearly ten years, the Unified Modeling Language (UML) has been the industry standard for visualizing, specifying, constructing, and documenting the artifacts of a software-intensive system. As the de facto standard modeling language, the UML facilitates communication and reduces confusion among project stakeholders. The recent standardization of UML 2.0 has further extended the language's scope and viability. Its inherent expressiveness allows users to model everything from enterprise information systems and distributed Web-based applications to real-time embedded systems. In this eagerly anticipated revision of the best-selling and definitive guide to the use of the UML, the creators of the language provide a tutorial to its core aspects in a two-color format designed to facilitate learning. Starting with an overview of the UML, the book explains the language gradually by introducing a few concepts and notations in each chapter. It also illustrates the application of the UML to complex modeling problems across a variety of application domains. The in-depth coverage and example-driven approach that made the first edition of *The Unified Modeling Language User Guide* an indispensable resource remain unchanged. However, content has been thoroughly updated to reflect changes to notation and usage required by UML 2.0. Highlights include: A new chapter on components and internal structure, including significant new capabilities for building encapsulated designs New details and updated coverage of provided and required interfaces, collaborations, and UML profiles Additions and changes to discussions of sequence diagrams, activity diagrams, and more Coverage of many other changes introduced by the UML 2.0 specification With this essential guide, you will quickly get up to speed on the latest features of the industry standard modeling language and be able to apply them to your next software project.

Tackling globalization is a great challenge – it is both extremely beneficial and essentially problematic. This comprehensive, multidisciplinary study confronts this ambivalence through the use of computer simulation. It discusses the findings of social interaction and social simulation through the use of understandable global examples. Readers can use this book as a tool to outline significant aspects of intercultural simulation and highlight the issues that need to be considered in the reader's analysis. The author leads the reader via sequential narration from a colloquial description of intercultural situations to final simulation prototypes; each step is accompanied by descriptive comments and program code. *Social Interaction, Globalization and Computer-aided Analysis* shows the reader how to acquire intercultural data from seemingly inconceivable information sources. Researchers and software developers engaged in interdisciplinary research projects in the field of Human-Computer Interaction will find this book to be a useful companion in their work. Alexander Osherenko is the founder of the start-up company Socioware Development, which implements psychologically-, sociologically- and culturally-aware software that scrutinizes information based on the findings of the cognitive sciences. Solutions created by Socioware Development can be implemented across a vast spectrum of industries, including car manufacturing, insurance and banking, Internet search engines and e-retailers.

"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

Users can dramatically improve the design, performance, and manageability of object-oriented code without altering its interfaces or behavior. "Refactoring" shows users exactly how to spot the best opportunities for refactoring and exactly how to do it, step by step.

New technologies are revolutionising the way manufacturing and supply chain management are implemented. These changes are delivering manufacturing firms the competitive advantage of a highly flexible and responsive supply chain and manufacturing system to ensure that they meet the high expectations of their customers, who, in today's economy, demand absolutely the best service, price, delivery time and product quality. To make e-manufacturing and supply chain technologies effective, integration is needed between various, often disparate systems. To understand why this is such an issue, one needs to understand what the different systems or system components do, their objectives, their specific focus areas and how they interact with other systems. It is also required to understand how these systems evolved to their current state, as the concepts used during the early development of systems and technology tend to remain in place throughout the life-cycle of the systems/technology. This book explores various standards, concepts and techniques used over the years to model systems and hierarchies in order to understand where they fit into the organization and supply chain. It looks at the specific system components and the ways in which they can be designed and graphically depicted for easy understanding by both information technology (IT) and non-IT personnel. Without a good implementation philosophy, very few systems add any real benefit to an organization, and for this reason the ways in which systems are implemented and installation projects managed are also explored and recommendations are made as to possible methods that have proven successful in the past. The human factor and how that impacts on system success are also addressed, as is the motivation for system

investment and subsequent benefit measurement processes. Finally, the vendor/user supply/demand within the e-manufacturing domain is explored and a method is put forward that enables the reduction of vendor bias during the vendor selection process. The objective of this book is to provide the reader with a good understanding regarding the four critical factors (business/physical processes, systems supporting the processes, company personnel and company/personal performance measures) that influence the success of any e-manufacturing implementation, and the synchronization required between these factors. · Discover how to implement the flexible and responsive supply chain and manufacturing execution systems required for competitive and customer-focused manufacturing · Build a working knowledge of the latest plant automation, manufacturing execution systems (MES) and supply chain management (SCM) design techniques · Gain a fuller understanding of the four critical factors (business and physical processes, systems supporting the processes, company personnel, performance measurement) that influence the success of any e-manufacturing implementation, and how to evaluate and optimize all four factors

The professional programmer's Deitel® guide to Java® 9 and the powerful Java platform Written for programmers with a background in another high-level language, this book applies the Deitel signature live-code approach to teaching programming and explores the Java® 9 language and APIs in depth. The book presents concepts in fully tested programs, complete with code walkthroughs, syntax shading, code highlighting and program outputs. It features hundreds of complete Java 9 programs with thousands of lines of proven code, and hundreds of software-development tips that will help you build robust applications. Start with an introduction to Java using an early classes and objects approach, then rapidly move on to more advanced topics, including JavaFX GUI, graphics, animation and video, exception handling, lambdas, streams, functional interfaces, object serialization, concurrency, generics, generic collections, database with JDBC™ and JPA, and compelling new Java 9 features, such as the Java Platform Module System, interactive Java with JShell (for discovery, experimentation and rapid prototyping) and more. You'll enjoy the Deitels' classic treatment of object-oriented programming and the object-oriented design ATM case study, including a complete Java implementation. When you're finished, you'll have everything you need to build industrial-strength, object-oriented Java 9 applications. New Java® 9 Features Java® 9's Platform Module System Interactive Java via JShell—Java 9's REPL Collection Factory Methods, Matcher Methods, Stream Methods, JavaFX Updates, Using Modules in JShell, Completable Future Updates, Security Enhancements, Private Interface Methods and many other language and API updates. Core Java Features Classes, Objects, Encapsulation, Inheritance, Polymorphism, Interfaces Composition vs. Inheritance, "Programming to an Interface not an Implementation" Lambdas, Sequential and Parallel Streams, Functional Interfaces with Default and Static Methods, Immutability JavaFX GUI, 2D and 3D Graphics, Animation, Video, CSS, Scene Builder Files, I/O Streams, XML Serialization Concurrency for Optimal Multi-Core Performance, JavaFX Concurrency APIs Generics and Generic Collections Recursion, Database (JDBC™ and JPA) Keep in Touch Contact the authors at: deitel@deitel.com Join the Deitel social media communities LinkedIn® at bit.ly/DeitelLinkedIn Facebook® at facebook.com/DeitelFan Twitter® at twitter.com/deitel YouTube™ at youtube.com/DeitelTV Subscribe to the Deitel @ Buzz e-mail newsletter at www.deitel.com/newsletter/subscribe.html For source code and updates, visit: www.deitel.com/books/Java9FP

For a second microprocessor course for students enrolled in Electrical/Computer Engineering Microcontroller courses. Designed for a senior- or graduate-level embedded systems design course, Embedded Systems Design and Applications with the 68HC12 introduces readers to unique issues associated with designing, testing, integrating, and implementing microcontroller/microprocessor-based embedded systems.

For courses in Software Engineering, Software Development, or Object-Oriented Design and Analysis at the Junior/Senior or Graduate level. This text can also be utilized in short technical courses or in short, intensive management courses. Shows students how to use both the principles of software engineering and the practices of various object-oriented tools, processes, and products. Using a step-by-step case study to illustrate the concepts and topics in each chapter, Bruegge and Dutoit emphasize learning object-oriented software engineer through practical experience: students can apply the techniques learned in class by implementing a real-world software project. The third edition addresses new trends, in particular agile project management (Chapter 14 Project Management) and agile methodologies (Chapter 16 Methodologies).

Artificial Intelligence: A Modern Approach offers the most comprehensive, up-to-date introduction to the theory and practice of artificial intelligence. Number one in its field, this textbook is ideal for one or two-semester, undergraduate or graduate-level courses in Artificial Intelligence.

Widely considered one of the best practical guides to programming, Steve McConnell's original CODE COMPLETE has been helping developers write better software for more than a decade. Now this classic book has been fully updated and revised with leading-edge practices—and hundreds of new code samples—illustrating the art and science of software construction. Capturing the body of knowledge available from research, academia, and everyday commercial practice, McConnell synthesizes the most effective techniques and must-know principles into clear, pragmatic guidance. No matter what your experience level, development environment, or project size, this book will inform and stimulate your thinking—and help you build the highest quality code. Discover the timeless techniques and strategies that help you: Design for minimum complexity and maximum creativity Reap the benefits of collaborative development Apply defensive programming techniques to reduce and flush out errors Exploit opportunities to refactor—or evolve—code, and do it safely Use construction practices that are right-weight for your project Debug problems quickly and effectively Resolve critical construction issues early and correctly Build quality into the beginning, middle, and end of your project

This book constitutes the refereed proceedings of the 29th International Conference on Advanced Information Systems Engineering, CAiSE 2017, held in Essen, Germany, in June 2017. The 37 papers presented together with 3 keynote papers in this volume were carefully reviewed and selected from 175 submissions. The papers are organized in topical sections on information systems architecture; business process alignment; user knowledge discovery; business process performance; big data exploration; process variability management; information systems transformation and evolution; business process modeling readability; business process adaption; data mining; process discovery; business process modeling notation.

This title provides a forum where expert insights are presented on the subject of linking three current phenomena: software evolution, UML and XML.

In recent years, our world has experienced a profound shift and progression in available computing and knowledge sharing innovations. These emerging advancements have developed at a rapid pace, disseminating into and affecting numerous aspects of contemporary society. This has created a pivotal need for an innovative compendium encompassing the latest trends, concepts, and issues surrounding this relevant discipline area. During the past 15 years, the Encyclopedia of Information Science and Technology has become recognized as one of the landmark sources of the latest knowledge and discoveries in this discipline. The Encyclopedia of Information Science and Technology, Fourth Edition is a 10-volume set which includes 705 original and previously unpublished research articles covering a full range of perspectives, applications, and techniques contributed by thousands of experts and researchers from around the globe. This authoritative encyclopedia is an all-encompassing, well-established reference source that is ideally designed to disseminate the most forward-thinking and diverse research findings. With critical perspectives on the impact of information science management and new technologies in modern settings, including but not limited to computer science, education, healthcare, government, engineering, business, and natural and physical sciences, it is a pivotal and relevant source

of knowledge that will benefit every professional within the field of information science and technology and is an invaluable addition to every academic and corporate library.

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.

The Systems Modeling Language (SysML) extends UML with powerful systems engineering capabilities for modeling a wider spectrum of systems and capturing all aspects of a system's design. SysML Distilled is the first clear, concise guide for everyone who wants to start creating effective SysML models. (Drawing on his pioneering experience at Lockheed Martin and NASA, Lenny Delligatti illuminates SysML's core components and provides practical advice to help you create good models and good designs. Delligatti begins with an easy-to-understand overview of Model-Based Systems Engineering (MBSE) and an explanation of how SysML enables effective system specification, analysis, design, optimization, verification, and validation. Next, he shows how to use all nine types of SysML diagrams, even if you have no previous experience with modeling languages. A case study running through the text demonstrates the use of SysML in modeling a complex, real-world sociotechnical system. Modeled after Martin Fowler's classic UML Distilled, Delligatti's indispensable guide quickly teaches you what you need to know to get started and helps you deepen your knowledge incrementally as the need arises. Like SysML itself, the book is method independent and is designed to support whatever processes, procedures, and tools you already use. Coverage Includes Why SysML was created and the business case for using it Quickly putting SysML to practical use What to know before you start a SysML modeling project Essential concepts that apply to all SysML diagrams SysML diagram elements and relationships Diagramming block definitions, internal structures, use cases, activities, interactions, state machines, constraints, requirements, and packages Using allocations to define mappings among elements across a model SysML notation tables, version changes, and sources for more information

The practice of enterprise application development has benefited from the emergence of many new enabling technologies. Multi-tiered object-oriented platforms, such as Java and .NET, have become commonplace. These new tools and technologies are capable of building powerful applications, but they are not easily implemented. Common failures in enterprise applications often occur because their developers do not understand the architectural lessons that experienced object developers have learned. Patterns of Enterprise Application Architecture is written in direct response to the stiff challenges that face enterprise application developers. The author, noted object-oriented designer Martin Fowler, noticed that despite changes in technology--from Smalltalk to CORBA to Java to .NET--the same basic design ideas can be adapted and applied to solve common problems. With the help of an expert group of contributors, Martin distills over forty recurring solutions into patterns. The result is an indispensable handbook of solutions that are applicable to any enterprise application platform. This book is actually two books in one. The first section is a short tutorial on developing enterprise applications, which you can read from start to finish to understand the scope of the book's lessons. The next section, the bulk of the book, is a detailed reference to the patterns themselves. Each pattern provides usage and implementation information, as well as detailed code examples in Java or C#. The entire book is also richly illustrated with UML diagrams to further explain the concepts. Armed with this book, you will have the knowledge necessary to make important architectural decisions about building an enterprise application and the proven patterns for use when building them. The topics covered include · Dividing an enterprise application into layers · The major approaches to organizing business logic · An in-depth treatment of mapping between objects and relational databases · Using Model-View-Controller to organize a Web presentation · Handling concurrency for data that spans multiple transactions · Designing distributed object interfaces

"Designing Software Product Lines with UML is well-written, informative, and addresses a very important topic. It is a valuable contribution to the literature in this area, and offers practical guidance for software architects and engineers." --Alan Brown Distinguished Engineer, Rational Software, IBM Software Group "Gomaa's process and UML extensions allow development teams to focus on feature-oriented development and provide a basis for improving the level of reuse across multiple software development efforts. This book will be valuable to any software development professional who needs to manage across projects and wants to focus on creating software that is consistent, reusable, and modular in nature." --Jeffrey S Hammond Group Marketing Manager, Rational Software, IBM Software Group "This book brings together a good range of concepts for understanding software product lines and provides an organized method for developing product lines using object-oriented techniques with the UML. Once again, Hassan has done an excellent job in balancing the needs of both experienced and novice software engineers." --Robert G. Pettit IV, Ph.D. Adjunct Professor of Software Engineering, George Mason University "This breakthrough book provides a comprehensive step-by-step approach on how to develop software product lines, which is of great strategic benefit to industry. The development of software product lines enables significant reuse of software architectures. Practitioners will benefit from the well-defined PLUS process and rich case studies." --Hurley V. Blankenship II Program Manager, Justice and Public Safety, Science Applications International Corporation "The Product Line UML based Software engineering (PLUS) is leading edge. With the author's wide experience and deep knowledge, PLUS is well harmonized with architectural and design pattern technologies." --Michael Shin Assistant Professor, Texas Tech University Long a standard practice in traditional manufacturing, the concept of product lines is quickly earning recognition in the software industry. A software product line is a family of systems that shares a common set of core technical assets with preplanned extensions and variations to address the needs of specific customers or market segments. When skillfully implemented, a product line strategy can yield enormous gains in productivity, quality, and time-to-market. Studies indicate that if three or more systems with a degree of common functionality are to be developed, a product-line approach is significantly more cost-effective. To model and design families of systems, the analysis and design concepts for single product systems need to be extended to support product lines. Designing Software Product Lines with UML shows how to employ the latest version of the industry-standard Unified Modeling Language (UML 2.0) to reuse software requirements and architectures rather than starting the development of each new system from scratch. Through real-world case studies, the book illustrates the fundamental concepts and technologies used in the design and implementation of software product lines. This book describes a new UML-based software design method for product lines called PLUS (Product Line UML-based Software engineering). PLUS provides a set of concepts and techniques to extend UML-based design methods and processes for single systems in a new dimension to address software product lines. Using PLUS, the objective is to explicitly model the commonality and variability in a software product line. Hassan Gomaa explores how each of the UML modeling views--use case, static, state machine, and interaction modeling--can be extended to address software product families. He also discusses how software architectural patterns can be used to develop a reusable component-based architecture for a product line and how to express this architecture as a UML platform-independent model that can then be mapped to a platform-specific model. Key topics include: Software product line engineering process, which extends the Unified Development Software Process to address software product lines Use case modeling, including modeling the common and variable functionality of a product line Incorporating feature modeling into UML for modeling common, optional, and alternative product line features Static modeling, including modeling the boundary of the product line and information-intensive entity classes Dynamic modeling, including using interaction modeling to address use-case variability State machines for modeling state-dependent variability Modeling class variability using inheritance and parameterization Software architectural patterns for product lines Component-based distributed design using the new UML 2.0 capability for modeling components, connectors, ports, and provided and required interfaces Detailed case studies giving a step-by-step solution to real-world product line problems Designing Software Product Lines with UML is an invaluable resource for all designers and developers in this growing field. The information, technology, and case studies presented here show how to harness the promise of software product lines and the practicality of

the UML to take software design, quality, and efficiency to the next level. An enhanced online index allows readers to quickly and easily search the entire text for specific topics.
[Copyright: 570d3d0102ab01ab2236b36fad3e5bf8](#)