

# Elegant Objects Volume 1

If you ally habit such a referred **Elegant Objects Volume 1** books that will manage to pay for you worth, acquire the utterly best seller from us currently from several preferred authors. If you want to hilarious books, lots of novels, tale, jokes, and more fictions collections are plus launched, from best seller to one of the most current released.

You may not be perplexed to enjoy every ebook collections Elegant Objects Volume 1 that we will agreed offer. It is not in relation to the costs. Its more or less what you habit currently. This Elegant Objects Volume 1 , as one of the most energetic sellers here will certainly be in the course of the best options to review.

[An Introduction to Object-Oriented Programming with Java 1.5 Update with OLC Bi-Card](#) - C. Thomas Wu 2004

An Introduction to Object-Oriented Programming with Java provides an accessible and thorough introduction to the basics of programming in java. This much-anticipated revision continues its emphasis on object-oriented programming. Objects are used early so students begin thinking in an object-oriented way, then later Wu teaches students to define their own classes. In the third edition, the author has eliminated the author-written classes, so students get accustomed to using the standard java libraries. In the new update, the author has included the Scanner Class for input, a new feature of Java 1.5. Also new is the use of smaller complete code examples to enhance student learning. The larger sample development programs are continued in this edition, giving students an opportunity to walk incrementally walk through program design, learning the fundamentals of software engineering. The number and variety of examples makes this a student-friendly text that teaches by showing. Object diagrams continue to be an important element of Wu's approach. The consistent, visual approach assists students in understanding concepts.

**JavaScript: The Good Parts** - Douglas Crockford 2008-05-08

Most programming languages contain good and bad parts, but JavaScript has more than its share of the bad, having been developed and released in a hurry before it could be refined. This authoritative book scrapes away these bad features to reveal a subset of JavaScript that's more reliable, readable, and maintainable than the language as a whole—a subset you can use to create truly extensible and efficient code. Considered the JavaScript expert by many people in the development community, author Douglas Crockford identifies the abundance of good ideas that make JavaScript an outstanding object-oriented programming language—ideas such as functions, loose typing, dynamic objects, and an expressive object literal notation. Unfortunately, these good ideas are mixed in with bad and downright awful ideas, like a programming model based on global variables. When Java applets failed, JavaScript became the language of the Web by default, making its popularity almost completely independent of its qualities as a programming language. In JavaScript: The Good Parts, Crockford finally digs through the steaming pile of good intentions and blunders to give you a detailed look at all the genuinely elegant parts of JavaScript, including: Syntax Objects Functions Inheritance Arrays Regular expressions Methods Style Beautiful features The real beauty? As you move ahead with the subset of JavaScript that this book presents, you'll also sidestep the need to unlearn all the bad parts. Of course, if you want to find out more about the bad parts and how to use them badly, simply consult any other JavaScript book. With JavaScript: The Good Parts, you'll discover a beautiful, elegant, lightweight and highly expressive language that lets you create effective code, whether you're managing object libraries or just trying to get Ajax to run fast. If you develop sites or applications for the Web, this book is an absolute must.

**Code Ahead** - Yegor Bugayenko 2018-07-06

TL;DR It's a semi-autobiographical fiction book about a software architect who is involved in programming, debugging, releasing, testing, organizing, team work, and management issues.

**Precious Objects** - Alicia Oltuski 2011-07-19

In the middle of New York City lies a neighborhood where all secrets are valuable, all assets are liquid, and all deals are sealed with a blessing rather than a contract. Welcome to the diamond district. Ninety percent of all diamonds that enter America pass through these few blocks, but the inner workings of this mysterious

world are known only to the people who inhabit it. In Precious Objects, twenty-six-year-old journalist Alicia Oltuski, the daughter and granddaughter of diamond dealers, seamlessly blends family narrative with literary reportage to reveal the fascinating secrets of the diamond industry and its madcap characters: an Elvis-impersonating dealer, a duo of diamond-detective brothers, and her own eccentric father. With insight and drama, Oltuski limns her family's diamond-paved move from communist Siberia to a displaced persons camp in post-World War II Germany to New York's diamond district, exploring the connections among Jews and the industry, the gem and its lore, and the exotic citizens of this secluded world. Entertaining and illuminating, Precious Objects offers an insider's look at the history, business, and society behind one of the world's most coveted natural resources, providing an unforgettable backstage pass to an extraordinary and timeless show.

*PHP in Action* - Marcus Baker 2007-06-30

To keep programming productive and enjoyable, state-of-the-art practices and principles are essential. Object-oriented programming and design help manage complexity by keeping components cleanly separated. Unit testing helps prevent endless, exhausting debugging sessions. Refactoring keeps code simple and readable. PHP offers all this—and more. PHP in Action shows you how to apply PHP techniques and principles to all the most common challenges of web programming, including: Web presentation and templates User interaction including the Model-View-Controller architecture Input validation and form handling Database connection and querying and abstraction Object persistence Purchase of the print book comes with an offer of a free PDF, ePub, and Kindle eBook from Manning. Also available is all code from the book.

**Elegant Objects** - Yegor Bugayenko 2016-02-17

TL;DR There are 23 practical recommendations for object-oriented programmers. Most of them are completely against everything you've read in other books. For example, static methods, NULL references, getters, setters, and mutable classes are called evil.

*The Object-Oriented Thought Process* - Matt Weisfeld 2008-08-25

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.

[APPLYING UML & PATTERNS 3RD EDITION](#) - Craig Larman 2015

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

*Python 3 Object-oriented Programming* - Dusty Phillips 2015-08-20

Unleash the power of Python 3 objects About This Book Stop writing scripts and start architecting programs Learn the latest Python syntax and libraries A practical, hands-on tutorial that teaches you all about abstract design patterns and how to implement them in Python 3 Who This Book Is For If you're new to object-oriented programming techniques, or if you have basic Python skills and wish to learn in depth how and when to correctly apply object-oriented programming in Python to design software, this is the book for you. What You Will Learn Implement objects in Python by creating classes and defining methods Separate related objects into a taxonomy of classes and describe the properties and behaviors of those objects via the class interface Extend class functionality using inheritance Understand when to use object-oriented features, and more importantly when not to use them Discover what design patterns are and why they are different in Python Uncover the simplicity of unit testing and why it's so important in Python Grasp common concurrency techniques and pitfalls in Python 3 Exploit object-oriented programming in key Python technologies such as Kivy and Django. Object-oriented programming concurrently with asyncio In Detail Python 3 is more versatile and easier to use than ever. It runs on all major platforms in a huge array of use cases. Coding in Python minimizes development time and increases productivity in comparison to other languages. Clean, maintainable code is easy to both read and write using Python's clear, concise syntax. Object-oriented programming is a popular design paradigm in which data and behaviors are encapsulated in such a way that they can be manipulated together. Many modern programming languages utilize the powerful concepts behind object-oriented programming and Python is no exception. Starting with a detailed analysis of object-oriented analysis and design, you will use the Python programming language to clearly grasp key concepts from the object-oriented paradigm. This book fully explains classes, data encapsulation, inheritance, polymorphism, abstraction, and exceptions with an emphasis on when you can use each principle to develop well-designed software. You'll get an in-depth analysis of many common object-oriented design patterns that are more suitable to Python's unique style. This book will not just teach Python syntax, but will also build your confidence in how to program. You will also learn how to create maintainable applications by studying higher level design patterns. Following this, you'll learn the complexities of string and file manipulation, and how Python distinguishes between binary and textual data. Not one, but two very powerful automated testing systems will be introduced in the book. After you discover the joy of unit testing and just how easy it can be, you'll study higher level libraries such as database connectors and GUI toolkits and learn how they uniquely apply object-oriented principles. You'll learn how these principles will allow you to make greater use of key members of the Python eco-system such as Django and Kivy. This new edition includes all the topics that made *Python 3 Object-oriented Programming* an instant Packt classic. It's also packed with updated content to reflect recent changes in the core Python library and covers modern third-party packages that were not available on the Python 3 platform when the book was first published. Style and approach Throughout the book you will learn key object-oriented programming techniques demonstrated by comprehensive case studies in the context of a larger project.

*The Syntax of Objects* - Tim McCreight 2009

By what code do objects connect with us, embrace us, refute us, and in the end, inform us? From this question, the author reaches into her personal life to search for those universal moments where the teacups and trinkets of our lives linger.

**Forecasting: principles and practice** - Rob J Hyndman 2018-05-08

Forecasting is required in many situations. Stocking an inventory may require forecasts of demand months in advance. Telecommunication routing requires traffic forecasts a few minutes ahead. Whatever the circumstances or time horizons involved, forecasting is an important aid in effective and efficient planning. This textbook provides a comprehensive introduction to forecasting methods and presents enough information about each method for readers to use them sensibly.

[Apprenticeship Patterns](#) - Dave Hoover 2009-10-02

Are you doing all you can to further your career as a software developer? With today's rapidly changing and ever-expanding technologies, being successful requires more than technical expertise. To grow professionally, you also need soft skills and effective learning techniques. Honing those skills is what this book is all about. Authors Dave Hoover and Adewale Oshineye have cataloged dozens of behavior patterns to help you perfect essential aspects of your craft. Compiled from years of research, many interviews, and feedback from O'Reilly's online forum, these patterns address difficult situations that programmers, administrators, and DBAs face every day. And it's not just about financial success. *Apprenticeship Patterns* also approaches software development as a means to personal fulfillment. Discover how this book can help you make the best of both your life and your career. Solutions to some common obstacles that this book explores in-depth include: Burned out at work? "Nurture Your Passion" by finding a pet project to rediscover the joy of problem solving. Feeling overwhelmed by new information? Re-explore familiar territory by building something you've built before, then use "Retreat into Competence" to move forward again. Stuck in your learning? Seek a team of experienced and talented developers with whom you can "Be the Worst" for a while. "Brilliant stuff! Reading this book was like being in a time machine that pulled me back to those key learning moments in my career as a professional software developer and, instead of having to learn best practices the hard way, I had a guru sitting on my shoulder guiding me every step towards master craftsmanship. I'll certainly be recommending this book to clients. I wish I had this book 14 years ago!"-Russ Miles, CEO, OpenCredo

[A Primer on Scientific Programming with Python](#) - Hans Petter Langtangen 2016-07-28

The book serves as a first introduction to computer programming of scientific applications, using the high-level Python language. The exposition is example and problem-oriented, where the applications are taken from mathematics, numerical calculus, statistics, physics, biology and finance. The book teaches "Matlab-style" and procedural programming as well as object-oriented programming. High school mathematics is a required background and it is advantageous to study classical and numerical one-variable calculus in parallel with reading this book. Besides learning how to program computers, the reader will also learn how to solve mathematical problems, arising in various branches of science and engineering, with the aid of numerical methods and programming. By blending programming, mathematics and scientific applications, the book lays a solid foundation for practicing computational science. From the reviews: Langtangen ... does an excellent job of introducing programming as a set of skills in problem solving. He guides the reader into thinking properly about producing program logic and data structures for modeling real-world problems using objects and functions and embracing the object-oriented paradigm. ... Summing Up: Highly recommended. F. H. Wild III, Choice, Vol. 47 (8), April 2010 Those of us who have learned scientific programming in Python 'on the streets' could be a little jealous of students who have the opportunity to take a course out of Langtangen's Primer." John D. Cook, The Mathematical Association of America, September 2011 This book goes through Python in particular, and programming in general, via tasks that scientists will likely perform. It contains valuable information for students new to scientific computing and would be the perfect bridge between an introduction to programming and an advanced course on numerical methods or computational science. Alex Small, IEEE, CiSE Vol. 14 (2), March /April 2012 "This fourth edition is a wonderful, inclusive textbook that covers pretty much everything one needs to know to go from zero to fairly sophisticated scientific programming in Python..." Joan Horvath, Computing Reviews, March 2015

[Elegant Objects](#) - Yegor Bugayenko 2017-04-18

TL;DR Compound variable names, validators, private static literals, configurable objects, inheritance, annotations, MVC, dependency injection containers, reflection, ORM and even algorithms are our enemies.

[Object-Process Methodology](#) - Dov Dori 2011-06-27

Object-Process Methodology (OPM) is an intuitive approach to systems engineering. This book presents the theory and practice of OPM with examples from various industry segments and engineering disciplines, as well as daily life. OPM is a generic, domain independent approach that is applicable almost anywhere in systems engineering.

**Crave** - Tracy Wolff 2020-04-07

The #1 New York Times Bestselling Series An Amazon Best YA Book of 2020 Glitter Magazine's #1 Pick for Best YA of 2020 Optioned for Film by Universal My whole world changed when I stepped inside the academy. Nothing is right about this place or the other students in it. Here I am, a mere mortal among gods...or monsters. I still can't decide which of these warring factions I belong to, if I belong at all. I only know the one thing that unites them is their hatred of me. Then there's Jaxon Vega. A vampire with deadly secrets who hasn't felt anything for a hundred years. But there's something about him that calls to me, something broken in him that somehow fits with what's broken in me. Which could spell death for us all. Because Jaxon walled himself off for a reason. And now someone wants to wake a sleeping monster, and I'm wondering if I was brought here intentionally—as the bait. \*\*\*INCLUDES 3 BONUS SCENES FROM THE HERO'S POV\*\*\* Don't miss a single book in the series that spawned a phenomenon! The Crave series is best enjoyed in order: Crave Crush Covet Court Charm Cherish

[Data Structures and Algorithms in Python](#) - Michael T. Goodrich 2013-03-08

Based on the authors' market leading data structures books in Java and C++, this textbook offers a comprehensive, definitive introduction to data structures in Python by authoritative authors. Data Structures and Algorithms in Python is the first authoritative object-oriented book available for the Python data structures course. Designed to provide a comprehensive introduction to data structures and algorithms, including their design, analysis, and implementation, the text will maintain the same general structure as Data Structures and Algorithms in Java and Data Structures and Algorithms in C++.

**Think Julia** - Ben Lauwens 2019-04-05

If you're just learning how to program, Julia is an excellent JIT-compiled, dynamically typed language with a clean syntax. This hands-on guide uses Julia 1.0 to walk you through programming one step at a time, beginning with basic programming concepts before moving on to more advanced capabilities, such as creating new types and multiple dispatch. Designed from the beginning for high performance, Julia is a general-purpose language ideal for not only numerical analysis and computational science but also web programming and scripting. Through exercises in each chapter, you'll try out programming concepts as you learn them. Think Julia is perfect for students at the high school or college level as well as self-learners and professionals who need to learn programming basics. Start with the basics, including language syntax and semantics Get a clear definition of each programming concept Learn about values, variables, statements, functions, and data structures in a logical progression Discover how to work with files and databases Understand types, methods, and multiple dispatch Use debugging techniques to fix syntax, runtime, and semantic errors Explore interface design and data structures through case studies

[Object Thinking](#) - David West 2004

Object Thinking blends historical perspective, experience, and visionary insight - exploring how developers can work less like the computers they program and more like problem solvers.

[Data Visualization](#) - Kieran Healy 2018-12-18

An accessible primer on how to create effective graphics from data This book provides students and researchers a hands-on introduction to the principles and practice of data visualization. It explains what makes some graphs succeed while others fail, how to make high-quality figures from data using powerful and reproducible methods, and how to think about data visualization in an honest and effective way. Data Visualization builds the reader's expertise in ggplot2, a versatile visualization library for the R programming language. Through a series of worked examples, this accessible primer then demonstrates how to create plots piece by piece, beginning with summaries of single variables and moving on to more

complex graphics. Topics include plotting continuous and categorical variables; layering information on graphics; producing effective "small multiple" plots; grouping, summarizing, and transforming data for plotting; creating maps; working with the output of statistical models; and refining plots to make them more comprehensible. Effective graphics are essential to communicating ideas and a great way to better understand data. This book provides the practical skills students and practitioners need to visualize quantitative data and get the most out of their research findings. Provides hands-on instruction using R and ggplot2 Shows how the "tidyverse" of data analysis tools makes working with R easier and more consistent Includes a library of data sets, code, and functions

[Practical Object-oriented Design in Ruby](#) - Sandi Metz 2013

The Complete Guide to Writing More Maintainable, Manageable, Pleasing, and Powerful Ruby Applications Ruby's widely admired ease of use has a downside: Too many Ruby and Rails applications have been created without concern for their long-term maintenance or evolution. The Web is awash in Ruby code that is now virtually impossible to change or extend. This text helps you solve that problem by using powerful real-world object-oriented design techniques, which it thoroughly explains using simple and practical Ruby examples. This book focuses squarely on object-oriented Ruby application design. Practical Object-Oriented Design in Ruby will guide you to superior outcomes, whatever your previous Ruby experience. Novice Ruby programmers will find specific rules to live by; intermediate Ruby programmers will find valuable principles they can flexibly interpret and apply; and advanced Ruby programmers will find a common language they can use to lead development and guide their colleagues. This guide will help you Understand how object-oriented programming can help you craft Ruby code that is easier to maintain and upgrade Decide what belongs in a single Ruby class Avoid entangling objects that should be kept separate Define flexible interfaces among objects Reduce programming overhead costs with duck typing Successfully apply inheritance Build objects via composition Design cost-effective tests Solve common problems associated with poorly designed Ruby code

[The Book in History, the Book as History](#) - Heidi Brayman 2016-01-01

The essays in this collection reach beyond book history to address fundamental questions about historicism with a broad range of issues such as gender and sexuality, religion, political theory, economic history, adaptation and appropriation, and quantitative analysis and digital humanities.

**Junior Objects** - Yegor Bugayenko 2020-06-21

TL;DR Object-oriented programming is a mess. However, if you want to find a good job, you have to use it. This book will help you learn it, from scratch. Also, it will point you to all necessary topics in software engineering that you need to learn before you call yourself a senior software developer.

[C# 2.0](#) - Michel de Champlain 2005-03-30

You don't need coddling; you don't need to be told what you already know. What you need is a book that uses your experience as a Java or C++ programmer to give you a leg up into the challenges and rewards of C#. And this Practical Guide is precisely what you're after. Written by a team that boasts extensive experience teaching C# to professionals, this book provides a practical, efficient explanation of the language itself, covering basic to advanced features and calling out all that's new in 2.0. Its instruction is always firmly situated within the context of the .NET framework and bolstered by code examples, key lessons in object-oriented programming, and installments of a realistic application programming tutorial. Concise and incisive, this is the best way to master the world's fastest-growing and most marketable programming language. Features: Provides a carefully focused explanation of every aspect of the C# language, including entire chapters on the unified type system, advanced types, collections, generics, reflection and attributes. Highlights all features new to the latest version of C# and organizes its presentation of C# according to the key principles of object-oriented programming and the .NET framework. Using end-of-chapter exercises, incrementally develops a cohesive application programming tutorial. Provides a carefully focused explanation of every aspect of the C# language, including entire chapters on the unified type system, advanced types, collections, generics, reflection and attributes. Highlights all features new to the latest version of C# and organizes its presentation of C# according to the key principles of object-oriented programming and the .NET framework. Using end-of-chapter exercises, incrementally develops a cohesive application programming tutorial.

**Elegant Eating** - Philippa Glanville 2002

International in outlook, *Elegant Eating* combines a rich array of individual items used to dress the table, many from the unrivalled collections of the V&A, and authentic historical settings to give them context.

[Learning JavaScript Design Patterns](#) - Addy Osmani 2012-07-08

With *Learning JavaScript Design Patterns*, you'll learn how to write beautiful, structured, and maintainable JavaScript by applying classical and modern design patterns to the language. If you want to keep your code efficient, more manageable, and up-to-date with the latest best practices, this book is for you. Explore many popular design patterns, including Modules, Observers, Facades, and Mediators. Learn how modern architectural patterns—such as MVC, MVP, and MVVM—are useful from the perspective of a modern web application developer. This book also walks experienced JavaScript developers through modern module formats, how to namespace code effectively, and other essential topics. Learn the structure of design patterns and how they are written. Understand different pattern categories, including creational, structural, and behavioral. Walk through more than 20 classical and modern design patterns in JavaScript. Use several options for writing modular code—including the Module pattern, Asynchronous Module Definition (AMD), and CommonJS. Discover design patterns implemented in the jQuery library. Learn popular design patterns for writing maintainable jQuery plug-ins. "This book should be in every JavaScript developer's hands. It's the go-to book on JavaScript patterns that will be read and referenced many times in the future."—Andrée Hansson, Lead Front-End Developer, *prsis!*

[The Eye of the World](#) - Robert Jordan 1990-01-15

Relates a tale of the bestial Trollocs, the witch Moiraine, and three boys, one of whom is fated to become the Dragon—the World's only hope and the sure means of its destruction.

**Ordinary Objects** - Amie Lynn Thomasson 2010

'Ordinary Objects' shows how to develop a common-sense ontology and defend it against a variety of eliminativist arguments. The text argues that the apparently diverse eliminativist arguments rest on a few shared assumptions, and that questioning these gives us reason to reevaluate the proper methods and limits of metaphysics.

[Blindsight](#) - Peter Watts 2006-10-03

*Blindsight* is the Hugo Award-nominated novel by Peter Watts, "a hard science fiction writer through and through and one of the very best alive" (*The Globe and Mail*). Two months have past since a myriad of alien objects clenched about the Earth, screaming as they burned. The heavens have been silent since—until a derelict space probe hears whispers from a distant comet. Something talks out there: but not to us. Who should we send to meet the alien, when the alien doesn't want to meet? Send a linguist with multiple-personality disorder and a biologist so spliced with machinery that he can't feel his own flesh. Send a pacifist warrior and a vampire recalled from the grave by the voodoo of paleogenetics. Send a man with half his mind gone since childhood. Send them to the edge of the solar system, praying you can trust such freaks and monsters with the fate of a world. You fear they may be more alien than the thing they've been sent to find—but you'd give anything for that to be true, if you knew what was waiting for them. . . . At the Publisher's request, this title is being sold without Digital Rights Management Software (DRM) applied.

[Pragmatic Thinking and Learning](#) - Andy Hunt 2008-10-28

Printed in full color. Software development happens in your head. Not in an editor, IDE, or design tool. You're well educated on how to work with software and hardware, but what about wetware—our own brains? Learning new skills and new technology is critical to your career, and it's all in your head. In this book by Andy Hunt, you'll learn how our brains are wired, and how to take advantage of your brain's architecture. You'll learn new tricks and tips to learn more, faster, and retain more of what you learn. You need a pragmatic approach to thinking and learning. You need to Refactor Your Wetware. Programmers have to learn constantly; not just the stereotypical new technologies, but also the problem domain of the application, the whims of the user community, the quirks of your teammates, the shifting sands of the industry, and the evolving characteristics of the project itself as it is built. We'll journey together through bits of cognitive and neuroscience, learning and behavioral theory. You'll see some surprising aspects of how our brains work, and how you can take advantage of the system to improve your own learning and thinking skills. In this book you'll learn how to: Use the Dreyfus Model of Skill Acquisition to become more

expert. Leverage the architecture of the brain to strengthen different thinking modes. Avoid common "known bugs" in your mind. Learn more deliberately and more effectively. Manage knowledge more efficiently.

[Elegant Debts](#) - Craig Clunas 2004-06-30

This book takes an innovative approach to one of the great figures of Chinese culture, the writer and painter Wen Zhengming (1470–1559). Renowned as one of the great "scholar painters" of the Ming dynasty, Wen was enmeshed in a complex web of social obligations, his "elegant debts" as he called them, which led to many of his most celebrated works. Using an unprecedented quantity of primary sources for his life and work, *Elegant Debts* looks at the ways in which social obligation and gift exchange were central to personal and individual identity in the Ming period. The book also examines Wen's family relationships, his friends, mentors, and pupils, his sense of a distinct local identity, and the interplay of national and regional politics with the achievements of his long life. It uses the insights of a range of scholarship—art history, social and literary history, and anthropology—to show how "self" was constructed in Ming China. In doing so, it makes a major contribution toward a more diverse art history that is less dependent on European conceptions of artists and their work. Craig Clunas has published extensively in the field, and is widely recognized as one of the world's leading scholars of Ming culture. Featuring many images of the work of one of China's major painters, this book is accessible to all who are interested in China's culture and history, as well as to students and scholars of art history and the history of culture.

[Objects of Desire](#) - Clare Sestanovich 2021-06-29

"A debut story collection of the rarest kind ... you wish that every single entry could be an entire novel." —*Entertainment Weekly Fresh*, intimate stories of women's lives from an extraordinary new literary voice, laying bare the unexpected beauty and irony in contemporary life. A college freshman, traveling home, strikes up an odd, ephemeral friendship with the couple next to her on the plane. A mother prepares for her son's wedding, her own life unraveling as his comes together. A long-lost stepbrother's visit to New York prompts a family's reckoning with its old taboos. A wife considers the secrets her marriage once contained. An office worker, exhausted by the ambitions of the men around her, emerges into a gridlocked city one afternoon to make a decision. In these eleven powerful stories, thrilling desire and melancholic yearning animate women's lives, from the brink of adulthood to the labyrinthine path between twenty and thirty, to middle age, when certain possibilities quietly elapse. Tender, lucid, and piercingly funny, *Objects of Desire* is a collection pulsing with subtle drama, rich with unforgettable scenes, and alive with moments of recognition each more startling than the last—a spellbinding debut that announces a major talent.

[Object-Oriented Analysis and Design](#) - Sarnath Ramnath 2010-12-06

Object-oriented analysis and design (OOAD) has over the years, become a vast field, encompassing such diverse topics as design process and principles, documentation tools, refactoring, and design and architectural patterns. For most students the learning experience is incomplete without implementation. This new textbook provides a comprehensive introduction to OOAD. The salient points of its coverage are: • A sound footing on object-oriented concepts such as classes, objects, interfaces, inheritance, polymorphism, dynamic linking, etc. • A good introduction to the stage of requirements analysis. • Use of UML to document user requirements and design. • An extensive treatment of the design process. • Coverage of implementation issues. • Appropriate use of design and architectural patterns. • Introduction to the art and craft of refactoring. • Pointers to resources that further the reader's knowledge. All the main case-studies used for this book have been implemented by the authors using Java. The text is liberally peppered with snippets of code, which are short and fairly self-explanatory and easy to read. Familiarity with a Java-like syntax and a broad understanding of the structure of Java would be helpful in using the book to its full potential.

**You Don't Know JS: Scope & Closures** - Kyle Simpson 2014-03-10

No matter how much experience you have with JavaScript, odds are you don't fully understand the language. This concise yet in-depth guide takes you inside scope and closures, two core concepts you need to know to become a more efficient and effective JavaScript programmer. You'll learn how and why they work, and how an understanding of closures can be a powerful part of your development skillset. Like other books in the "You Don't Know JS" series, *Scope and Closures* dives into trickier parts of the language that many JavaScript programmers simply avoid. Armed with this knowledge, you can achieve true JavaScript

mastery. Learn about scope, a set of rules to help JavaScript engines locate variables in your code Go deeper into nested scope, a series of containers for variables and functions Explore function- and block-based scope, “hoisting”, and the patterns and benefits of scope-based hiding Discover how to use closures for synchronous and asynchronous tasks, including the creation of JavaScript libraries

**Eloquent JavaScript** - Marijn Haverbeke 2011-01-15

JavaScript is at the heart of almost every modern Web application, whether it's Google Apps, Twitter, or the newest browser-based game. Though it's simple for beginners to pick up and play with, JavaScript is not a toy—it's a flexible and complex language that can be used to build full-scale applications. Eloquent JavaScript dives into this flourishing language and teaches you to write code that's beautiful and effective. By immersing you in example code and encouraging experimentation right from the start, the author quickly gives you the tools you need to build your own programs. As you follow along with examples like an artificial life simulation and a version of the classic game Sokoban, you'll learn to: -Understand the essential elements of programming: syntax, control, and data -Use object-oriented and functional programming techniques to organize and clarify your programs -Script the browser and make basic Web applications -Work with tools like regular expressions and XMLHttpRequest objects And since programming is an art that's best learned by doing, all example code is available online in an interactive sandbox for you to experiment with. With Eloquent JavaScript as your guide, you can tweak, expand, and modify the author's code, or throw it away and build your own creations from scratch. Before you know it, you'll be fluent in the language of the Web.

**Human Dimension and Interior Space** - Julius Panero 2014-01-21

The study of human body measurements on a comparative basis is known as anthropometrics. Its applicability to the design process is seen in the physical fit, or interface, between the human body and the various components of interior space. Human Dimension and Interior Space is the first major anthropometrically based reference book of design standards for use by all those involved with the physical planning and detailing of interiors, including interior designers, architects, furniture designers, builders, industrial designers, and students of design. The use of anthropometric data, although no substitute for good design or sound professional judgment should be viewed as one of the many tools required in the design process. This comprehensive overview of anthropometrics consists of three parts. The first part deals with the theory and application of anthropometrics and includes a special section dealing with physically disabled and elderly people. It provides the designer with the fundamentals of anthropometrics

and a basic understanding of how interior design standards are established. The second part contains easy-to-read, illustrated anthropometric tables, which provide the most current data available on human body size, organized by age and percentile groupings. Also included is data relative to the range of joint motion and body sizes of children. The third part contains hundreds of dimensioned drawings, illustrating in plan and section the proper anthropometrically based relationship between user and space. The types of spaces range from residential and commercial to recreational and institutional, and all dimensions include metric conversions. In the Epilogue, the authors challenge the interior design profession, the building industry, and the furniture manufacturer to seriously explore the problem of adjustability in design. They expose the fallacy of designing to accommodate the so-called average man, who, in fact, does not exist. Using government data, including studies prepared by Dr. Howard Stoudt, Dr. Albert Damon, and Dr. Ross McFarland, formerly of the Harvard School of Public Health, and Jean Roberts of the U.S. Public Health Service, Panero and Zelnik have devised a system of interior design reference standards, easily understood through a series of charts and situation drawings. With Human Dimension and Interior Space, these standards are now accessible to all designers of interior environments.

*Design Patterns* - Erich Gamma 1995

Software -- Software Engineering.

The Elegant Universe - Brian Greene 2000

Introduces the superstring theory that attempts to unite general relativity and quantum mechanics

**Elegant Anatomy** - Marieke M.A. Hendriksen 2015-01-27

In Elegant Anatomy Marieke Hendriksen offers an account of the material culture of the eighteenth-century Leiden anatomical collections, which have not been studied in detail before. Starting from the materiality of preparations, it introduces the novel analytical concept of aesthesis.

**Data Structures and Algorithms in Java** - Michael T. Goodrich 2014-01-28

The design and analysis of efficient data structures has long been recognized as a key component of the Computer Science curriculum. Goodrich, Tomassia and Goldwasser's approach to this classic topic is based on the object-oriented paradigm as the framework of choice for the design of data structures. For each ADT presented in the text, the authors provide an associated Java interface. Concrete data structures realizing the ADTs are provided as Java classes implementing the interfaces. The Java code implementing fundamental data structures in this book is organized in a single Java package, net.datastructures. This package forms a coherent library of data structures and algorithms in Java specifically designed for educational purposes in a way that is complimentary with the Java Collections Framework.