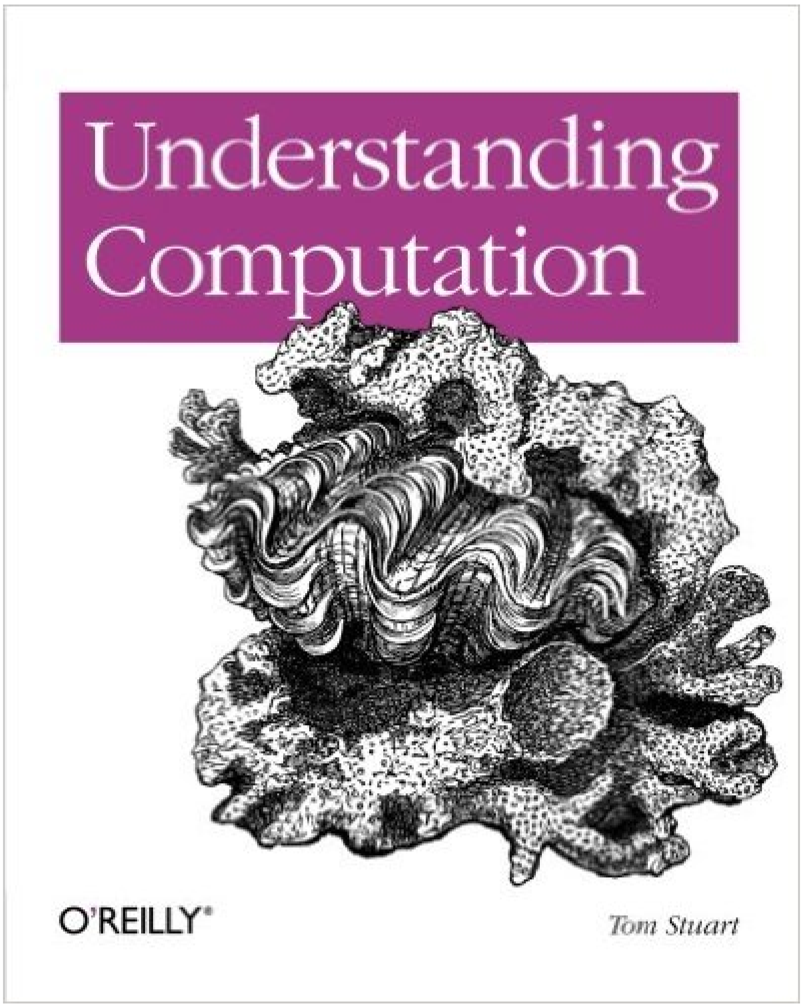
Understanding Computation: From Simple Machines To Impossible Programs



Synopsis

Finally, you can learn computation theory and programming language design in an engaging, practical way. Understanding Computation explains theoretical computer science in a context youâ ™ll recognize, helping you appreciate why these ideas matter and how they can inform your day-to-day programming.Rather than use mathematical notation or an unfamiliar academic programming language like Haskell or Lisp, this book uses Ruby in a reductionist manner to present formal semantics, automata theory, and functional programming with the lambda calculus. Itâ ™s ideal for programmers versed in modern languages, with little or no formal training in computer science.Understand fundamental computing concepts, such as Turing completeness in languagesDiscover how programs use dynamic semantics to communicate ideas to machinesExplore what a computer can do when reduced to its bare essentialsLearn how universal Turing machines led to todayâ ™s general-purpose computersPerform complex calculations, using simple languages and cellular automataDetermine which programming language features are essential for computationExamine how halting and self-referencing make some computing problems unsolvableAnalyze programs by using abstract interpretation and type systems

Book Information

Paperback: 332 pages
Publisher: O’Reilly Media; 1 edition (June 3, 2013)
Language: English
ISBN-10: 1449329276
ISBN-13: 978-1449329273
Product Dimensions: 7 x 0.8 x 9.2 inches
Shipping Weight: 1 pounds (View shipping rates and policies)
Average Customer Review: 3.8 out of 5 starsÂ Â See all reviewsÂ (12 customer reviews)
Best Sellers Rank: #524,636 in Books (See Top 100 in Books) #166 inÂ Books > Computers & Technology > Computer Science > Information Theory #643 inÂ Books > Textbooks > Computer Science > Software Design & Engineering #1425 inÂ Books > Computers & Technology > Programming > Software Design, Testing & Engineering > Software Development

Customer Reviews

Despite the fact that there’s no _real_ reason to be apologetic, I also haven’t yet reached the point in my career as a software developer where I’ve stopped apologizing for the fact that I have no "real" Computer Science Background. And/but that’s exactly what draws me to books like

"Understanding Computation" by Tom Stuart (O’Reilly, 2013). Stuart describes the books as for:"...programmers who are curious about programming languages and the theory of computation, especially those who don’t have a formal background in mathematics or computer science."In other words, people like me. The people that Eric Miraglia described as the "liberal arts majors drafted into web-developer service during the dotcom boom". (source: Professional JavaScript for Web Developers 2nd ed.) Yes: the liberal artsy non-computer science degree holders that wound up doing computer sciencey type software work just the same. Smart people that nevertheless are exploring some of these concepts for the first time.For a taste of what I mean, observe the following quote:"In the end, syntax is only concerned with the surface appearance of programs, not with their meanings."If that made you smile just a little bit, because you want to peel the onion layers away and get at the semantic questions underneath... then this book is for you.Now before we go any further -- a couple words on what this book is **not**. This is _not_ a book about software engineering. "Big O notation" does not make an appearance here in the text, not once.

Download to continue reading...

Understanding Computation: From Simple Machines to Impossible Programs Algorithms + Data Structures = Programs (Prentice-Hall Series in Automatic Computation) Vintage Coca-cola Machines a Price and Identification Guide to Collectible Coolers and Machines Slot Machines and Coin-Op Games: A Collector’s Guide to One-Armed Bandits and Amusement Machines Effective TCP/IP Programming: 44 Tips to Improve Your Network Programs: 44 Tips to Improve Your Network Programs Insider’s Guide to Graduate Programs in Clinical and Counseling Psychology (Insider’s Guide to Graduate Programs in Clinical & Counseling Psychology) Graduate Programs in Business, Education, Information Studies, Law & Social Work 2017 (Peterson’s Graduate Programs in Business, Education, Health, Information Studies, Law and Social Work) Brooks/Cole Empowerment Series: Social Welfare Policy and Social Programs (SW 323K Social Welfare Programs, Policies, and Issues) Understanding Bergson, Understanding Modernism (Understanding Philosophy, Understanding Modernism) Introduction to Computation and Programming Using Python: With Application to Understanding Data (MIT Press) Simple Machines Quantum Computing: A Gentle Introduction (Scientific and Engineering Computation) Quantum Computation with Topological Codes: From Qubit to Topological Fault-Tolerance (SpringerBriefs in Mathematical Physics) Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation) Evolutionary Computation 1: Basic Algorithms and Operators Evolutionary Computation in Gene Regulatory Network Research (Wiley Series in Bioinformatics) Evolutionary Computation in Bioinformatics (The Morgan Kaufmann Series in Artificial Intelligence) Modern

Fortran Explained (Numerical Mathematics and Scientific Computation) The Calculus of Computation: Decision Procedures with Applications to Verification Linear Genetic Programming (Genetic and Evolutionary Computation)