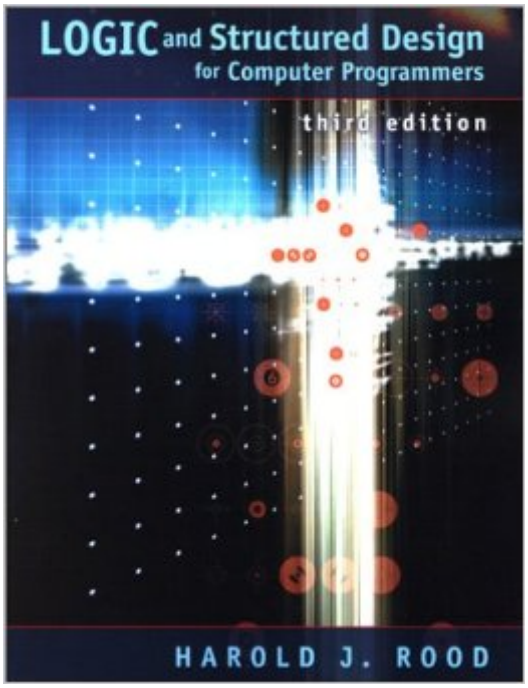
Logic And Structured Design For Computer Programmers



Synopsis

LOGIC AND STRUCTURED DESIGN is an introduction to the logic of data processing. It is intended for those who plan, but have not yet begun, to study programming, particularly those with little background in mathematics or logic. The author avoids reference to specific programming languages, isolating questions of logic from questions of syntax. This approach enables readers to concentrate on the logic of problems. The book walks readers through logical problems common to a variety of programming languages and provides the background in logic that many programming texts and courses assume.

Book Information

Paperback: 464 pages

Publisher: Course Technology; 3 edition (December 26, 2000)

Language: English

ISBN-10: 0534373860

ISBN-13: 978-0534373863

Product Dimensions: 10.9 x 8.5 x 0.8 inches

Shipping Weight: 2.5 pounds

Average Customer Review: 4.6 out of 5 starsÂ Â See all reviewsÂ (5 customer reviews)

Best Sellers Rank: #1,765,806 in Books (See Top 100 in Books) #58 inÂ Books > Engineering & Transportation > Engineering > Electrical & Electronics > Circuits > Logic #269 inÂ Books > Computers & Technology > Programming > Software Design, Testing & Engineering > Structured Design #286 inÂ Books > Computers & Technology > Programming > Software Design, Testing & Engineering > Logic

Customer Reviews

Simple Question: How and why can this book be relevant at this writing (in 2016) given UML? Simple answer: the purpose of UML is to achieve a level of systems and program design abstraction that manages complexity, yet maintains enough precision to be machine readable. The old fashioned diagrams were thus like pseudocode vs. actual code, giving a representative, simplified way to understand what lies beneath the code, and why, while maintaining enough formality to avoid "custom" interpretations of the diagrams. This book is at the programming level, not the meta level of UML 2, for example, for systems rather than code design, or for portable code generation and transportability, so you WILL need to graduate to UML at some point.So, think of this wonderful volume as a "prelude to UML" if you will-- exercising your brain with page after page that relates

pseudocode to many logic diagrams, generally independent of language. In fact, UML IS independent of language, although most of us who teach programming think of it in terms of our favorite code and languages. But both this fine volume and UML can both be used with Julia, Java, R, or Haskell-- do it all the time. Of course UML was designed by the OOP community so does favor its modularity (as in Java/C#), but we work in functional languages all the time, and use both UML and this book without problems or misunderstandings.I teach both undergrad and grad programming, including assembly and circuit level sims for robotics and HPC, as well as the design of software proof assistants based on Topology. This won't show you how the tabs work in UML (for one of the best books I use for that, for .