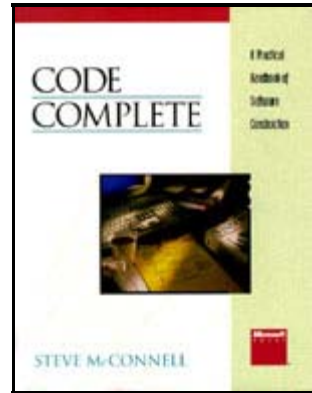


Off The Shelf: Resources For The Programmer's Library

By Jim Douglas

Code Complete, by Steve McConnell, was written for the working programmer. It provides an extraordinary amount of detail about practical programming issues with suggestions on how to evolve a design from Program Design Language (Structured English) to finished code, discussions about how to organize your code to improve readability and reliability, guidelines for naming variables, and strategies and techniques for tuning code. Unlike most current books about software construction, which are often directed exclusively toward C programmers, the 500+ examples of good and bad code scattered throughout the book are written in a number of different languages, including Pascal, C/C++, Ada, BASIC, FORTRAN, and COBOL. As he notes in the Preface, McConnell has "avoided esoteric language features except where they're specifically discussed...in order to make the multiple-language burden as light as possible." As a result, the Business Basic programmer should have no difficulty in following any of these examples.



According to McConnell, the substance of **Code Complete** was distilled from "dozens of software engineering books and...hundreds of journal articles." Using the detailed citations that appear throughout the book, I searched for some of these items on the web. While several of the books are currently out of print and would therefore be difficult to track down, I found that McConnell had accurately summarized the key points of the articles I was able to locate. In doing so, McConnell keeps the material covered in the book relevant to the professional programmer in the 1990s. For example, the original 1975 paper on Program Design Language (www1.cfg.com/pdl81/pdlpaper.html) would be considered archaic by a modern reader, as it goes to a lot of trouble to justify the concept of structured programming, which we accept as a given today. By eliminating unnecessary background information that is no longer directly applicable to the main thrust of an article, McConnell makes the lessons taught by these kinds of materials easier to learn.

McConnell's style is relaxed and extremely readable; it's never a burden on the reader. For the busy programmer who might be reluctant to read an 850-page book all at once, each of the thirty-three chapters can be read as an individual essay. Every chapter starts with its own table of contents, followed by cross-references to related information. Most chapters end with suggestions for further reading, and all chapters end with a summary of their key points.

Throughout the book, margin notes and icons ("Key Point," "Further Reading," "Hard Data," "Cross Reference," and "Coding Horror") nicely complement the main text.

I can't recommend this book highly enough. If you read only one book on software development this year, **Code Complete** should be the one.

Code Complete: A Practical Handbook of Software Construction, Steve McConnell, Microsoft Press, 1993, ISBN 1-55615-484-4, 857 pages, US\$ 35. Available from [Amazon.com](https://www.amazon.com) for US\$ 28 plus shipping. See www.construx.com/stevemcc/cc.htm for a link to Amazon.com, a detailed table of contents, excerpts (including the Preface, "Valid Reasons to Create a Routine," "Using gotos," "Finding an Error," "Introduction to Code Tuning," and "The Library of a Software Professional"), reviews, and a link to the Errata page at www.microsoft.com.