

The Techie's Guide to C Programming

Episode the third, in which Max O'blivion pauses on his quest for the lost function long enough for a quick read.

STEVE RIMMER

This month we're going to have a quick look at C language books. There are quite a lot of them lurking about, and this includes a few pretty dreadful ones. As helpful as I've tried to make this series of articles, a few reference books will do you enormous good. They're great when you're uncertain about syntactical questions concerning C, and, if you have enough of them, you can drop them on cats. This rarely harms the books, but it'll keep the cats off your keyboard in the future.

These books can be found at your local book store. Computers are very trendy, and just about every book store on the planet has a shelf devoted especially to them. If you buy your books at a store wherein the cashiers are particularly stunned, you can often get away with telling them that the prices are in hexadecimal — being computer books — and translate them into the appropriate, lower, decimal prices.

Dead Trees

The quintessential C manual is *The C Programming Language* by Brian W. Kernighan and Dennis M. Ritchie, published by Prentice-Hall. These are the two characters originally responsible for the creation of C. This book is often referred to simply as "Kernighan and Ritchie", inasmuch as all C books have pretty similar titles and this sound slicker, like, maybe you knew the dudes personally, y'know.

Being quintessential and all, Kernighan and Ritchie is a difficult read. You should bear in mind that C was originally developed for mainframe computers, and that this book was written before the PC was even a gleam in anyone's eye. It is, however, the authoritative reference for the language, and everyone ought to own a copy. It has recently been revised.

Somewhat less quintessential, to the

point of obscurity, *C Primer Plus* by Mitchell Waite, Stephen Prata and Donald Martin, published by Sams, is a lot easier to get through, as well as being thicker and more devastating to felines. It's intended for use with PC implementations of C, and it gets into many hardware specific areas of writing specifically for the PC. It's user friendly to the point of being user obsequious, but after a few nights of head banging with C you'll probably need a friend. This is a beginner's book.

For serious C programmers, *The C Toolbox* by William James Hunt, published by Addison-Wesley, will get you into writing code that no one will ever dare even snigger at. It's full of things like sorting algorithms, BTREE code and so on. It's a bit advanced and daunting when you first get started, but it's a good effort later on. This is an experienced programmer's book.

Variations In C by Steve Schustack, published by Microsoft, is very much into Microsoft's compilers. It seems very much like a book written by someone who had a lot of C functions lying about and nothing to do with them. Most of the examples in it are intensely specific, such that nothing in the book is likely to be of any use as it stands. This by no means invalidates *Variations In C*... the approaches are sound and you'll learn a lot from seeing how things are done. The purple type gets to your eyes after a time. This is an intermediate book.

A somewhat less intense work, *Proficient C* by Augie Hansen, also published by Microsoft, is a fairly gentle introduction to C. It uses comprehensible example programs, is readable and clear and doesn't attempt to stun you with the author's blinding insights into the zen of programming. It's a good beginner's book.

One of my favourite books for flinging at cats, *C Programmer's Library* by Jack J. Purdum, Timothy C. Leslie and Alan L.

Stegemoller, published by Que, is serious to the point of polyester. It gets into some very specific C code which elegantly solves all sorts of problems you'll probably never come across. It illustrates things like complex device and file control fairly well. I suspect that the examples in it came from a mini or mainframe environment — they're very non-specific, and ignore some of the cruel realities of programming on the PC. This is an experienced programmer's book.

If you can't find Kernighan and Ritchie, a book called *C: A Reference Manual* by Samuel P. Harbison and Guy L. Steele Jr., published by Prentice Hall, is almost as good. It's considerably expanded, with more verbose explanations and examples, and it covers a number of recent changes to C in greater detail. I still prefer Kernighan and Ritchie — once you get into C, it's brevity, like that of the language itself, is easier on your brain. This is a beginner's book.

Finally, *The C Primer* by Les Hancock and Morris Krieger, published by McGraw Hill, is a thin, non-descript little C manual that wouldn't even wound a kitten... much. It's a fairly painless introduction to the language. It doesn't go very far, but it also doesn't drive you into carburetor repair as an alternative to programming. If you don't have to pay a lot for it, it's not a bad beginner's book.

Book of Dreams

This is a tiny fraction of the C books that exist, of course. I've omitted about three times as many that I've read and trashed, and probably three times as many again that I haven't checked out. Once all the cats are dead, there's no way to bring them back for more.

If you seriously want to get into C, I'd suggest popping for at least a few of these. There's nothing like knowing what you're doing to make a job go quicker. ■