

Tutorial: Programming in C

Chapter 1, 6/27/93

I will be teaching the C language by way of examples and explanation. I encourage folks to comment on (and correct) my work. Important corrections and selected comments will make their way into the archived versions of the tutorial.

When the essential features of the language have been covered, I'll begin to branch out by presenting chapters on various programming techniques as expressed in C. The C++ language is a potential future topic, but naturally I can't cover every possible subject at once.

I'll post chapters as often as is practical given my available time. (Keep in mind that I'm being paid the fabulous sum of zero dollars and zero cents for my services before complaining about the amount of time between chapters, folks!)

(The tutorial will be archived for anonymous FTP access on isis.cshl.org in the subdirectory `pub/ctutor`. If you miss an installment or want to catch up, you can retrieve the tutorial from there. If you need help using FTP, ask your system administrator, or read the frequently asked questions list for `comp.newusers.questions`, which is recommended reading for everyone on the net! You can find it in `news.answers`, which has the FAQ lists for nearly every newsgroup. If you CANNOT use FTP, send mail to me at boutell@cshl.org with a specific request. If the volume of such requests is high, I will attempt to put together a mailing list server to cope with them.)

I will be teaching ANSI standard C, with notes on how to make the programs work on a non-ANSI compiler.

In order to keep up with the tutorial without reading all of `comp.lang.c`, just search for messages with "TUTORIAL" (capitalized) in the "Subject" line. (In `rn`, this is done using the `/` key.) If the volume of discussion becomes truly spectacular, I'll seek the creation of new newsgroups, but after considering the possibilities (such as an `alt.uu` Usenet University group) I've determined that the best thing for now is to start out in the mainstream of `comp.lang.c` and migrate when and if there's a demand to do so.

To follow along with the tutorial, you will need the following:

1. A reasonable working knowledge of the system you're using. It is NOT the goal of this tutorial to teach Unix or VMS or MSDOS or MacOS or what have you! I will give a quick explanation of the commands to be used to compile programs under Unix; for personal computers, the commands are usually self-evident, as most C programming software for PCs provides a mouse-driven interface.

2. A C compiler.

"A what?" A C compiler is a program which translates your C programs into the native language of the computer. FOR UNIX: you probably already have one. Type `cc` at the `%` prompt. If the `%` prompt comes back with no intervening message, you have a compiler accessible to you. If you get a message to the effect of "command not found", you may not have a compiler; try `gcc` instead. If you get a different error message complaining that you ran `cc` without giving it any work to do, that's OK. If neither `cc` nor `gcc` works, talk to your local system guru. FOR MSDOS (IBM COMPATIBLES): There are several good products available. Microsoft Quick C and Borland Turbo C/C++ are both good products for beginners; you can use more sophisticated C compilers from these companies if you wish. Power C from Mix Software is cheaper than either, but has its own tricks and traps, but for the price (dirt cheap!) you can't beat it. Don't ask me for ordering information -- pick up any issue of Computer Shopper and look for yourself. FOR THE MACINTOSH: The Mac isn't a strong personal area of expertise for me, so I'll appreciate

suggestions to fill this out, but products I'm aware of include Think C from Symantec and Zortech C/C++. FOR VMS: You probably have one, and it may well be called cc, but I am not and never have been a VMS programmer, so ask your system administrator for assistance on this! (I'd appreciate suggestions here as well.)

3. RECOMMENDED: a copy of Kernighan and Ritchie's "The C Programming Language," second edition. This is not a tutorial, but rather an excellent reference guide to the C language as well as an opportunity to learn more about the language from its creators, Brian Kernighan and Dennis Ritchie.

4. An editor: for MSDOS (IBM compatible) and Macintosh systems this is no problem as an editing environment usually comes with the compiler. For Unix/VMS/what have you, you'll need to be familiar with vi, emacs or another text editor in which to compose your programs. Unix users who post to USENET news are already familiar with vi in most cases, since it's the editor in which messages are composed. To edit a file in vi (for those who have only seen it when posting to news or editing mail) use the command "vi filename" (where filename is the name of the file you will be editing) at the "%" prompt of Unix.

... AND NOW WE BEGIN

Once you have the items listed above, proceed to the next chapter of the tutorial, in which I will present the first program.