Tutorial: Programming in C
Copyright 1993, By Thomas Boutell
Corrections and Additions Through Chapter 4, 7/13/93
NOTE: Previous chapters are available by anonymous ftp from isis.cshl.org in the directory pub/tutorial. Look there first before you ask me to send copies!
This special note is intended to cover a variety of errors and omissions in the first four chapters that have been pointed out to me. I have attempted to give credit; my apologies in advance if I have failed to do so in any case. Note that while I give credit for corrections below, I am not presenting direct quotes unless specified, so don't blame others for my opinions. Where several people submitted a correction, I have generally credited the first person to do so.
I have left out certain corrections/ additions which, while I agree with them, are not critical at this stage and will be taken into account in a more thorough revision.
CHAPTER 1:
I indicated that those needing assistance with basic Unix tasks like using FTP should read comp.newusers.questions. That group name is incorrect; it should have been news.newusers.questions. There is also a news.announce.newusers group. (Lars Wirzenius)
In my list of MSDOS (IBM compatible) compilers, I neglected to mention the free gcc port, djgpp. While it's free, it's still not necessarily a good choice for beginners-- a 386 is ABSOLUTELY required, and the debugger is paleolithic. (Lars Wirzenius)
I said Kernighan & Ritchie's text is not a tutorial. This was utterly incorrect and I apologize for it. (Lars Wirzenius)
I refer to the "%" prompt of Unix; there are other popular Unix shells with other prompts, so I should not have been so specific. (J J Farrell)
CHAPTER 2
I have consistently placed the conditional expression in for statements in parentheses. While this is not "wrong," it is not necessary, either. I was taught this habit as an undergraduate and it has stuck with me. "for (x = 0; x < 10; x++)" is just as good as "for (x=0; (x < 10); x++)". (J J Farrell)
I have referred to "\n" as a carriage return. It isn't; it's a new line. \r is a carriage return. (Most operating systems do a carriage return upon a new line. My code is correct, but my discussion is somewhat misleading.) (J J Farrell)
I have been provided with a list of compilers for Amiga users to complement my notes for other machines: SAS C, which is a commercial product, DICE, which is a shareware product, and a GCC port. The GCC port unfortunately requires 5mb of RAM. (Ross Smith)
I have been provided with notes on compiling and running programs under VMS. The compile, link and run sequence for the interest program is: (Vera Noest)
$ cc interest $ link interest, rtl.opt/opt $ run interest
CHAPTER 3
I referred to compilers which set uninitialized variables to zero as "faulty." They're not, since the C standard says the values of variables that haven't yet been initialized are undefined; 0 is a legitimate value to set them to. I personally feel that providing this "feature" is a mistake, however, since it works against the writing of portable code. (Jutta Degener)
CHAPTER 4
I was provided with a shuffling routine superior to my own in that it arrives at a perfect distribution and is just about as simple as the original: (wilson@blaze.cs.jhu.edu)

for (i = DECK_SIZE - 1; i > 0; i--) { cpos = rand() % (i + 1); c = deck[i]; deck[i] = deck[cpos]; deck[cpos] = c; }
I stated that variables must have constant initializers. Specifically I said that:
main() { int x = 2; }
is OK, but that:
main() { int y = 2; int x = y; }
is not OK. I was completely and utterly wrong on this point. The latter program is completely legitimate. (I know there are further caveats to be discussed when global variables and static variables come into the picture, but I haven't introduced them yet in any case.) (wilson@blaze.cs.jhu.edu)
I was provided with a solution to the problem of output scrolling under VMS. Run the program this way, assuming bj is the program name:
$ define/user sys$output bj.out ! redirect output to file bj.out $ run bj ! no output will be visible on screen $ type/page bj.out ! look at bj.out, 1 page at a time ! this works just like |more
This will dump the output a page at a time. (Vera Noest)
I need to be more specific about ranges of numbers; not "between 0 and 51" but "from 0 to 51 inclusive", for instance. (Mark Brader)
I have stated that when an array is passed to a function, it behaves in exactly the same manner in the called function as it does in the function in which it is declared. This isn't quite true, and I'll have to attend to it in a later chapter. It's a useful simplication, however. (Mark Brader)
I stated that the time() function returns the number of seconds that have elapsed since January 1st, 1970. That's not universally true. It returns a numeric value representing the time, but to arrive at typical representations of time one must use the rest of the time.h values to break down the result. (Mark Brader)
... Thanks, folks. Chapter 5 will follow shortly.
-T