

# An Introduction to C

*As encountered by Michael Roth*

What does C bring to mind? In my particular case it brings to mind the image of a giraffe wading through a large vat of chocolate pudding. However, almost everything brings that to mind... So clearly a more quantifiably significant approach must be taken to describe and explore various attributes and facets of "C".

C is a programming language. That is the primary function of C. Its secondary functions are to begin various words (such as cloud, crustacean, cactus, and kettle) and large scale temperature regulation (more on that later). One of the primary attributes of C is that it allows you to do anything you want, however stupid or misguided that may be. Once an aspiring C programmer, with some clever use of the preprocessor, changed the fundamental laws of physics in such a way that no unified theory of quantum gravity exists. Furthermore, in C there are at least 57 ways of doing any given task, with exactly 17 of them being right, and at most 3 of them being decipherable 10 minutes after coding. C is also used extensively in the creation of music, depending on what key is being used (I keep all mine on a key group, which is much simpler to understand than a key ring). C was especially useful for this purpose after C# was invented by Microsoft in an attempt to take over the world. However, it is simple to derive from basic information theory that Google already controls the world, so really it was a lot of wasted effort on the part of Microsoft, but that seems to be something they are good at. It is also noteworthy that C was used to create the ocean, although this led to sea anomalies, which are brightly colored animals that like to sit on coral reefs and ignore people.

It should also be noted that C is a very common word in Spanish, which can signify agreement, affirmation, certainly, exactly, definitely, indubitably, and probably about a hundred other English words. As a matter of fact, C is one of only three Spanish words that I know. The other two are taco-bell (the name of a very authentic Mexican restaurant near where I live) and tueselestupidoguerroynopalletayelwheigholaestrayaenel-momentonobienoestupidoguerro, which I learned because of an interesting experience involving an angry mob of migrant workers of questionable legal status armed with buckets of strawberries. As far as I know this is a type of muffin. At one point I intended to learn some additional Spanish words and phrases, such as hola (which means "I hate you") and hospital (which is a big building full of sick people, but that's not important right now). I also wanted to learn to do math in Spanish, but soon discovered that the only thing different is what things are called, the actual math works the same, and seeing as I was already quite used to calling such things by their English names, I lost interest after learning to count to 59 and a half (true story...maybe).

Alright so now I can hear all of you asking (no, really, I can...) "Okay, so C is probably the greatest thing since amoebic dysentery, but how does one go about learning to use it in everyday life?" Well, I am so glad you asked that, because that is now what I am going to talk about. It is probably best to start at the beginning. The first important fact to remember is that C runs on computers and when someone first told me that I replied by saying that I run on shoes, and as a curious coincidence, it also happens that I run on sentences. However, this was only met by a blank stare, most likely because experienced C programmers appear to an outside observer to have no sense of humor. This however is not true, it's just that certain ways of doing things in C are so funny that an average joke seems mundane by comparison. I would explain these to you, except that it would inevitably result in your death by laughter. The only way for this not to happen is if you are simultaneously trying to track down an obnoxious memory leak, which is always the case for C programmers, so there is little for them to worry about.

So now I can hear you asking (not really this time, I'm just employing a literary technique to demonstrate empathy with the reader) "Alright, so C runs on computers, you run on shoes and sentences, and diesel locomotives run on electricity (haha, a joke), but what else do I need to know before I start programming in C?" A very good question... or at least a somewhat acceptable question. The first thing a C programmer needs to know is you should put semi-colons (;) everywhere. If a program doesn't work, the first thing to try is going through and adding lots of them. I recommend using one after every character (except white-space, since C doesn't see it) and at least 5 at the end of every line. I know that sounds really annoying, but you will get used to it in time. The next thing a C programmer needs to know is how pointers work. This is very mysterious to many people; however, it is really quite simple: pointers point to things. Even young children

are able to master this skill (except in Canada, where children are normally kept in small dark boxes for the first 11 years of their life). In C, to point to something, you use an asterisk (\*), which I know sounds ridiculous seeing as it points in all directions, but that's the way it works. For example, if you wanted to point to a tree, you would use \*tree, and if you wanted to point at the thing pointing to the tree, you would use \*\*tree, and so on. There is also another symbol (&) which is almost, but not entirely unlike \*. However, the actual details of what it does are only understood by three people, and all of them are dead on account of having eliminated all the memory leaks in their programs.

So now that you understand the basics of C, it's time to examine a simple C program, as well as some things from C++, to keep it from drilling holes (haha, another joke).

```
#include <studio.mpg>
#include "random_functions.h"
#include "vegetables.h"
#include "SmallWoodlandCreatures.idk"

TSmallWoodlandCreatures aSWC;
TAsparagus              aAsparagus;

int main() {

    aAsparagus.cook(); //look, this is a comment

    for(i=x=y=z=yourmom= 0, for(j=0; j !=100;
        printf("%5f, z), j++);;i++,++x, --y, y--) {

        foldinhalf(aSWC.getferret())

        if(aSWC.ferretisokay())
            break;
    }

    return 11.2 and a weasel;
}
```

Now, you should notice that care was taken to make everything easy to read, and that no further explanation beyond the program comments is necessary to understand the program operation. While it may be hard for a beginning programmer, one should always strive to make things easy to read, printing things out in size 72 font can help with this. You can see in the above program many [many!] places where things were defibrilately done in a way that is easy to read, even though there may be a more direct approach (such as hitting someone over the head with a large rubber mallet).

Okay, with that you should be ready to begin exploring the exciting world of programming in C! Just make sure you use parenthesis every time, to prevent stupidly tampered-with data(s), and to protect against memory leaks growing inside your program. If you want to learn more about advanced programming topics, please read the follow-up article titled "object-oriented potato hashing" or the slightly esoteric and theoretical "Cb!B" Or if you actually wanted to learn something [editors note: this seems highly unlikely], you could always read a book on C (and yes, I actually mean a book called "A Book on C", the authors apparently weren't very creative, or else they probably would have called it "An Apology from the Cane Toad and all its Friends").

```
//EOF encountered
```