Why C

Understanding the need to learn C

Wanted to learn C? Don't worry, you are at the right place! If you are wondering if this series on C is meant for you, then here's the answer. I wanted to write these tutorials in such a way that beginners will easily be able to grasp what's in here and at the same time I did not want to compromise on the content. And that is one helluva task to achieve! So I might dare say that beginners as well as those people who are somewhat familiar with C will find this series helpful. Just stick around, and find this out for yourself!

Why do we need to bother ourselves with this programming language called C? This is the first question that should crop up in your minds before we actually get into the subject. Let me answer that for you. C was actually designed in the seventies with an objective of re-building the **Unix operating system**. It has since become one of the most widely used programming languages of all time. As time went by, many high-level languages have since been developed. But C still stands tall amidst those newer programming languages. I only came to know about the power of C after I began to learn it.

Those who know a little bit of programming would ask why learn C when he/she is already acquainted with high-end languages like Python, Ruby, Java, etc (only to name a few). In this post, I would like to answer that question and list out a few reasons about why you'd try to learn C.

0. The very reason most of you are possibly reading this is because C is in your college curriculum, you need to learn it just to pass and you are dreaded of it. Fair enough! Well, though that may not be a proper reason (That's why it's numbered as the zeroth reason) but I promise you that if that's the case with you, in coming tutorials you'll see that C is really a fun language to learn.
1. What I think is important to learn as a programmer is how your program is being implemented "behind-the-scenes". With C, you can learn that easily. For me this was the reason for learning C (I usually code in Python). The language is so basic that you'd be easily able to understand what is happening behind the scenes (or so to say!). So if you want to migrate to some so-called advanced languages later, it is better to get some insight into the very basics of implementation of C and then proceed.
2. Even today, C is faster than almost all the newer languages. Nothing beats the speed of execution of a program designed in C and hence its speed is efficiently used in many places. This is because C is very "close to the machine". What I mean by this is C is very basic in its implementation, there is nothing very special about the language and that is the reason why it is so fast. The newer languages have many advanced features like

dynamic typing, garbage collection, etc for the programmer's convenience which costs a great deal of additional processing but C does not do any of that. Even the task of allocation and freeing of memory is left on the programmer (I guess you really have to be smart about that, eh? Nah! These are some real advanced things that we'll deal with later).

3. Finally, I can say that since almost all the languages borrow heavily from C, this will make things easier when you try to migrate to some other language from C. You can learn it better once you get comfortable with C.

The vast range of applications of C even today only goes on to prove how powerful this language is. To show how much extensively C is used even today, I'll just outline a few important applications of C (and trust me, there are many more!).

- Most of the modern operating systems like Windows, UNIX, Linux are extensively written in C because of the excellent performance which can be obtained from C. Moreover, device driver programs used to make the OS compatible with new hardware devices are also written in C.
- Most of the "smart" devices are so smart because of C. Even the "not-so-smart" devices need C to work. What I mean is most devices like mobiles, tablets, microwave ovens, televisions, washing machines, radios all work on a microprocessor, the program for which has to work very fast and in very less memory. No wonder C saves the day once again!
- This one is for you, gamers! Many popular gaming frameworks (including DirectX, idTech (Doom) and many others) are written in C and one of its many children, C++.
- Many compilers and interpreters are written in C as well. Most interestingly the compiler of C is written in C itself! Actually the first compiler for C was written in assembly language and B programming language (C's predecessor). But then it evolved and GCC, a modern compiler of C is almost completely written in C (Ahh! That's so confusing. Darn it!).

Enough talking about why, let's get into learning how next. Wait for the next tutorial from which we'll start learning how to code in C and see many interesting things about C as we go along. Happy Journey!!