Structured Programming Language
Lecture – 1

C Tutorial

C programming is a general-purpose, procedural, imperative computer programming language developed in 1972 by Dennis M. Ritchie at the Bell Telephone Laboratories to develop the UNIX operating system. C is the most widely used computer language. It keeps fluctuating at number one scale of popularity along with Java programming language, which is also equally popular and most widely used among modern software programmers.

Why to Learn C Programming?

C programming language is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Software Development Domain. I will list down some of the key advantages of learning C Programming:

- Easy to learn
- Structured language
- It produces efficient programs
- It can handle low-level activities
- It can be compiled on a variety of computer platforms

Facts about C

- C was invented to write an operating system called UNIX.
- C is a successor of B language which was introduced around the early 1970s.
- The language was formalized in 1988 by the American National Standard Institute (ANSI).
- The UNIX OS was totally written in C.
- Today C is the most widely used and popular System Programming Language.
- Most of the state-of-the-art software have been implemented using C.

Applications of C Programming

C was initially used for system development work, particularly the programs that make-up the operating system. C was adopted as a system development language because it produces code that runs nearly as fast as the code written in assembly language. Some examples of the use of C are -

- Operating Systems
- Language Compilers
- Assemblers
- Text Editors
- Print Spoolers
- Network Drivers
- Modern Programs
- Databases
- Language Interpreters
- Utilities

C Program Structure

A C program basically consists of the following parts −

- Preprocessor Commands
- Functions
- Variables
- Statements & Expressions
- Comments

Hello World using C Programming

Just to give you a little excitement about **C programming**, I'm going to give you a small conventional C Programming Hello World program, You can try it using Demo link.

```
#include <stdio.h>
int main()
 {
 /* my first program in C */
printf("Hello, World! \n");
 return 0;
 }
```

C Structure

Let us take a look at the various parts of the above program −

•The first line of the program *#include <stdio.h>* is a preprocessor command, which tells a C compiler to include stdio.h file before going to actual compilation.

•The next line *int main()* is the main function where the program execution begins.

•The next line /*...*/ will be ignored by the compiler and it has been put to add additional comments in the program. So such lines are called comments in the program.

•The next line *printf(...)* is another function available in C which causes the message "Hello, World!" to be displayed on the screen.

•The next line **return 0;** terminates the main() function and returns the value 0.

C Language Basic Syntax Rules

The C language syntax specifies the rules for writing the code in the C language. In simple words, these rules inform how to form statements in a C language program - How should the line of code start, how it should end, where to use double quotes, where to use curly brackets, where to use parenthesis, etc.

C Language Basic Syntax Rules

❑ What is Syntax?

Any language, be it English or Hindi, or Spanish, has a **grammar** that defines the **rules for using the language**, for example *how to form a sentence*, what different *words mean*, etc.

In a normal spoken language or a computer programming language, syntax means how to arrange words, characters, special characters, to make a meaningful statement, or expression, etc.

If someone says there is a **syntax error** in the program, means you have not written the program correctly, you might have missed some semicolon or some other general mistake in typing the code for the program.

Having a **syntax error** doesn't mean your code's logic is incorrect, it means you have written it incorrectly. Once the **syntax is correct**, then only the **code is compiled and then run**.

C Language Basic Syntax Rules

C Tokens

The smallest individual unit in the C program is known as C Token. Tokens are either keywords or identifiers, constants, variables, or any other symbol which has some meaning in C language. The C program can also be called a collection of various tokens.

Hence the syntax for C language defines how to use these tokens together while writing the C language code.

Let's take an example,

```
#include <stdio.h>
 int main()
 {
 printf("Hello,World");
 return 0;
}
```

C Language Basic Syntax Rules

There are **two ways** in which we can write comments.
1.Using //: This is used to write a **single-line comment**.
2.Using /* */: Anything enclosed within /* and */ , will treated as **multi-line comments**.

C Structure

In a C program, the semicolon is a statement terminator. That is, each individual statement must be ended with a semicolon. It indicates the end of one logical entity.

Given below are two different statements −

```
printf("Hello, World! \n");
return 0;
```