Introduction and History of C Language

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.

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.
- Today's most popular Linux OS and RDBMS MySQL have been written in C.

History

Language	**Year**	**Developed By**
Algol	1960	International Group
BCPL	1967	Martin Richard
B	1970	Ken Thompson
Traditional C	1972	Dennis Ritchie
K & R C	1978	Kernighan & Dennis Ritchie
ANSI C	1989	ANSI Committee
ANSI/ISO C	1990	ISO Committee
C99	1999	Standardization Committee

Applications of C Programming

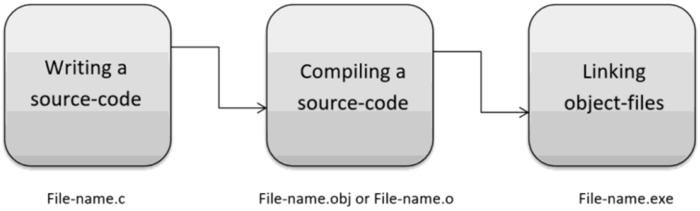
C was initially used for system development work, particularly the programs that make-up the operating system. C was adopted as a system development language because it produces code that runs nearly as fast as the code written in assembly language. Some examples of the use of C are -

- Operating Systems
- Language Compilers
- Assemblers
- Text Editors
- Print Spoolers
- Network Drivers

- Modern Programs
- Databases
- Language Interpreters
- Utilities

How 'C' Works?

C is a compiled language. A compiler is a special tool that compiles the program and converts it into the object file which is machine readable. After the compilation process, the linker will combine different object files and creates a single executable file to run the program. The following diagram shows the execution of a 'C' program



Nowadays, various compilers are available online, and you can use any of those compilers. The functionality will never differ and most of the compilers will provide the features required to execute both 'C' and 'C++' programs.

Following is the list of popular compilers available online:

- Clang compiler
- MinGW compiler (Minimalist GNU for Windows)
- Portable 'C' compiler
- Turbo C

Features of C Language

C is the widely used language. It provides many **features** that are given below.

1. **Simple:** C is a simple language in the sense that it provides a **structured approach** (to break the problem into parts), **the rich set of library functions**, **data types**, etc.
2. **Machine Independent or Portable:** Unlike assembly language, c programs **can be executed on different machines** with some machine specific changes. Therefore, C is a machine independent language.
3. **Mid-level programming language:** Although, C is **intended to do low-level programming**. It is used to develop system applications such as kernel, driver, etc. It **also supports the features of a high-level language**. That is why it is known as mid-level language.
4. **structured programming language:** C is a structured programming language in the sense that **we can break the program into parts using functions**. So, it is easy to understand and modify. Functions also provide code reusability.
5. **Rich Library:** C **provides a lot of inbuilt functions** that make the development fast.

6. **Memory Management:** It supports the feature of **dynamic memory allocation**. In C language, we can free the allocated memory at any time by calling the **free()** function.
7. **Fast Speed:** The compilation and execution time of C language is fast since there are lesser inbuilt functions and hence the lesser overhead.
8. **Pointers:** C provides the feature of pointers. We can directly interact with the memory by using the pointers. We **can use pointers for memory, structures, functions, array**, etc.
9. **Recursion:** In C, we **can call the function within the function**. It provides code reusability for every function. Recursion enables us to use the approach of backtracking.
10. **Extensible:** C language is extensible because it **can easily adopt new features**.

C Hello World! Example

Here, is a Hello World program in C

```
#include<stdio.h>       //Pre-processor directive
void main()             //main function declaration
{
printf("Hello World"); //to output the string on a display
getch ();               //terminating function
}
```

Pre-processor directive

#include is a pre-processor directive in 'C.'

#include <stdio.h>, stdio is the library where the function **printf** is defined**.** printf is used for generating output. Before using this function, we have to first include the required file, also known as a header file (.h).

You can also create your own functions, group them in header files and declare them at the top of the program to use them. To include a file in a program, use pre-processor directive

```
#include <file-name>.h
```

File-name is the name of a file in which the functions are stored. Pre-processor directives are always placed at the beginning of the program.

The main function

The main function is a part of every 'C' program. We can represent the main function in various forms, such as:

- main()
- int main()
- void main()
- main(void)
- void main(void)
- int main(void)

The empty parentheses indicate that this function does not take any argument, value or a parameter. You can also represent this explicitly by placing the keyword void inside the parentheses. The keyword void means the function does not return any value, in this case, the last statement is always getch ().

```
#include<stdio.h>      //Pre-processor directive
int main()             //main function declaration
{
printf("Hello World"); //to output the string on a display
return 0;              //terminating function
}
```

In the above example, the keyword int means the function will return an integer value. In this case, the last statement should always return 0.

The source code

After the main function has been declared, we have to specify the opening and closing parentheses. **Curly brackets { },** indicate the starting and end of a program. These brackets must be always put after the main function. All the program code is written inside these brackets, such as declarative and executable part.

The **printf** function generates the output by passing the text "Hello World!"

The **semicolon ;** determines the end of the statement. In C, each statement must end with a semicolon.

Alphabets of C

Like every other language 'C' also has its own character set. A program is a set of instructions that when executed, generate an output. The data that is processed by a program consists of various characters and symbols. The output generated is also a combination of characters and symbols.

A character set in 'C' is divided into,

- Letters: Uppercase characters (A-Z), Lowercase characters (a-z)
- Numbers: All the digits from 0 to 9
- White spaces: Blank space, New line, Carriage return, Horizontal tab
- Special characters:

, (comma)

. (period)

; (semi-colon)

: (colon)

? (question mark)

' (apostrophe)

" (double quotation mark)

{ (opening curly bracket)

} (closing curly bracket)

[(left bracket)

] (right bracket)

((opening left parenthesis)

) (closing right parenthesis)

& (ampersand)

! (exclamation mark)

|(vertical bar)

/ (forward slash)

\ (backward slash)

~ (tilde)

_ (underscore)

$ (dollar sign)

% (percentage sign)

^ (caret)

+ (addition)

- (subtraction)

* (multiplication)

/ (division)

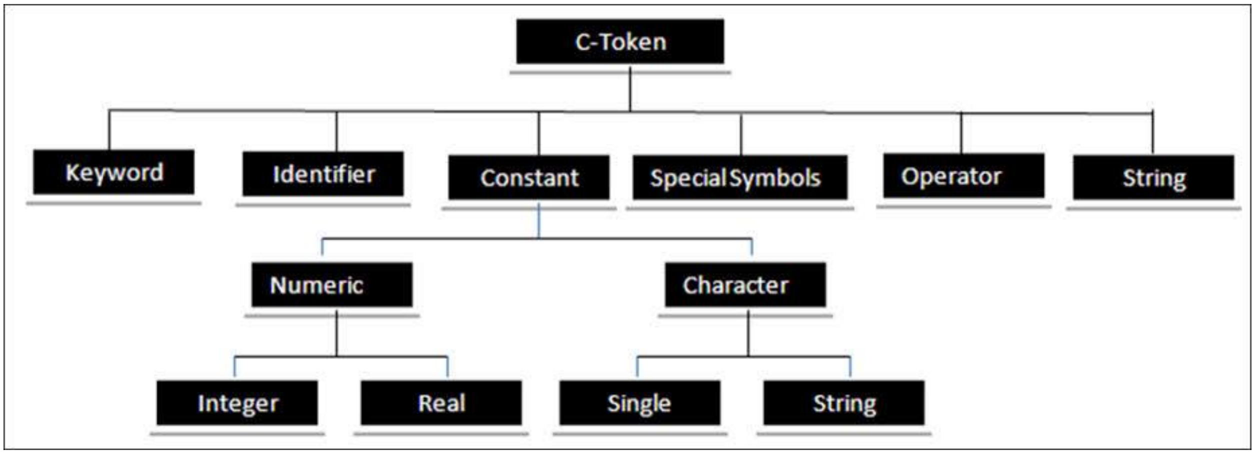
> (greater than or closing angle bracket)

< (less than or opening angle bracket)

(hash sign)

C Tokens

A token is the smallest unit in a 'C' program. A token is divided into six different types as follows,


C-Token
Keyword
Identifier
Constant
Special Symbols
Operator
String
Numeric
Character
Integer
Real
Single
String