

# C Programming: Structured Programming Language

**Mrs. Tapasya M. Mate**

Assistant Professor, Computer Science Department, Kaveri college pune.

## Abstract

The c programming language was developed by Dennis Ritchie at AT&T bell laboratories in early 1970s. C is powerful language that provides fast program execution. its power and fast program execution come from its ability to access low level commands, similar to assembly language, but with high level syntax. The purpose of this is to present an overview of c programming language its origins, its uses and program structures. C is strongly associated with UNIX, as it was developed to write the UNIX operating system.

**Keywords:** Programming, language, execution

## 1. Introduction

C language is one of the oldest and most powerful programming languages widely used across the world often called a middle-level computer language. Due to its portability, clean style, and closeness to the hardware, it is ideal for the low-level development of operating system kernels and embedded software. C language is considered as the base for many other programming languages and is known as the mother language. C was developed to overcome the challenges experienced with BASIC, B, and BCPL programming languages. Many other languages, such as Java, PHP, and JavaScript have borrowed syntax and features directly or indirectly from the C programming language. C Programming is still widely used in various applications. It is a popular language for developing software for microcontrollers, which are small embedded systems that control various devices. It is also commonly used for game development, scientific computing, and other applications that require high performance and efficient memory usage.

## History of C Language

1960	ALGOL	International Group
1967	BCPL	Martin Richards
1970	B	Ken Thompson
1972	Traditional c	Dennis Ritchie
1978	K & RC	Kernighan & Ritchie
1989	ANCI C	ANCI committee
1990	ANSI/ ISO C	ISO Committee
1999	C99	Standardization committee

### Advantages of C Language

**Simple:** C is a simple language as it offers a structured approach to solve problems. It also has a rich set of library functions and data types. C is usually taught as an introductory programming language as it is a well-established fact that it becomes easier to learn any other programming language if a person already knows the C language.

**Portable:** C programs can be written on one platform and can be executed in the same way on another operating system. C is a machine-independent language.

**Structured programming language:** C provides different functions that enable us to break the code into small parts, that is why C programs are easy to understand and modify. Functions also offer code reusability.

**Fast and Efficient:** Other programming languages, such as Python and Java offer more features than C language but their performance rate gets down due to additional processing. C programming language provides programmers access to direct manipulation with the computer hardware. The compilation and execution time of the C language is also fast.

**Extensible:** C programming can easily adopt new features.

Helps understand the fundamentals of Computer Theories: Many computer theories like operating systems, computer networks, compiler designing, and computer architecture are based on the C programming language and need a good knowledge of C programming to work on them.

### How does a C Program Work?

**1. Writing the code:** In this step, we write the source code for our program using a text editor or an IDE. For example, let's say we want to write a simple program that prints the message "Hello, world!" to the console. Here is what the code might look like:

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

**2. Compiling the code:** Once it has been written, we need to compile it using a C compiler. For example, we might use the GCC compiler on a Linux system. We would run the following command from the terminal: `gcc -o hello_world hello_world.c`

**3. Linking the code:** In this step, the C compiler may link our code with any necessary libraries or other code modules. Our simple "Hello, world!" program requires no additional libraries, so this step is relatively straightforward.

**4. Running the program:** With our program compiled and linked, we can now run it. On a Linux system, we would run the following command from the terminal: `./hello_world`

This would execute the `hello_world` binary and print the message "Hello, world!" to the console.

**5. Input and output:** In our "Hello, world!" program, there is no input required, and the only output is the message printed to the console.

**6. Memory management:** Our simple program does not require dynamic memory allocation or deallocation, so memory management is not a concern.

**7. Debugging and testing:** Finally, we want to test and debug our program to ensure it is working correctly. For a simple program like this, we might manually run it and verify that the output is correct. For

more complex programs, we might use debugging tools like gdb or automated testing frameworks to help identify and fix issues.

## Applications of C

**C language is used in a variety of applications:**

1. Language compiler development
2. Network drivers
3. IoT
4. Embedded systems, systems application, etc.
5. Language interpreters
6. Modern programs
7. Financial application development, like trading apps

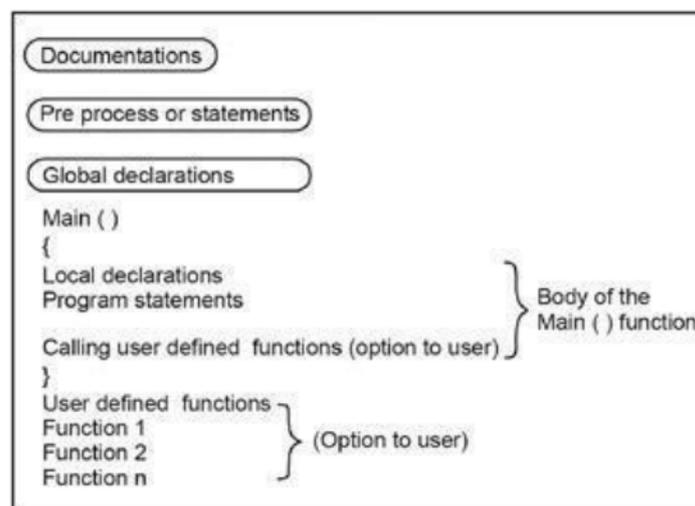
## Structure of C program

The basic structure of a C program is divided into 6 parts which makes it easy to read, modify, document, and understand in a particular format. C program must follow the below-mentioned outline in order to successfully compile and execute. Debugging is easier in a well-structured C program.

## Sections of the C Program

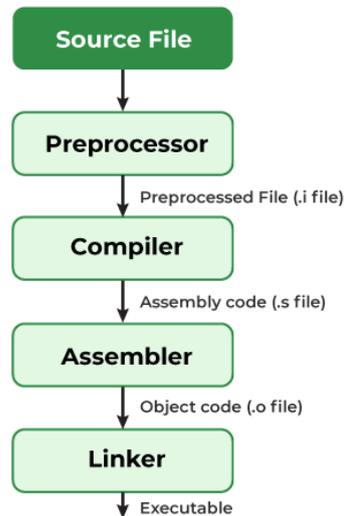
There are 6 basic sections responsible for the proper execution of a program. Sections are mentioned below:

1. Documentation
2. Preprocessor Section
3. Definition
4. Global Declaration
5. Main() Function
6. Sub Programs



## C Program Development Life Cycle

The compilation is a process of converting the source code into object code. It is done with the help of the compiler. The compiler checks the source code for the syntactical or structural errors, and if the source code is error-free, then it generates the object code.



Step 1. Writing Source code

Step 2. Compiling the source code( compilation)

Step 3. Linking the Object code

Step 4: Executing the Program (Execution)

## References

1. The C Programming Language - Dennis Ritchie, Brian Kernighan
2. C Programming: A Modern Approach: K. N King
3. [https://www.w3schools.com/c/c\\_intro.php](https://www.w3schools.com/c/c_intro.php)
4. <https://www.programiz.com/c-programming>
5. <https://www.geeksforgeeks.org/c-programming-language/>