Introduction to C

C is a procedural programming language initially developed by Dennis Ritchie in the year 1972 at Bell Laboratories of AT&T Labs. It was mainly developed as a system programming language to write the UNIX operating system.

The main features of the C language include:

- General Purpose and Portable
- Low-level Memory Access
- Fast Speed
- Clean Syntax

These features make the C language suitable for system programming like an operating system or compiler development.

Why Should We Learn C?

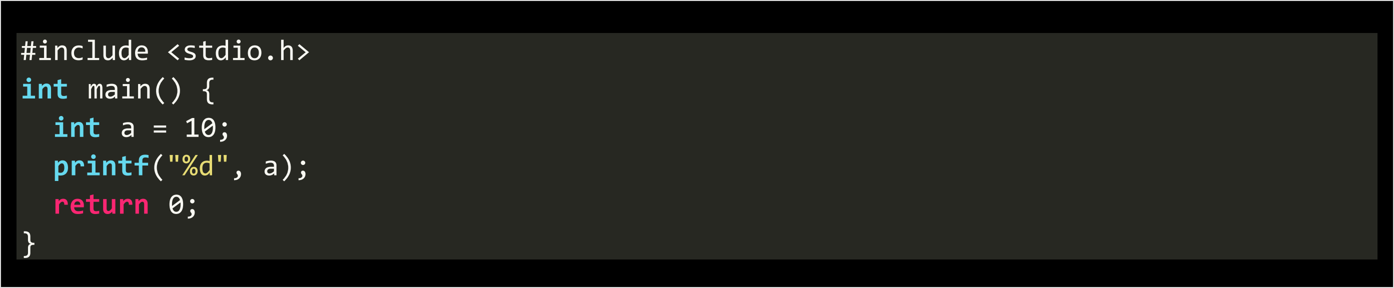
Many later languages have borrowed syntax/features directly or indirectly from the C language. Like syntax of Java, PHP, JavaScript, and many other languages are mainly based on the C language. C++ is nearly a superset of C language (Only a few programs may compile in C, but not in C++).

So, if a person learns C programming first, it will help him to learn any modern programming language as well. As learning C help to understand a lot of the underlying architecture of the operating system. Like pointers, working with memory locations, etc.

❖ Beginning with C programming:

Writing the First Program in C

The following code is one of the simplest C programs that will help us the basic syntax structure of a C program.



```c
#include <stdio.h>
int main() {
  int a = 10;
  printf("%d", a);
  return 0;
}
```

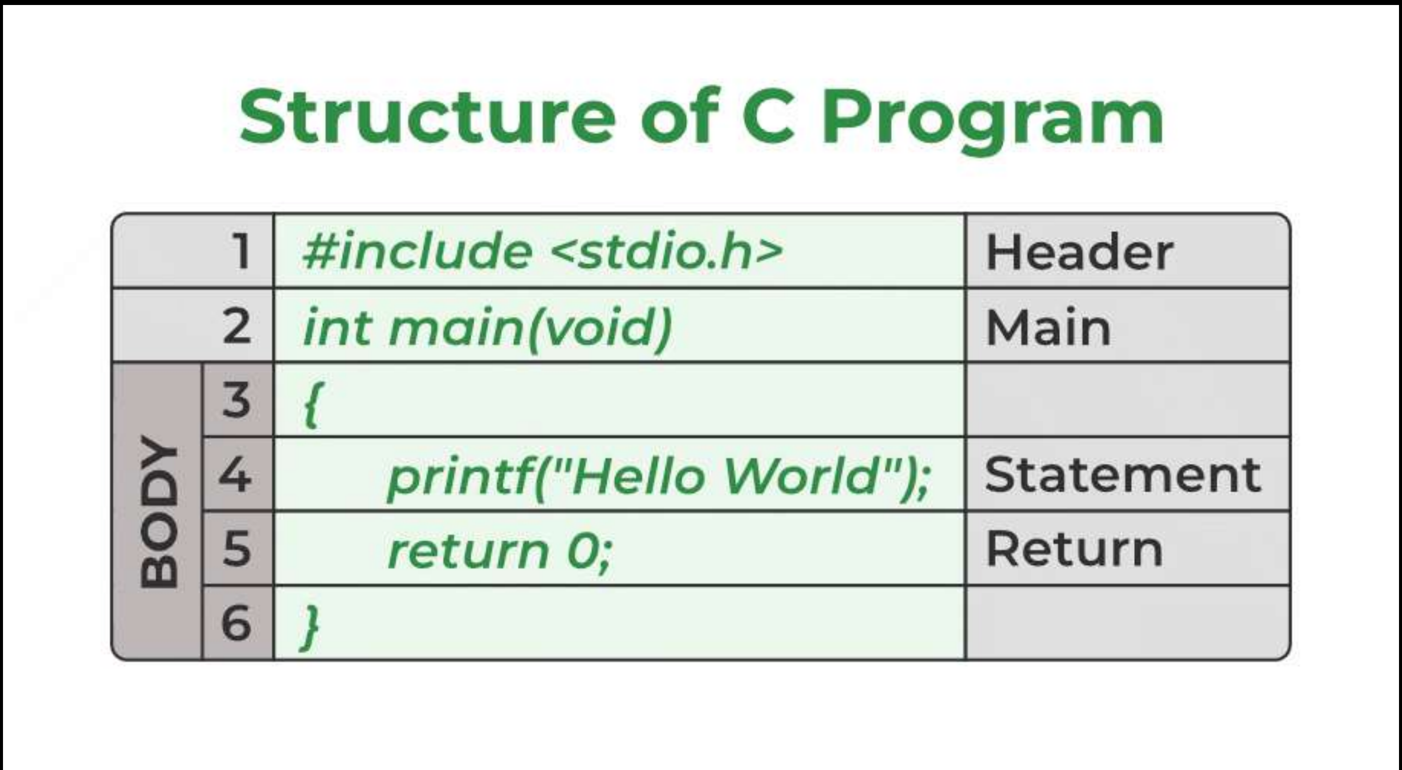
Output

```
10
```

Let us analyze the structure of our program line by line.

❖ Structure of the C program

After the above discussion, we can formally assess the structure of a C program. By structure, it is meant that any program can be written in this structure only. Writing a C program in any other structure will hence lead to a Compilation Error. The structure of a C program is as follows:



Structure of C Program

	1	#include <stdio.h>	Header
	2	int main(void)	Main
BODY	3	{	
	4	printf("Hello World");	Statement
	5	return 0;	Return
	6	}	

❖ Components of a C Program:

1. Header Files Inclusion – Line 1 [#include <stdio.h>]

The first and foremost component is the inclusion of the Header files in a C program. A header file is a file with extension .h which contains C function declarations and macro definitions to be shared between several source files. All lines that start with # are processed by a preprocessor which is a program invoked by the compiler. In the above example, the preprocessor copies the preprocessed code of stdio.h to our file. The .h files are called header files in C.

Some of the C Header files:

- stddef.h – Defines several useful types and macros.
- stdint.h – Defines exact width integer types.
- stdio.h – Defines core input and output functions
- stdlib.h – Defines numeric conversion functions, pseudo-random network generator, and memory allocation
- string.h – Defines string handling functions
- math.h – Defines common mathematical functions.

2. Main Method Declaration – Line 2 [int main()]

The next part of a C program is to declare the main() function. It is the entry point of a C program and the execution typically begins with the first line of the main(). The empty brackets indicate that the main doesn't take any parameter (See this for more details). The int that was written before the main indicates the return type of main(). The value returned by the main indicates the status of program termination. See this post for more details on the return type.

3. Body of Main Method – Line 3 to Line 6 [enclosed in {}]

The body of a function in the C program refers to statements that are a part of that function. It can be anything like manipulations, searching, sorting, printing, etc. A pair of curly brackets define the body of a function. All functions must start and end with curly brackets.

4. Statement – Line 4 [printf("Hello World");]

Statements are the instructions given to the compiler. In C, a statement is always terminated by a semicolon (;). In this particular case, we use printf() function to instruct the compiler to display "Hello World" text on the screen.

5. Return Statement – Line 5 [return 0;]

The last part of any C function is the return statement. The return statement refers to the return values from a function. This return statement and return value depend upon the return type of the function. The return statement in our program returns the value from main(). The returned value may be used by an operating system to know the termination status of your program. The value 0 typically means successful termination.

Application of C

- Operating systems: C is widely used for developing operating systems such as Unix, Linux, and Windows.
- Embedded systems: C is a popular language for developing embedded systems such as microcontrollers, microprocessors, and other electronic devices.
- System software: C is used for developing system software such as device drivers, compilers, and assemblers.
- Networking: C is widely used for developing networking applications such as web servers, network protocols, and network drivers.
- Database systems: C is used for developing database systems such as Oracle, MySQL, and PostgreSQL.
- Gaming: C is often used for developing computer games due to its ability to handle low-level hardware interactions.
- Artificial Intelligence: C is used for developing artificial intelligence and machine learning applications such as neural networks and deep learning algorithms.
- Scientific applications: C is used for developing scientific applications such as simulation software and numerical analysis tools.
- Financial applications: C is used for developing financial applications such as stock market analysis and trading systems.