**UNIT 1 : INTRODUCTION TO C :**

**C Language Introduction :**

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.

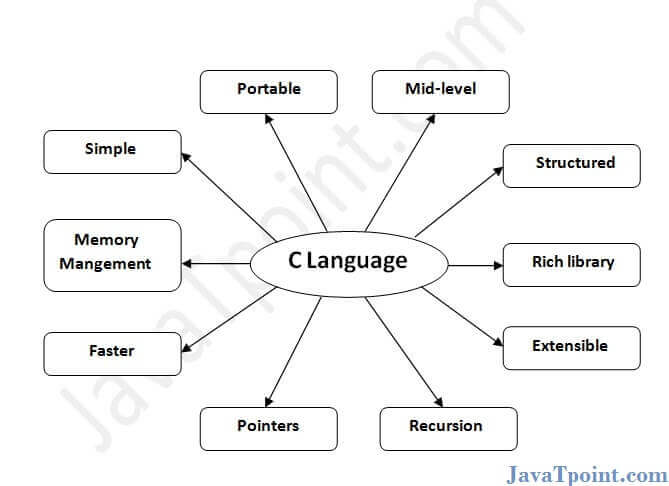
**The main features of the C language include:**

1. General Purpose and Portable
2. Low-level Memory Access
3. Fast Speed
4. Clean Syntax

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

**Features of c :**

* Simple
* Machine Independent or Portable
* Mid-level programming language
* structured programming language
* Rich Library
* Memory Management
* Fast Speed
* Pointers
* Recursion
* Extensible



**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) Speed**

The compilation and execution time of C language is fast since there are lesser inbuilt functions and hence the lesser overhead.

**8) Pointer**

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.

**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.

**Example:**

#include <stdio.h>

int main() {

int a = 10;

printf("%d", a);

return 0;

}

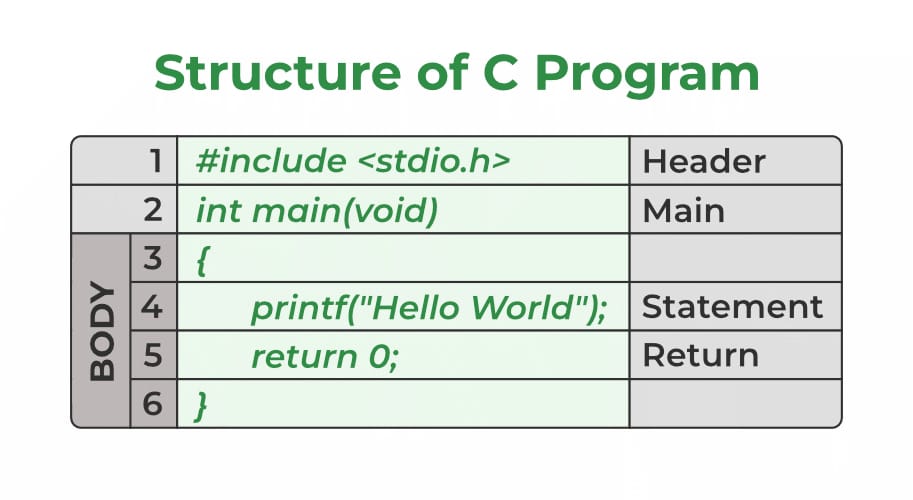
**Output**

10

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

**Structure of the C program :**

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:



**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.

**How to Execute the Above Program?**

In order to execute the above program, we need to first compile it using a compiler and then we can run the generated executable. There are online IDEs available for free like GeeksforGeeksIDE, that can be used to start development in C without installing a compiler.

**Windows:** There are many free IDEs available for developing programs in C like Code Blocks and **Dev-CPP**. IDEs provide us with an environment to develop code, compile it and finally execute it. We strongly recommend Code Blocks.

**Linux:** GCC compiler comes bundled with Linux which compiles C programs and generates executables for us to run. Code Blocks can also be used with Linux.

**macOS:** macOS already has a built-in text editor where you can just simply write the code and save it with a “.c” extension.