**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.

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;

}

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

 is a general-purpose, high-level language that was originally developed by Dennis M. Ritchie to develop the UNIX operating system at Bell Labs. C was originally first implemented on the DEC PDP-11 computer in 1972.

In 1978, Brian Kernighan and Dennis Ritchie produced the first publicly available description of C, now known as the K&R standard.

The UNIX operating system, the C compiler, and essentially all UNIX application programs have been written in C. C has now become a widely used professional language for various reasons −

* 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

## **More 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.

## **C Programs**

A C program can vary from 3 lines to millions of lines and it should be written into one or more text files with extension **".c"**; for example, *hello.c*. You can use **"vi"**, **"vim"** or any other text editor to write your C program into a file.

This tutorial assumes that you know how to edit a text file and how to write source code inside a program file.

f you want to set up your environment for C programming language, you need the following two software tools available on your computer, (a) Text Editor and (b) The C Compiler.

## **Text Editor**

This will be used to type your program. Examples of few a editors include Windows Notepad, OS Edit command, Brief, Epsilon, EMACS, and vim or vi.

The name and version of text editors can vary on different operating systems. For example, Notepad will be used on Windows, and vim or vi can be used on windows as well as on Linux or UNIX.

The files you create with your editor are called the source files and they contain the program source codes. The source files for C programs are typically named with the extension "**.c**".

Before starting your programming, make sure you have one text editor in place and you have enough experience to write a computer program, save it in a file, compile it and finally execute it.

## **The C Compiler**

The source code written in source file is the human readable source for your program. It needs to be "compiled", into machine language so that your CPU can actually execute the program as per the instructions given.

The compiler compiles the source codes into final executable programs. The most frequently used and free available compiler is the GNU C/C++ compiler, otherwise you can have compilers either from HP or Solaris if you have the respective operating systems.

The following section explains how to install GNU C/C++ compiler on various OS. We keep mentioning C/C++ together because GNU gcc compiler works for both C and C++ programming languages.

## **Installation on UNIX/Linux**

If you are using **Linux or UNIX**, then check whether GCC is installed on your system by entering the following command from the command line −

$ gcc -v

If you have GNU compiler installed on your machine, then it should print a message as follows −

Using built-in specs.

Target: i386-redhat-linux

Configured with: ../configure --prefix=/usr .......

Thread model: posix

gcc version 4.1.2 20080704 (Red Hat 4.1.2-46)

If GCC is not installed, then you will have to install it yourself using the detailed instructions available at <https://gcc.gnu.org/install/>

This tutorial has been written based on Linux and all the given examples have been compiled on the Cent OS flavor of the Linux system.

## **Installation on Mac OS**

If you use Mac OS X, the easiest way to obtain GCC is to download the Xcode development environment from Apple's web site and follow the simple installation instructions. Once you have Xcode setup, you will be able to use GNU compiler for C/C++.

Xcode is currently available at [developer.apple.com/technologies/tools/](https://developer.apple.com/technologies/tools/).

## **Installation on Windows**

To install GCC on Windows, you need to install MinGW. To install MinGW, go to the MinGW homepage, [www.mingw.org](https://www.mingw-w64.org/downloads/), and follow the link to the MinGW download page. Download the latest version of the MinGW installation program, which should be named MinGW-<version>.exe.

While installing Min GW, at a minimum, you must install gcc-core, gcc-g++, binutils, and the MinGW runtime, but you may wish to install more.

Add the bin subdirectory of your MinGW installation to your **PATH** environment variable, so that you can specify these tools on the command line by their simple names.

After the installation is complete, you will be able to run gcc, g++, ar, ranlib, dlltool, and several other GNU tools from the Windows command line.