



Andhra Pradesh State Skill Development Corporation



Programming in C

Introduction to C Language



Introduction of C language

Programming language:

A programming language is a set of commands, instructions, and other syntax used to create a software program. Languages that programmers use to write code are called "high-level languages." This code can be compiled into a "low-level language," which is recognized directly by the computer hardware.

High-level languages are designed to be easy to read and understand. This allows programmers to write source code in a natural fashion, using logical words and symbols. For example, reserved words like function, while, if, and else are used in most major programming languages. Symbols like <, >, ==, and != are common operators. Many high-level languages are similar enough that programmers can easily understand source code written in multiple languages.

Introduction of C :

C is a general-purpose programming language, and is used for writing programs in many different domains, such as operating systems, numerical computing, graphical applications, etc. It is a small language, with just 32 keywords. It provides "high-level" structured programming constructs such as statement grouping, decision making, and looping, as well as "low level" capabilities such as the ability to manipulate bytes and addresses.

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:

1. Easy to learn
2. Structured language
3. It produces efficient programs
4. It can handle low-level activities
5. It can be compiled on a variety of computer platforms

Applications of C language:

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 might be:

- Operating Systems
- Language Compilers
- Assemblers
- Text Editors
- Print Spoolers
- Network Drivers
- Modern Programs
- Databases
- Language Interpreters
- Utilities



History of C language:

- C programming language was developed in 1972 by Dennis Ritchie at bell laboratories of AT&T (American Telephone & Telegraph), located in the U.S.A.
- Dennis Ritchie is known as the founder of the C language.
- It was developed to overcome the problems of previous languages such as B, BCPL, etc.
- Initially, C language was developed to be used in the UNIX operating system. It inherits many features of previous languages such as B and BCPL.

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

Structure of C Program:

A C program, whatever its size, consists of functions and variables. A function contains statements that specify the computing operations to be done, and variables store values used during the computation.

```

1      /* First C program: Hello World */
2      #include <stdio.h>
3      int main()
4      {
5          printf("Hello World!\n");
6          return 0;
7      }
```

1. Comments in C start with /* and are terminated with */. They can span multiple lines and are not nestable. For example,
 - a. /* this attempt to nest two comments /* results in just one comment, ending here: */ and the remaining text is a syntax error. */



2. Inclusion of a standard library header-file. Most of C's functionality comes from libraries. Header Files contain the information necessary to use these libraries, such as function declarations and macros.

3. All C programs have main() as the entry-point function. This function comes in two forms:

- a. `int main()`
- b. `int main(int argc, char *argv[])`

The first takes no arguments, and the second receives command-line arguments from the environment in which the program was executed—typically a command-shell. The function returns a value of type `int`

4. & 7. The braces { and } delineate the extent of the function block. When a function completes, the program returns to the calling function. In the case of main(), the program terminates and control returns to the environment in which the program was executed. The integer return value of main() indicates the program's exit status to the environment, with 0 meaning normal termination.

5. This program contains just one statement: a function call to the standard library function printf(), which prints a character string to standard output (usually the screen). Note, printf() is not a part of the C language, but a function provided by the standard library (declared in header stdio.h). The standard library is a set of functions mandated to exist on all systems conforming to the ISO C standard. In this case, the printf() function takes one argument (or input parameter): the string constant "Hello World!\n". The \n at the end of the string is an escape character to start a new line. Escape characters provide a mechanism for representing hard-to-type or invisible characters (e.g., \t for tab, \b for backspace, \" for double quotes). Finally, the statement is terminated with a semicolon (;). C is a free-form language, with program meaning unaffected by whitespace in most circumstances. Thus, statements are terminated by ; not by a new line.

6. `return 0` in C Programming Language: return value of the main function is considered the "Exit Status" of the application. On most operating systems returning 0 is a success status like saying "The program worked fine"

The C Compiler:

The source code written in the 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 the 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 of Geany editor for windows:

Geany is an open source cross platform text editor that is designed specifically for programmers thanks to its built-in support for over 50 programming languages.



1. First click on this link:

<https://www.geany.org/download/releases/>

2. After we will get window like the below:

Distribution	File	GPG Signature	GPG Key
Source (tar.gz)	geany-1.36.tar.gz	geany-1.36.tar.gz.sig (Instructions)	colombanw-pubkey.bt
Source (tar.bz2)	geany-1.36.tar.bz2	geany-1.36.tar.bz2.sig (Instructions)	colombanw-pubkey.bt
Windows	geany-1.36_setup.exe	geany-1.36_setup.exe.sig (Instructions)	eht16-pubkey.bt
Mac OSX	geany-1.36_osx-2.dmg	-	-

Release notes for Geany 1.36

Plugins

Geany has a few plugins included (Classbuilder, Export, Filebrowser, HTML Characters, Save Actions and Split Window) but many more plugins are available in the [Geany-Plugins](#) project.

For more details, see the [plugin page](#).

Older versions

For older versions, please see <https://download.geany.org/>.

geany-1.36_setup.exe
5.3/15.4 MB, 28 secs left

3. Choose appropriate software which is related to your OS (Operating System)
4. After completing the download just install and use that software.

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 <http://gcc.gnu.org/install/>.

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 the GNU compiler for C/C++. Xcode is currently available at 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, and follow the link to the MinGW download page. Download the latest version of the MinGW installation program, which should be named MinGW- .exe. While installing MinGW, 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.