

INTRODUCTION TO C LANGUAGE

Rimi Nesha

Dronacharya College of Engineering

Haryana

Abstract :- C language is basic programming language.c language is also known as system language and procedure oriented language.C language is developed by Dennis Ritchi in 1971. C language is also known as mother language because by using c language we can implement all other language.c language has rich library and it is simple language.c language is called mid level language because it is combination of low level language and high level language.

I. INTRODUCTION to c language:

Every full C program begins inside a function called "main". A function is simply a collection of commands that do "something". The main function is always called when the program first executes. From main, we can call other functions, whether they be written by us or by others or use built-in language features. To access the standard functions that comes with your compiler, you need to include a header with the #include directive. What this does is effectively take everything in the header and paste it into your program. Let's look at a working program.

Let's look at the elements of the program. The #include is a "preprocessor" directive that tells the compiler to put code from the header called stdio.h into our program before actually creating the executable. By including header files, you can gain access to many different functions--both the printf and getchar functions are included in stdio.h.

DESCRIPTION:

The very first thing you need to do, before starting out in C, is to make sure that you have a compiler. What is a compiler, you ask? A compiler turns the program that you write into an **executable** that your computer can actually understand and run. If you're taking a course, you probably have one provided through your school. If you're starting out on your own, your best bet is to use [Code::Blocks with MinGW](#). If you're on Linux, you can use [gcc](#), and if you're on Mac OS X, you can use [XCode](#). If you haven't yet done so, go ahead and get a compiler set up--you'll need it for the rest of the tutorial.The next

important line is int main(). This line tells the compiler that there is a function named main, and that the function returns an integer, hence int. The "curly braces" ({ and }) signal the beginning and end of functions and other code blocks. If you have programmed in Pascal, you will know them as BEGIN and END. Even if you haven't programmed in Pascal, this is a good way to think about their meaning.

C TOKENS: A token is a group of characters that basically or logically belong together.

Identifiers

Keywords

Variables

Constants

The next important line is int main(). This line tells the compiler that there is a function named main, and that the function returns an integer, hence int. The "curly braces" ({ and }) signal the beginning and end of functions and other code blocks. If you have programmed in Pascal, you will know them as BEGIN and END. Even if you haven't programmed in Pascal, this is a good way to think about their meaning.

The printf function is the standard C way of displaying output on the screen. The quotes tell the compiler that you want to output the literal string as-is (almost). The '\n' sequence is actually treated as a single character that stands for a newline (we'll talk about this later in more detail); for the time being, just remember that there are a few sequences that, when they appear in a string literal, are actually not displayed literally by printf and that '\n' is one of them. The actual effect of '\n' is to move the cursor on your screen to the next line. Notice the semicolon: it tells the compiler that you're at the end of a command, such as a function call. You will see The next command is getchar(). This is another function call: it reads in a single character and waits for the user to hit enter before reading the character. This line is included because many compiler environments will open a new console window, run the program, and then close the window before you can see the output. This command keeps that

window from closing because the program is not done yet because it waits for you to hit enter. Including that line gives you time to see the program run that the semicolon is used to end many lines in C. Finally, at the end of the program, we return a value from main to the operating system by using the return statement. This return value is important as it can be used to tell the operating system whether our program succeeded or not. A return value of 0 means success.

The next command is `getchar()`. This is another function call: it reads in a single character and waits for the user to hit enter before reading the character. This line is included because many compiler environments will open a new console window, run the program, and then close the window before you can see the output. This command keeps that window from closing because the program is not done yet because it waits for you to hit enter. Including that line gives you time to see the program run.

Now we see This is where things start to get interesting: the `scanf` function works by taking a string and some variables modified with `&`. The string tells `scanf` what variables to look for: notice that we have a string containing only `"%d"` -- this tells the `scanf` function to read in an integer. The second argument of `scanf` is the variable, sort of. We'll learn more about what is going on later, but the gist of it is that `scanf` needs to know where the variable is stored in order to change its value. Using `&` in front of a variable allows you to get its location and give that to `scanf` instead of the value of the variable. Think of it like giving someone directions to the soda aisle and letting them go get a coca-cola instead of fetching the coke for that person. The `&` gives the `scanf` function directions to the variable. Of course, no matter what type you use, variables are uninteresting without the ability to modify them. Several operators used with variables include the following: `*`, `-`, `+`, `/`, `=`, `==`, `>`, `<`. The `*` multiplies, the `/` divides, the `-` subtracts, and the `+` adds. It is of course important to realize that to modify the value of a variable inside the program it is rather important to use the equal sign. In some languages, the equal sign compares the value of the left and right values, but in C `==` is used for that task. The equal sign is still extremely useful. It sets the value of the variable on the left side of the equals sign equal to the value on the right side of the equals sign. The operators that perform mathematical functions should be used

on the right side of an equal sign in order to assign the result to a variable on the left side.

. Advantage

C language is simple language.

Structured language

Procedure oriented language

Disadvantage

C language is not able to programme high technologies.

Conclusion

C language is simple language. C language is not able to programme high technologies.