

Basic Syntax of a C Program :-

We are going to understand the basic syntax of a C program. We will be using the same program we have been using for previous two programs, so you may develop a better understanding about the syntax before proceeding further. So, let's get started with our tutorial.

A C program is made up of different **tokens** combined together. These tokens include:

- Keywords
- Identifiers
- Constants
- String Literal
- Symbols

```
• int a;  
• printf("Enter number a\n");  
• scanf("%d", &a)  
• return 0;
```

I have written a four-line code above so I can explain the tokens in a better way by using the references from the code above.

Keywords:

Keywords are reserved words that cannot be used elsewhere in the program for naming a variable or function, instead they have a specific function or tasks and they are solely used for that. In the above given code, the return statement in the third line is a keyword.

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

Identifiers:

Identifiers are names given to variables or functions in order to differentiate them from one another. They are solely based on our choice but there are few rules that we have to follow while naming identifiers. According to the rules the name can not contain special symbols such as @, -, *, <, etc. In the above given code the “a” integer is an identifier.

Note: C is a case sensitive language so an identifier containing a capital letter and another one containing a small letter at the same place will be different. For example the three words: Code, code and cOde can be used as three different identifiers.

Constant:

Constant are very similar to variable and their values can be of any data type. The only difference between constant and variable is that a constant's value never changes. We will see constants in more detail in the upcoming tutorials. In the above given code the “0” in the last line is a constant.

String literal:

String literal or string constant is a line of characters enclosed by double quotes. In the above given code “Enter number a” is a string literal. printf is being used there to print string literal onto the screen.

Symbol:

Symbols are special characters reserved to perform certain actions. They are used to notify the compiler so they can perform specific tasks on the given data. In the above example code & is being used as a symbol.

Let's talk a little about **white space**. White space or blank space does not create any difference while using C. Unlike Python where we have to press enter to go to new line, in C we use semi-colon (;) to end a line of code. So until a semi colon arrives, the compiler will treat the code as a single liner so no matter how many lines we consume the code will run accurately if written correctly.

There are two code snippets given below. You can notice that they differ a lot regarding white space but their execution wills how the same output onto the screen i.e. “**Hello World**”.

Code1:

```
#include <stdio.h>

int main()
{
    printf("Hello World\n");
    return 0;
}
```

Code2:

```
#include <stdio.h>

int main()
{
    Printf
(
"Hello World\n"
)

;

    return 0;
}
```