

Tokens in C

Tokens are an elementary part of the program. Everything is part of a token from the beginning of the C program, from primary () to the closing parenthesis. Tokens in C programming language is also an important part of the [GATE CSE syllabus](#).

Tokens in C Definition

Tokens in C are defined as the smallest individual unit in a program. Every chunk of the source code is formed with the help of tokens.

Like C programming, every programming language has its own set of tokens. A compiler takes the source code from the programmer and processes this source code to generate tokens. This task is assigned to the first phase of the compiler, known as the lexical analysis phase of the compiler.

Types of Tokens in C

In [C programming](#), there are various types of tokens. These tokens in C increase the code readability while providing modularity of code. A compiler breaks down the source code into different token types and then processes these tokens into the next phase until the entire principle is processed. Tickets can be divided into five classes in C. They are:

- Keywords
- Identifiers
- Operators
- Special Symbols
- Constants

Let us now briefly discuss each token to understand the meaning and usage of tokens in C programming.

Keywords in C

The keywords are predefined words, also known as reserved words. There are 32 keywords present in C programming. A keyword is always in lowercase and does not contain any particular symbol. Some of the keywords in C programming important for the [GATE exam](#) are as follows: Central, volatile, int, getch, signed, void, char, struct, break, continue, etc.

Identifiers in C

An identifier is a name the user or programmer gives to any memory location. They have no predefined meanings attached to them. They can be in upper case or lower case. They allow only underscore and no other special symbol. Example: function(), int length; here, funct is an identifier used to name a part, and length is an identifier

used to call a variable. You can learn [the difference between Keywords and Identifiers](#) to know them better.

Operators in C

Operators in C allow the user to perform various arithmetic and logical operations. In C, operators are divided into three types that is Unary operators, binary operators, and Ternary operators. Unary operators consist of a single operator and a single operand, and binary operators consist of two operands with an operator in between. The ternary operator has a defined condition, resulting in accurate and false conclusions.

Special Symbols in C

In C, various special symbols are used to provide the functionality to the user. Some of the special symbols are _ (underscore), *(Asterisk symbol), {}, (), etc. These symbols have special meanings and are therefore called so.

Constants in C

Constants in C allow a value to remain the same throughout the execution, unlike variables. As per the [GATE notes](#), we can define constants in C using two methods:

- By using preprocessor directives
- By using the const keyword

Once a value is defined as [constant in C programming language](#), we cannot modify it throughout the process's lifetime. Some of the examples of constants are 59.88 (floating-point stable), 78(integer constant), byjusEP (String constant), etc.

Tokens in C Language Uses

Well, for instance, say you want to communicate with a person, now if you do not know the language they speak, how will you share with them? You need to know the language they speak similarly to interact with the compiler; a program needs to be bonded with tokens.

Tokens give meaning to the source code and provide the functionality to the user to conveniently interact with the compiler.