

# Variables in C

A variable in C acts like a container that stores information or data values. Once the data values are stored in the variable, we can use this variable as many times as we want. A variable in C has some characteristics, like its lifetime, scope, and range of values it can store.

A variable in C can take different types of values like integer, character, string, float, and double values, which are important parts of the [GATE CSE syllabus](#). Each variable is associated with a data type that allows for storing values of a particular type.

## Variable in C Meaning

The name of a memory address we use to store data is all that a variable means. In C or any other programming language, we can modify a variable's value and reuse it several times. Variables play an integral part in any source code. Once a variable is declared, we can perform multiple tasks using it within the same source code.

## Variable in C Example

As we already know, every variable declared in the program is associated with a data type. It may take a range of values depending on the data type. Along with the data type, it may also use storage classes that precisely define the scope and lifetime of a variable. And these data types are essential as per the [GATE exam](#).

Let us see an example of variables in C:

```
#include<stdio.h>

int main()
{
int day= 12;

char grade= 'A';

float marks= 98.7;

string name="Saif";

printf("%d\n%c\n%f\n%s\n", day, grade, marks, name);

return 0;

}
```

The above program declares and initializes four variables that store data values according to the associated data type. The variables here are day, grade, marks, and name.

## Use of Variable in C

Knowing the meaning and definition of a variable, one might wonder what is the use of defining the variables in C? Well, for the efficient working and execution of the program, we use variables. Every variable has a type and size.

The importance of variables can be seen in the [GATE question papers](#) as well. Variables store data that the programmer or user can later use to complete the execution of a particular task or code.

## Declaration of Variable in C

Variable declaration in C provides the compiler with the credibility that a particular storage area is assigned some name along with the size and type.

## Initialization of Variable in C

One can also initialize the value of a variable that is assigning a valid value

(belonging to the specified type and size of the variable) to the variable is termed initialization. The syntax for initializing a variable is: type variable name= value;

Here int and float represent the type of variable, respectively, and marks and weight represent variables' names with values assigned to them.

## Data Types of Variables in C

In C, variables are divided based on various categories based on size, scope, lifetime data types, etc. Based on data types, we have two categories:

- Primitive data types
- [User-defined data types](#)
- Derived data types

1. **Primitive data types:** It is mandatory to assign a data type to a variable in C, which defines the type of data stored under the variable name. Primitive data types in C are shown in the below table:

2. **User-defined data types:** These data types have been designed to ease the user's purpose. In [C programming language](#), we have the following three types of user-defined data types:

- Structure
- Union
- Enum

**3. Derived data types:** In C, we have derived data types that can contain multiple homogenous data at once. Following are the derived data types in C:

- Array
- Functions
- Pointers
- Storage classes of variables in C

Every variable in C has characteristics like the data type, scope lifetime, etc. The storage classes deal with the scope and lifetime of the variable. The storage classes are mainly classified into four categories:

- Auto storage class
- Register storage class
- Static storage class
- Extern storage class

### **Auto Storage Class**

All the local variables are auto variables by default. Auto variables are different from the [constants in a C program](#).

The default contains a garbage value if an auto variable is not initialized. For every function call, the auto variable is recreated and re-initialized. It has local scope.

### **Register Storage Class**

The keyword register is just a request to the compiler to allocate memory in the registers instead of RAM; if the accessible registers are available, then memory will be allocated in the registers. Otherwise, it will be allocated in the RAM only.

### **Static Storage Class**

For static variables, the memory will be allocated only once at compile time; if it is not initialized, the default contains zero. A static variable will persist in its value even after the destruction of several function calls.

### **Extern Storage Class**

The variable declared extern would not be allocated any memory. These are also sometimes used in questions formation of [GATE CSE exam](#). It will use the external memory (global variables).