

## Variable type in C / C ++

A variable provides named storage so that we can manipulate. Each variable in C / C ++ has a specific type, which determines: the size and memory layout of the variable; The range of values ??can be stored inside that memory; and the set of activities can be applied to that variable.

A variable provides named storage so that we can manipulate. Each variable in C / C ++ has a specific type, which determines: the size and memory layout of the variable; The range of values ??can be stored inside that memory; and the set of activities can be applied to that variable.

Variable names may include characters, digits, and underscores. It must start with either a character or an underscore. Uppercase and lowercase letters are different because C / C ++ is a case-sensitive language. Variables can be in various value types such as char, int, float, .

### Definition of variables in C / C ++

The variable definition in C / C ++ means telling the compiler where and how much memory to create to store that variable. A variable definition defines a data type, and contains a list of one or more variables of that type, as follows:

```
kieu_gia_tri danh_sach_bien ;
```

Here, **kieu\_gia\_tri** must be a valid data type in C / C ++, including char, w\_char, int, float, double, bool or any other user-defined object, . and **list\_sach\_bien** may contain one or multiple Identifier names separated by commas. Here are some valid declarations in C / C ++:

```
int j , q , k ; char v , viet ; float x , diemthi ; double luong ;
```

Lines **int j, q, k;** just declared and defined variables j, q, k, but instructed the compiler to create variables with the name j, q, k with int.

Variables can be initialized (initialized) in declarations. Initializer consists of an equal symbol followed by a Constant Expression (constant expression), as follows:

```
kieu_gia_tri ten_bien = giaTri ;
```

#### For example:

```
extern int d = 3 , f = 5 ; // khai bao bien d va f. int d = 3 , f = 5
```

With the definition without initialization: variables are initialized with NULL (all bytes are 0); The initial value of all other variables is not defined.

## Declare variables in C / C ++

Declaring the variable in C / C ++ to be certain to the compiler that there is an existing variable with a given type and name, so that the compiler continues the compiler without knowing the full details of that variable. A variable declaration only makes sense at compile time, the compiler needs to declare the actual variable at the time of connecting the program.

Variable declarations are useful when you are using multiple files, and you define your variables in one of those files that will be available at the time of program connection. You will use extern keyword to declare a variable anywhere. Although, you can declare a variable multiple times in a C / C ++ program, but it can only be defined once in a file, a function, or a block of code.

### For example

In the following example, a variable is declared at the beginning of the program, but it has been defined inside the main function.

```
#include using namespace std ; // Phan khai bao bien: extern int a , b ; ext
```

When the above code is compiled and executed, it gives the following result:

```
30 23.3333
```

Like the concept applied on the function declaration, you provide the function name at the time of the declaration and the actual definition of that function can be provided anywhere. For example:

```
// phan khai bao ham int func () ; int main () { // phan goi ham int i = fu
```

## Lvalue and Rvalue in C / C ++

There are two types of Expression in C / C ++:

**lvalue** : This is an expression related to a device location. Any lvalue may appear on the left or right of an assignment.

**rvalue** : This concept relates to a data value that is stored at some addresses in the device. An rvalue is an expression that cannot have a value assigned to it, ie: rvalue may appear on the right but cannot appear on the left of an assignment.

Variables are lvalue that can appear on the left of the assignment. The numeric literal rvalue cannot be assigned and cannot appear on the left side of the assignment. Here is a valid command in C / C ++:

```
int t = 10 ;
```

But the following command is invalid and will give a compile-time error (error at compile time):

```
16 = 30 ;
```

**According to Tutorialspoint**

Previous article: [Data type in C / C ++](#)

Next article: [Variable scope in C ++](#)

You finished reading the article "**Variable type in C / C ++**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.

---