

Variable in C programming

A variable in C is nothing but a name given to storage memory so that the program can be manipulated. Each variable in C has a defined type, to determine the size and layout for that variable memory. The range of values that can be stored in memory, setting of expressions can be applied to variables.

A variable in C is nothing but a name given to storage memory so that the program can be manipulated. Each variable in C has a defined type, to determine the size and layout for that variable memory. The range of values that can be stored in memory, setting of expressions can be applied to variables.

The name of the variable may include letters, numbers and underscores (_), but it must begin with an alphabetic character or an underscore. Uppercase and lowercase letters are usually two distinct objects because C is a case-sensitive language. Based on the basic types explained in the previous chapter, there are the following types of basic variables:

Type Description
 Char is an integer variable, 1 byte in size.
 int The type for natural numbers.
 float The floating point accuracy value is single.
 double Dual precision floating point value.
 void Represents a type without type.
 The C programming language allows the definition of different types of variables, which can be viewed in the following chapters such as *enumerated variables*, *pointer variables*, *array variables*, *structure variables*, *Union variables*, .

Variable definition in C language

Defining variables means informing the compiler where and how to create the storage for that variable. A variable definition defines a data type and contains a list of one or more variables of that type as follows:

```
kieu_du_lieu danh_sach_bien ;
```

Here, kieu_du_lieu is of data type of C language such as char, w_char, int, float, double, bool or any other type of user defined object . **list_list_in** may include one or more separated identifiers each other by commas. Some valid declaration examples of variables are as follows:

```
int i , j ; char ho , ten , c , ch ; float f , luong , diemthi ; double
```

Line **int i, j;** just declared and defined for variables i, j, k and instructed the compiler to create variables under the name i, j, k with type int.

The variable can be initialized (assigned initial values) in its declaration. An initialization consists of a "=" followed by a numeric constant expression as follows:

```
kieu_du_lieu ten_bien = gia_tri ;
```

Some examples below:

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

With the definition of no initial value, static variables can store with NULL values, (all bytes have a value of 0), the initial value of all variables of all other types is valid Unknown.

Declare the variable in C language:

The variable declaration provides a guarantee for the compiler to recognize that there are no variables with the same type and name as it was previously declared, otherwise a compiler error will occur. A variable declaration is only meaningful at compile time, the compiler needs to declare a specific variable at the time of connection to the program.

A variable declaration is useful when you use multiple files simultaneously and you define your variable in one of those files. You can use extern keyword to declare variables anywhere. So you can declare a variable multiple times in C program, but only define in a file, a function or a block of code.

Usually extern variable is declared in file.h because when you want to use you only need to include the .h file to be able to use the variable.

For example

Try the example below, where the variable is declared at the top, but they are defined and initialized in main function:

```
#include // khai bao bien: extern int a , b ; extern int c ; extern float
```

Compile and run C program to see results:

Some concepts can be applied in function declarations where you provide the function name at the time it declares and defines it anywhere. For example:

```
// khai bao ham int tenham () ; int main () { // loi goi ham int i = t
```

Lvalue and Rvalue in C:

There are two types of Expression:

1. **lvalue** : Expression that refers to memory location as " **lvalue** ". A lvalue may appear either on the left or the right of an assignment.
2. **rvalue** : Concerning data values ??stored at some addresses in memory. An rvalue is an expression that cannot have a value assigned to it, meaning a rvalue may appear on the right but not the left of an assignment.

Variables are lvalues ??and often appear on the left side of the assignment. Constants are numbers that are rvalue and cannot be assigned and cannot appear on the left side of the assignment. Here is a valid declaration:

```
int g = 20 ;
```

But the following is an invalid declaration and will contain an error message:

```
10 = 20 ;
```

According to Tutorialspoint

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

Next article: [Constant in programming C](#)

You finished reading the article "**Variable in C programming**" 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.