

COMPUTER SCIENCE : C++

Variables in C++

A variable is a name which is associated with a value that can be changed. For example when I write `int num=20;` here variable name is `num` which is associated with value `20`, `int` is a data type that represents that this variable can hold integer values. We will cover the data types in the next tutorial. In this tutorial, we will discuss about variables.

Syntax of declaring a variable in C++

```
data_type variable1_name = value1, variable2_name = value2;
```

For example:

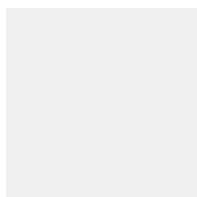
```
int num1=20, num2=100;
```

We can also write it like this:

```
int num1, num2;  
num1=20;  
num2=100;
```

Types of variables

Variables can be categorised based on their data type. For example, in the above example we have seen integer types variables. Following are the types of variables available in C++.



int: These type of of variables holds integer value.

char: holds character value like 'c', 'F', 'B', 'p', 'q' etc.

bool: holds boolean value true or false.

double: double-precision floating point value.

float: Single-precision floating point value.

Types of variables based on their scope

Before going further lets discuss what is scope first. When we discussed the [Hello World Program](#), we have seen the curly braces in the program like this:

```
int main {  
  
//Some code  
  
}
```

Any variable declared inside these curly braces have scope limited within these curly braces, if you declare a variable in main() function and try to use that variable outside main() function then you will get compilation error.

1. Global variable
2. Local variable

Global Variable

A variable declared outside of any function (including main as well) is called global variable. Global variables have their scope throughout the program, they can be accessed anywhere in the program, in the main, in the user defined function, anywhere.

Global variable example

Here we have a global variable myVar, that is declared outside of main. We have accessed the variable twice in the main() function without any issues.

```

#include <iostream>
using namespace std;
// This is a global variable
char myVar = 'A';
int main()
{
    cout <<"Value of myVar: "<< myVar<<endl;
    myVar='Z';
    cout <<"Value of myVar: "<< myVar;
    return 0;
}

```

Output:

```

Value of myVar: A
Value of myVar: Z

```

Local variable

Local variables are declared inside the braces of any user defined function, main function, loops or any control statements(if, if-else etc) and have their scope limited inside those braces.

Local variable example

```

#include <iostream>
using namespace std;

char myFuncn() {
// This is a local variable
char myVar = 'A';
}
int main()
{
    cout <<"Value of myVar: "<< myVar<<endl;
    myVar='Z';
    cout <<"Value of myVar: "<< myVar;
    return 0;
}

```

Output:

Compile time error, because we are trying to access the variable myVar outside of its scope. The scope of myVar is limited to the body of function myFuncn(), inside those braces.

Lets see an example having same name global and local variable.

```
#include <iostream>
using namespace std;
// This is a global variable
char myVar = 'A';
char myFuncn() {
    // This is a local variable
    char myVar = 'B';
    return myVar;
}
int main()
{
    cout <<"Funcn call: "<< myFuncn()<<endl;
    cout <<"Value of myVar: "<< myVar<<endl;
    myVar='Z';
    cout <<"Funcn call: "<< myFuncn()<<endl;
    cout <<"Value of myVar: "<< myVar<<endl;
    return 0;
}
```

Output:

```
Funcn call: B
Value of myVar: A
Funcn call: B
Value of myVar: Z
```

As you can see that when I changed the value of myVar in the main function, it only changed the value of global variable myVar because local variable myVar scope is limited to the function myFuncn().