**Methods**

Now, there are methods using which arguments are sent to the function. They are,

**Call by Value in C++**

Call by value is a method in C++ to pass the values to the function arguments. In the case of call by value the copies of actual parameters are sent to the formal parameter, which means that if we change the values inside the function that will not affect the actual values.

An example that demonstrates the call by value method is,

#include <iostream>

using namespace std;

void swap(int a, int b)

{

int temp = a;

a = b;

b = temp;

}

int main()

{

int x = 5, y = 6;

cout << "The value of x is " << x << " and the value of y is " << y << endl;

swap(x, y);

cout << "The value of x is " << x << " and the value of y is " << y << endl;

}

Output:

The value of x is 5 and the value of y is 6

The value of x is 5 and the value of y is 6

**Call by Pointer in C++**

A call by the pointer is a method in C++ to pass the values to the function arguments. In the case of call by pointer, the address of actual parameters is sent to the formal parameter, which means that if we change the values inside the function that will affect the actual values.

An example that demonstrates the call by pointer method is,

#include <iostream>

using namespace std;

void swap(int \*a, int \*b)

{

int temp = \*a;

\*a = \*b;

\*b = temp;

}

int main()

{

int x = 5, y = 6;

cout << "The value of x is " << x << " and the value of y is " << y << endl;

swap(&x, &y);

cout << "The value of x is " << x << " and the value of y is " << y << endl;

}

Copy

Output:

The value of x is 5 and the value of y is 6

The value of x is 6 and the value of y is 5

Copy

**Call by Reference in C++**

Call by reference is a method in C++ to pass the values to the function arguments. In the case of call by reference, the reference of actual parameters is sent to the formal parameter, which means that if we change the values inside the function that will affect the actual values.

An example that demonstrates the call by reference method is,

#include <iostream>

using namespace std;

void swap(int &a, int &b)

{

int temp = a;

a = b;

b = temp;

}

int main()

{

int x = 5, y = 6;

cout << "The value of x is " << x << " and the value of y is " << y << endl;

swap(x, y);

cout << "The value of x is " << x << " and the value of y is " << y << endl;

}

Output

The value of x is 5 and the value of y is 6

The value of x is 6 and the value of y is 5

Two special methods that are often used by programmers to have their program work efficiently are,

**Default Arguments in C++**

Default arguments are those values which are used by the function if we don’t input our value as parameter. It is recommended to write default arguments after the other arguments.

An example using default argument,

int sum(int a = 5, int b);

**Constant Arguments in C++**

Constant arguments are used when you don’t want your values to be changed or modified by the function. The const keyword is used to make the parameters non-modifiable.

An example using constant argument,

int sum(const int a, int b);