Passing Reference to a Pointer in C++

**Prerequisite**: [Pointers vs References in C++](https://www.geeksforgeeks.org/pointers-vs-references-cpp/).

For clear understanding, let’s compare the usage of a “pointer to pointer” VS “Reference to pointer” in some cases.

**Note:** It is allowed to use “pointer to pointer” in both C and C++, but we can use “Reference to pointer” only in C++.

* **Passing pointer to a function**

If a pointer is passed to a function as a parameter and tried to be modified then the changes made to the pointer does not reflects back outside that function. This is because only a copy of the pointer is passed to the function. It can be said that “pass by pointer” is [passing a pointer by value](https://www.geeksforgeeks.org/passing-by-pointer-vs-passing-by-reference-in-c/). In most cases, this does not present a problem. But the problem comes when you modify the pointer inside the function. Instead of modifying the variable, you are only modifying a copy of the pointer and the original pointer remains unmodified.

Below program illustrate this:

|  |
| --- |
| #include <iostream>    using namespace std;    int global\_Var = 42;    // function to change pointer value  void changePointerValue(int\* pp)  {      pp = &global\_Var;  }    int main()  {      int var = 23;      int\* ptr\_to\_var = &var;        cout << "Passing Pointer to function:" << endl;        cout << "Before :" << \*ptr\_to\_var << endl; // display 23        changePointerValue(ptr\_to\_var);        cout << "After :" << \*ptr\_to\_var << endl; // display 23        return 0;  } |

**Output:**

Passing Pointer to function:

Before :23

After :23

* **Passing “pointer to a pointer” as a parameter to function**

The above problem can be resolved by passing the address of the pointer to the function instead of a copy of the actual function. For this, the function parameter should accept a “pointer to pointer” as shown in the below program:

|  |
| --- |
| #include <iostream>    using namespace std;    int global\_var = 42;    // function to change pointer to pointer value  void changePointerValue(int\*\* ptr\_ptr)  {      \*ptr\_ptr = &global\_var;  }    int main()  {      int var = 23;      int\* pointer\_to\_var = &var;        cout << "Passing a pointer to a pointer to function " << endl;        cout << "Before :" << \*pointer\_to\_var << endl; // display 23        changePointerValue(&pointer\_to\_var);        cout << "After :" << \*pointer\_to\_var << endl; // display 42        return 0;  } |

**Output:**

Passing a pointer to a pointer to function

Before :23

After :42