[Skip to content](https://www.guru99.com/call-by-value-vs-call-by-reference.html" \l "primary)

* [Home](https://www.guru99.com/)
* [Testing](https://www.guru99.com/software-testing.html)
* [SAP](https://www.guru99.com/what-is-sap.html)
* [Web](https://www.guru99.com/java-tutorial.html)
* [Must Learn](https://www.guru99.com/design-analysis-algorithms-tutorial.html)
* [Big Data](https://www.guru99.com/bigdata-tutorials.html)
* [Live Project](https://www.guru99.com/live-testing-project.html)
* [AI](https://www.guru99.com/artificial-intelligence-tutorial.html)

**Difference between Call by Value and Call by Reference**

By[Matthew Martin](https://www.guru99.com/author/matthew)UpdatedSeptember 23, 2022

**Key Difference between Call by Value and Call by Reference**

* In the Call by value method original value is not modified, whereas in the Call by reference method, the original value is modified.
* In Call by value, a copy of the variable is passed, whereas in Call by reference, a variable itself is passed.
* In Call by value, actual and formal arguments will be created in different memory locations, whereas in Call by reference, actual and formal arguments will be created in the same memory location.
* Call by value is the default method in programming languages like C++, PHP, Visual Basic NET, and C#, whereas Call by reference is supported only in Java language.
* Call by Value variables is passed using a straightforward method, whereas Call by Reference pointers are required to store the address of variables.

**In this tutorial, you will learn:**

In programming languages, functions can be invoked in two ways: which is known as Call by Value and Call by Reference.

**What is Call by Value method?**

Call by value method copies the value of an argument into the formal parameter of that function. Therefore, changes made to the parameter of the main function do not affect the argument.

In this parameter passing method, values of actual parameters are copied to function’s formal parameters, and the parameters are stored in different memory locations. So any changes made inside functions are not reflected in actual parameters of the caller.

**What is Call by Reference method?**

Call by reference method copies the address of an argument into the formal parameter. In this method, the address is used to access the actual argument used in the function call. It means that changes made in the parameter alter the passing argument.

In this method, the memory allocation is the same as the actual parameters. All the operation in the function are performed on the value stored at the address of the actual parameter, and the modified value will be stored at the same address.

**C Code Example of a Call by Value method**

void main() {

int a = 10,

void increment(int);

Cout << "before function calling" << a;

increment(a);

Cout << "after function calling" << a;

getch();

void increment(int x) {

int x = x + 1;

Cout << "value is" << x;

}

**Output:**

before function calling 10

value is 11

after function calling 1-0

Because variable declared ‘a’in main() is different from variable ‘x’ in increment(). In this programme only variable names are similar, but their memory address are different and stored in different memory locations.

**Java Code Example of a Call by Reference method**

Public static void main(string args[]) {

int a = 10;

System.out.println("Before call Value of a = ", a);

Void increment();

System.out.println("After call Value of a = ", a);

}

Void increment(int x) {

int x = x + 1;

}

**Output:**

Before call Value of a =10

After call Value of a =11

Because variable declared ‘a’ in is referencing/ pointing to variable ‘a’ in main(). Here variable name is different, but both are pointing/referencing to same memory address locations.

**Call by Value vs. Call by Reference**

| **Parameters** | **Call by value** | **Call by reference** |
| --- | --- | --- |
| Definition | While calling a function, when you pass values by copying variables, it is known as “Call By Values.” | While calling a function, in programming language instead of copying the values of variables, the address of the variables is used it is known as “Call By References. |
| Arguments | In this method, a copy of the variable is passed. | In this method, a variable itself is passed. |
| Effect | Changes made in a copy of variable never modify the value of variable outside the function. | Change in the variable also affects the value of the variable outside the function. |
| Alteration of value | Does not allow you to make any changes in the actual variables. | Allows you to make changes in the values of variables by using function calls. |
| Passing of variable | Values of variables are passed using a straightforward method. | Pointer variables are required to store the address of variables. |
| Value modification | Original value not modified. | The original value is modified. |
| Memory Location | Actual and formal arguments will be created in different memory location | Actual and formal arguments will be created in the same memory location |
| Safety | Actual arguments remain safe as they cannot be modified accidentally. | Actual arguments are not Safe. They can be accidentally modified, so you need to handle arguments operations carefully. |
| Default | Default in many programming languages like C++.PHP. Visual Basic NET, and C#. | It is supported by most programming languages like JAVA, but not as default. |

**Advantages of using Call by value method in C**

Pros/benefits of a call by value in C:

* The method doesn’t change the original variable, so it is preserving data.
* Whenever a function is called it, never affect the actual contents of the actual arguments.
* Value of actual arguments passed to the formal arguments, so any changes made in the formal argument does not affect the real cases.

**Advantages of using Call by reference method**

Pros of using call by reference method:

* The function can change the value of the argument, which is quite useful.
* It does not create duplicate data for holding only one value which helps you to save memory space.
* In this method, there is no copy of the argument made. Therefore it is processed very fast.
* Helps you to avoid changes done by mistake
* A person reading the code never knows that the value can be modified in the function.

**Disadvantages of using Call by value method**

Here, are major cons/drawbacks of a call by value method:

* Changes to actual parameters can also modify corresponding argument variables
* In this method, arguments must be variables.
* You can’t directly change a variable in a function body.
* Sometime argument can be complex expressions
* There are two copies created for the same variable which is not memory efficient.

**Disadvantages of using Call by reference method**

Here, are major cons of using call by reference method:

* Strong non-null guarantee. A function taking in a reference need to make sure that the input is non-null. Therefore, null check need not be made.
* Passing by reference makes the function not pure theoretically.
* A lifetime guarantee is a big issue with references. This is specifically dangerous when working with lambdas and multi-threaded programs.

**You Might Like:**

* [**WebPagetest API Tutorial with Example**](https://www.guru99.com/how-to-use-webpagetest-api.html)
* [**Difference Between Waterfall vs Spiral and Incremental Model**](https://www.guru99.com/compare-waterfall-vs-incremental-vs-spiral-vs-rad.html)
* [**Capability Maturity Model (CMM) & it’s Levels in Software Engineering**](https://www.guru99.com/capability-maturity-model-cmm-cmm-levels-a-fool-s-guide.html)
* [**Incremental Model in SDLC: Use, Advantage & Disadvantage**](https://www.guru99.com/what-is-incremental-model-in-sdlc-advantages-disadvantages.html)
* [**What is RAD Model? Phases, Advantages and Disadvantages**](https://www.guru99.com/what-is-rad-rapid-software-development-model-advantages-disadvantages.html)

**Post navigation**

[Report a Bug](https://form.jotform.me/72391811797466)

[Prev](https://www.guru99.com/html-vs-html5.html)

[Next](https://www.guru99.com/difference-between-object-and-class.html)

Top Tutorials

**About**  
[About Us](https://www.guru99.com/about-us)  
[Advertise with Us](https://www.guru99.com/advertise-us)  
[Write For Us](https://www.guru99.com/become-an-instructor)  
[Contact Us](https://www.guru99.com/contact-us)

**[Python](https://www.guru99.com/python-tutorials.html)**

**[Testing](https://www.guru99.com/software-testing.html)**

**[Hacking](https://www.guru99.com/ethical-hacking-tutorials.html)**

**Career Suggestion**  
[SAP Career Suggestion Tool](https://www.guru99.com/best-sap-module.html)  
[Software Testing as a Career](https://www.guru99.com/software-testing-career-complete-guide.html)  
  
**Interesting**  
[eBook](https://www.guru99.com/ebook-pdf.html)  
[Blog](https://www.guru99.com/blog)  
[Quiz](https://www.guru99.com/tests.html)  
[SAP eBook](https://www.guru99.com/sap-ebook-pdf.html)

**[SAP](https://www.guru99.com/sap-training-hub.html)**

**[Java](https://www.guru99.com/java-tutorial.html)**

**[SQL](https://www.guru99.com/sql.html)**

**Execute online**  
[Execute Java Online](https://www.guru99.com/try-java-editor.html)  
[Execute Javascript](https://www.guru99.com/execute-javascript-online.html)  
[Execute HTML](https://www.guru99.com/execute-html-online.html)  
[Execute Python](https://www.guru99.com/execute-python-online.html)

**[Selenium](https://www.guru99.com/selenium-tutorial.html)**

**[Cryptocurrency](https://www.guru99.com/cryptocurrency-tutorial.html)**

**[VPNs](https://www.guru99.com/best-vpn.html)**

© Copyright - Guru99 2022         [Privacy Policy](https://www.guru99.com/privacy-policy)  |  [Affiliate Disclaimer](https://www.guru99.com/affiliate-earning-disclaimer)  |  [ToS](https://www.guru99.com/terms-of-service)