Pass by Reference in Functions

Learn a method to pass the reference of the actual parameters to the function.

We'll cover the following ^

Introduction#

Suppose you have sent an email to your friend with a link to a file present on **Google Drive**. Your friend made some changes to the document. Since you and your friend are sharing the same file, you will both see the changes made by either of you in the document.

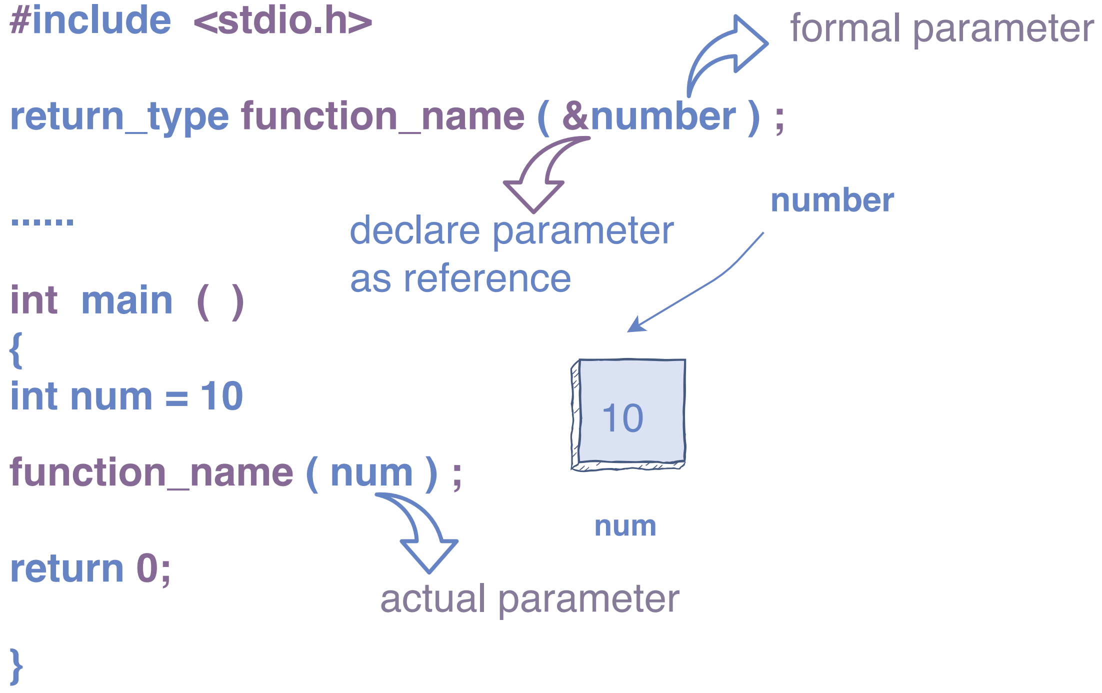
> *In **pass by reference**, when we call a function, we pass the address of the actual parameters to the formal parameters in the function.*

In pass by reference, the actual and formal parameters refer to the same memory location. Any changes made in the formal parameters inside the function affect the values of actual parameters in the main function.

function affect the values of actual parameters in the main function.

Basic syntax#

The general syntax for passing a reference to the function parameters is given below:

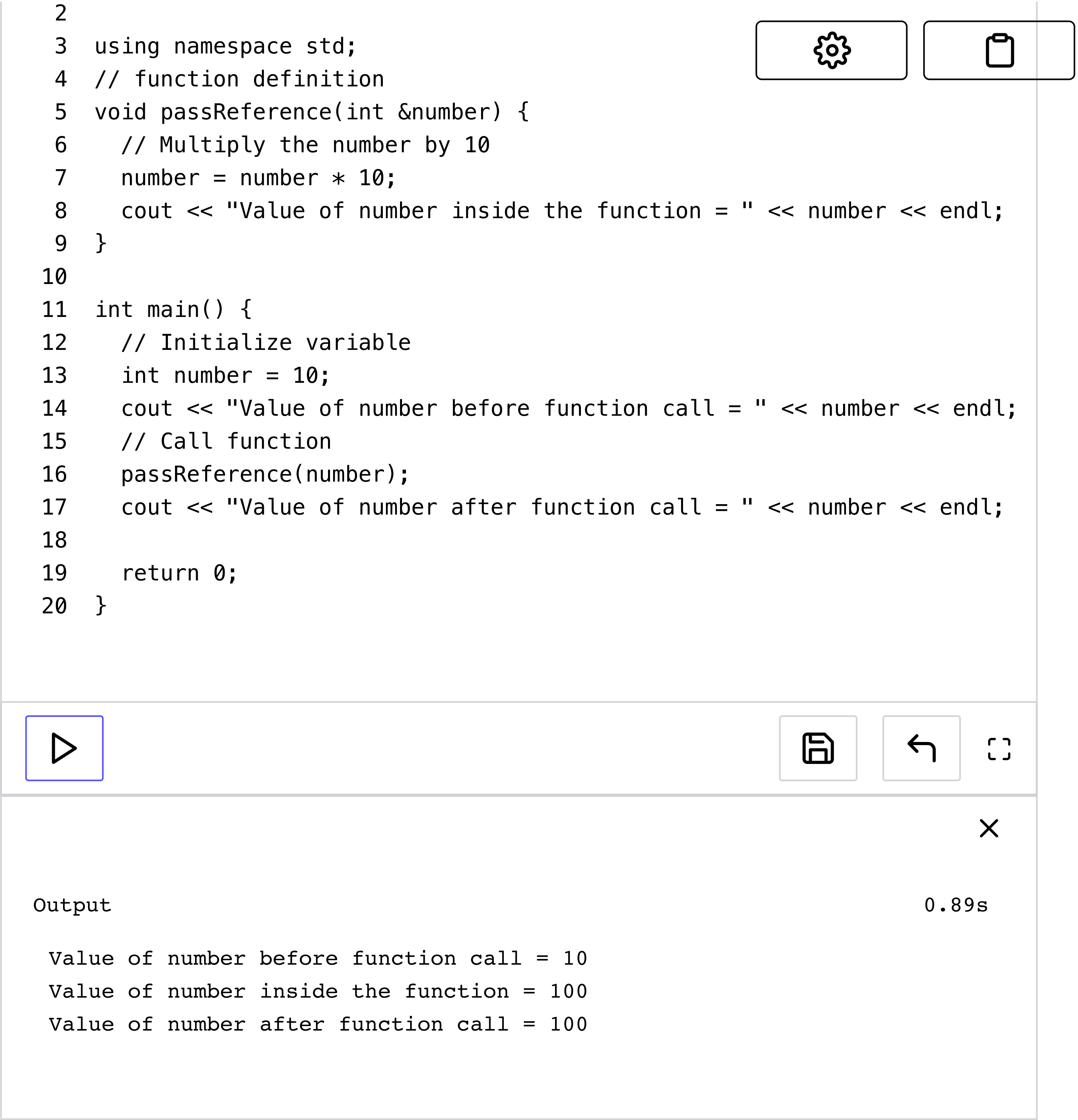


When we want to pass the value by reference, we declare function parameters as references rather than the normal parameters. To declare a function parameter as a reference, we have to ampersand `&` before the function parameter.

Example program#

Press the **RUN** button and see the output!

```
#include <iostream>
```



```

using namespace std;
// function definition
void passReference(int &number) {
  // Multiply the number by 10
  number = number * 10;
  cout << "Value of number inside the function = " << number << endl;
}

int main() {
  // Initialize variable
  int number = 10;
  cout << "Value of number before function call = " << number << endl;
  // Call function
  passReference(number);
  cout << "Value of number after function call = " << number << endl;

  return 0;
}
```

```
Value of number before function call = 10
Value of number inside the function = 100
Value of number after function call = 100
```

Explanation#

In the code above, we have two functions:

- `passReference` function
- `main` function

`passReference` function#

Line No. 5: The `passReference` function takes a value of type `int` by reference. It performs its task and then returns nothing in output.

Line No. 7: Multiplies the `number` by `10` and stores the result in the `number`.

Line No. 8: Prints the updated value of the `number`.

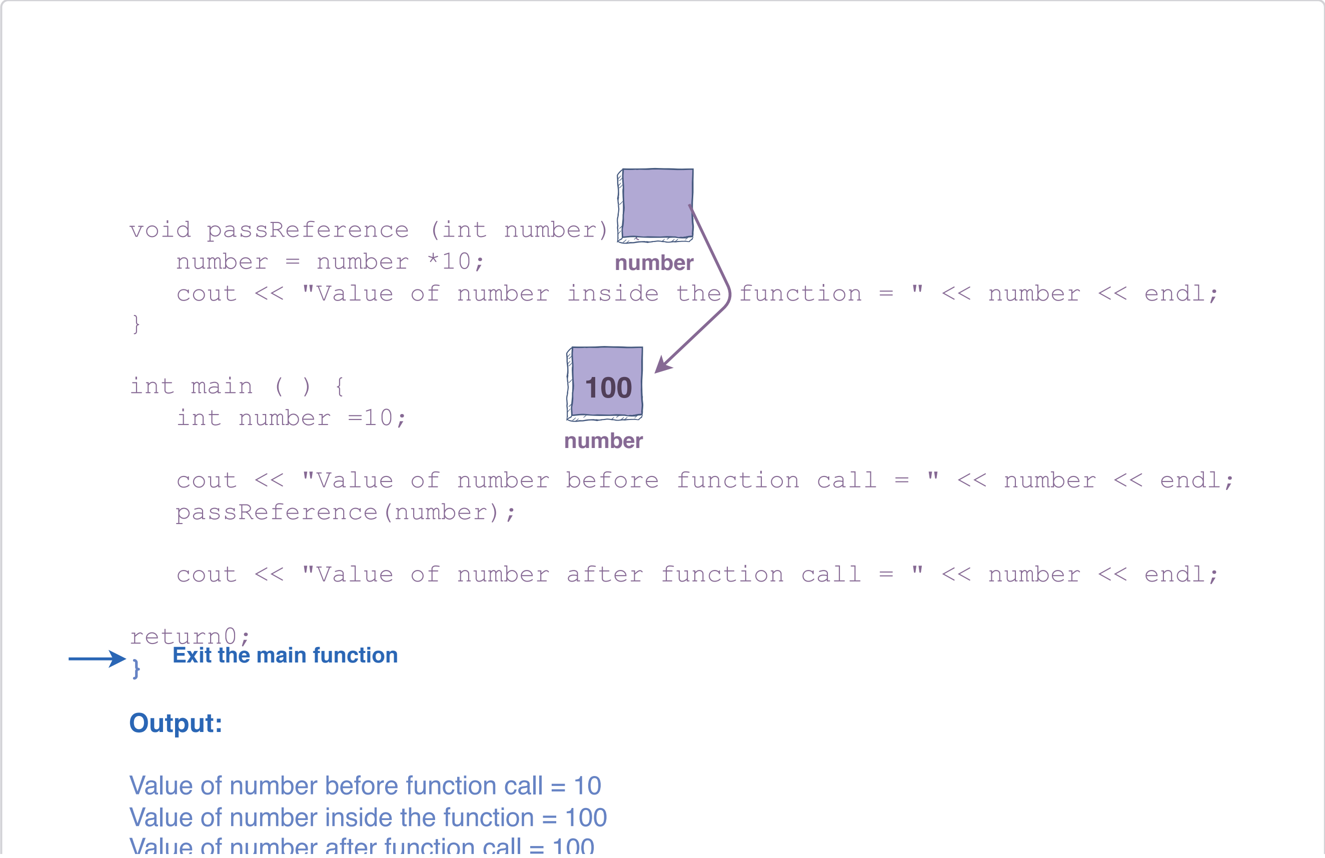
`main` function#

Line No. 13: Initializes a variable `number`.

Line No. 14: Prints the value of the `number` before the function call.

Line No. 16: Calls a function `passReference`. The execution control is transferred to **Line No. 5**.

Line No. 17: Prints the value of the `number` after the function call.



```
void passReference (int number)
   number = number *10;
   cout << "Value of number inside the function = " << number << endl;
}

int main ( ) {
   int number =10;

   cout << "Value of number before function call = " << number << endl;
   passReference(number);

   cout << "Value of number after function call = " << number << endl;

return0;
}
```

Output:

```
Value of number before function call = 10
Value of number inside the function = 100
Value of number after function call = 100
```

Value of number after function call = 100

11 of 11

Quiz

Q ✓ What is the output of the following code?

```
void cube(int &number) {
  number = number * number * number;
  cout << "number = " << number << endl;
}

int main() {
  int number = 5;
  cube(number);
  cout << "number = " << number << endl;

  return 0;
}
```

○ **A)** number = 5

number = 5

Your Answer

(/learn)

✓ **B)** number = 125

number = 125

Explanation

In pass by reference, the actual and formal paramters refer to the same memory location same memory location. Therefore, any changes made in the value of formal parameter will affect the actual parameter.

○ **C)** number = 125

number = 5

○ **D)** number = 5

number = 125

Submit Answer

Reset Quiz

This is all about passing values to functions. In the next lesson, we will classify variables according to their scope in the program.

← Back

Pass by Value in Functions

Next →

Scope of Variable

☑ Mark as Completed

Report an Issue