GeeksforGeeks
A computer science portal for geeks

Google Custom Search

Courses Login

Suggest an Article

How to return multiple values from a function in C or C++?

New programmers are usually in the search of ways to return multiple values from a function. Unfortunately, C and C++ do not allow this directly. But fortunately, with a little bit of clever programming, we can easily achieve this.

Below are the methods to return multiple values from a function in C:

1. By using pointers.
2. By using structures.
3. By using Arrays.

Example: Consider an example where the task is to find the greater and smaller of two distinct numbers. We could write multiple functions. The main problem is the trouble of calling more than one functions since we need to return multiple values and of course having more number of lines of code to be typed.

1. **Returning multiple values Using pointers:** Pass the argument with their address and make changes in their value using pointer. So that the values get changed into the original argument.

```
// Modified program using pointers
#include <stdio.h>

// add is the short name for address
void compare(int a, int b, int* add_great, int* add_small)
{
    if (a > b) {

        // a is stored in the address pointed
        // by the pointer variable *add_great
        *add_great = a;
        *add_small = b;
    }
    else {
        *add_great = b;
        *add_small = a;
    }
}

// Driver code
int main()
{
    int great, small, x, y;

    printf("Enter two numbers: \n");
    scanf("%d%d", &x, &y);

    // The last two arguments are passed
    // by giving addresses of memory locations
    compare(x, y, &great, &small);
    printf("\nThe greater number is %d and the"
           "smaller number is %d",
           great, small);

    return 0;
}
```

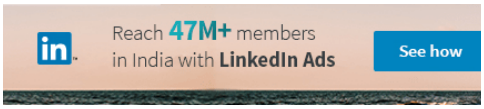
Output:

```
Enter two numbers:
5 8
The greater number is 8 and the smaller number is 5
```

2. **Returning multiple values using structures :** As the structure is a user-defined datatype. The idea is to define a structure with two integer variables and store the greater and smaller values into those variable, then use the values of that structure.

```
// Modified program using structures
#include <stdio.h>
struct greaterSmaller {
    int greater, smaller;
};

typedef struct greaterSmaller Struct;

Struct findGreaterSmaller(int a, int b)
{
    Struct s;
    if (a > b) {
        s.greater = a;
        s.smaller = b;
    }
    else {
        s.greater = b;
        s.smaller = a;
    }

    return s;
}

// Driver code
int main()
{
    int x, y;
    Struct result;

    printf("Enter two numbers: \n");
    scanf("%d%d", &x, &y);

    // The last two arguments are passed
    // by giving addresses of memory locations
    result = findGreaterSmaller(x, y);
    printf("\nThe greater number is %d and the"
           "smaller number is %d",
           result.greater, result.smaller);

    return 0;
}
```

Output:




```
Enter two numbers:
5 8
The greater number is 8 and the smaller number is 5
```

3. **Returning multiple values using an array (Works only when returned items are of same types):** When an array is passed as an argument then its base address is passed to the function so whatever changes made to the copy of the array, it is changed in the original array. Below is the program to return multiple values using array i.e. store greater value at arr[0] and smaller at arr[1].

```
// Modified program using array
#include <stdio.h>

// Store the greater element at 0th index
void findGreaterSmaller(int a, int b, int arr[])
{

    // Store the greater element at
    // 0th index of the array
    if (a > b) {
        arr[0] = a;
        arr[1] = b;
    }
    else {
        arr[0] = b;
        arr[1] = a;
    }
}

// Driver code
int main()
{
```

```
    int x, y;
    int arr[2];

    printf("Enter two numbers: \n");
    scanf("%d%d", &x, &y);

    findGreaterSmaller(x, y, arr);

    printf("\nThe greater number is %d and the"
           "smaller number is %d",
           arr[0], arr[1]);

    return 0;
}
```

Output:

```
Enter two numbers:
5 8
The greater number is 8 and the smaller number is 5
```

C++ Only Methods

1. **Returning multiple values Using References:** We use references in C++ to store returned values.

```
// Modified program using References in C++
#include <stdio.h>

void compare(int a, int b, int &add_great, int &add_small)
{
    if (a > b) {
        add_great = a;
        add_small = b;
    }
    else {
        add_great = b;
        add_small = a;
    }
}

// Driver code
int main()
{
    int great, small, x, y;

    printf("Enter two numbers: \n");
    scanf("%d%d", &x, &y);

    // The last two arguments are passed
    // by giving addresses of memory locations
    compare(x, y, great, small);
    printf("\nThe greater number is %d and the"
           "smaller number is %d",
           great, small);

    return 0;
}
```

Output:

```
Enter two numbers:
5 8
The greater number is 8 and the smaller number is 5
```

2. **Returning multiple values using Class and Object :** The idea is similar to structures. We create a class with two integer variables and store the greater and smaller values into those variable, then use the values of that structure.

```
// Modified program using class
#include <stdio.h>

class GreaterSmaller {
public:
    int greater, smaller;
};

GreaterSmaller findGreaterSmaller(int a, int b)
{
    GreaterSmaller s;
    if (a > b) {
        s.greater = a;
        s.smaller = b;
```

```
    }
    else {
        s.greater = b;
        s.smaller = a;
    }

    return s;
}

// Driver code
int main()
{
    int x, y;
    GreaterSmaller result;

    printf("Enter two numbers: \n");
    scanf("%d%d", &x, &y);

    // The last two arguments are passed
    // by giving addresses of memory locations
    result = findGreaterSmaller(x, y);
    printf("\nThe greater number is %d and the"
           "smaller number is %d",
           result.greater, result.smaller);

    return 0;
}
```

Output:

```
Enter two numbers:
5 8
The greater number is 8 and the smaller number is 5
```

Recommended Posts:

How can I return multiple values from a function?

C function argument and return values

Return values of printf() and scanf() in C/C++

std::tuple, std::pair | Returning multiple values from a function using Tuple and Pair in C++

Function overloading and return type

How to return a local array from a C/C++ function?

Return from void functions in C++

Implicit return type int in C

Using return value of cin to take unknown number of inputs in C++

return statement vs exit() in main()

What is return type of getchar(), fgetc() and getc() ?

Multiple Inheritance in C++

Find the multiple of x which is closest to a^b

Assigning multiple characters in an int in C language

Class template with multiple parameters

Amrutesh99
Check out this Author's contributed articles.

If you like GeeksforGeeks and would like to contribute, you can also write an article using contribute.geeksforgeeks.org or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please Improve this article if you find anything incorrect by clicking on the "Improve Article" button below.

Article Tags : C C Programs C++ C++ Programs C-Arrays C-Functions C-Pointers C-Structure & Union CPP-Functions cpp-pointer cpp-references

Practice Tags : C CPP

7

To-do Done

2.2

Based on **7** vote(s)

Feedback/ Suggest Improvement Add Notes Improve Article

Please write to us at contribute@geeksforgeeks.org to report any issue with the above content.

Writing code in comment? Please use ide.geeksforgeeks.org, generate link and share the link here.

Load Comments Share this post!

GeeksforGeeks
A computer science portal for geeks

5th Floor, A-118,
Sector-136, Noida, Uttar Pradesh - 201305
feedback@geeksforgeeks.org

COMPANY
About Us
Careers
Privacy Policy
Contact Us

LEARN
Algorithms
Data Structures
Languages
CS Subjects
Video Tutorials

PRACTICE
Company-wise
Topic-wise
Contests
Subjective Questions

CONTRIBUTE
Write an Article
Write Interview Experience
Internships
Videos

@geeksforgeeks, Some rights reserved