

**Mr. Giansante**



**C++ Programming**  
**Functions**

**August 2018**

# Functions in C++

Source:

[https://www.tutorialspoint.com/cplusplus/cpp\\_functions.htm](https://www.tutorialspoint.com/cplusplus/cpp_functions.htm)

## What is a Function?

A function is a group of statements that together perform a task.

We can write a function and then call it multiple times without having to write it all over again.

Functions are sometimes called various names such as: procedure, sub-routine, method, etc.

Every C++ program has at least one function, which is `main()`

## Why Use Functions?

Functions are helpful for several reasons.

They make your programs more readable since specific tasks are separated into logical chunks of code. Many programs have thousands of lines of code and putting all this in the `main()` function makes it hard to read, understand and debug.

In addition, functions are useful for when we wanted to perform the same command sequences at multiple places of our program. Without functions, we'd have to copy the code over again or to jump from one place in the program to another. Both of these options are very confusing. For this reason, using functions keeps your code shorter.

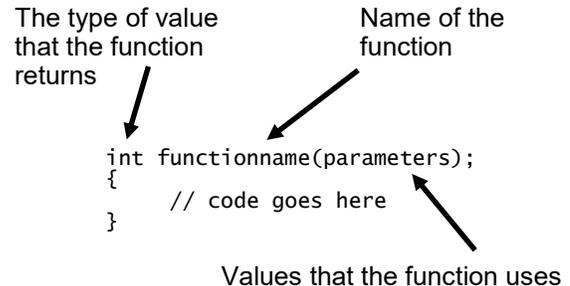
## Functional Decomposition

We sometimes refer to dividing a program into multiple functions as functional decomposition.

Think of this as divide and conquer.

## Declaring Functions

Functions are declared in the global scope, somewhere above the `main()` function.



Note: If the function does not return a value, place the word `void` in front of the function name.

A function can have no parameters, one parameter or multiple parameters.

The `return` command returns the value specified and then ends the function.

The `main()` function is defined to return an integer ...

```
int main()
```

... and the command that ends the `main()` function is `return 0;`

## Functions Example 1

The function `greet()` prints a message to the console and does not return any value.

```
#include <iostream>  
using namespace std;  
  
void greet()  
{  
    cout << "Hi, welcome!\n";  
}  
  
int main()  
{  
    greet();  
  
    return 0;  
}
```

# Functions with Parameters

## Functions with a Parameter

We will modify the `greet()` function so that it takes a parameter.

```
#include <iostream>
using namespace std;

void greet(string name)
{
    cout << "Hi, welcome " << name << "!\n";
}

int main()
{
    greet("Bob");
    greet("Susan");
    greet("John");

    return 0;
}
```

The following console output would be created:

```
Hi, welcome Bob!
Hi, welcome Susan!
Hi, welcome John!
```

## Function that Returns a Value

We will create a function named `rectangle_area` that returns the area of the rectangle. The parameters of the function will be two numbers: length and width.

```
#include <iostream>
using namespace std;

void rectangle_area(int length, int width)
{
    int result;
    result = length * width;
    return result;
}

int main()
{
    cout << rectangle_area(6, 10);
    return 0;
}
```

## Example

Write a function called `max()` that takes two parameters `num1` and `num2` and returns the biggest of the two.

```
#include <iostream>
using namespace std;

void max(int num1, int num2)
{
    // function that returning the max
    // between two numbers
    // local variable declaration
    int result;

    if (num1 > num2)
    {
        result = num1;
    }
    else
    {
        result = num2;
    }

    return result;
}

int main()
{
    cout << max(37, 11);
    return 0;
}
```