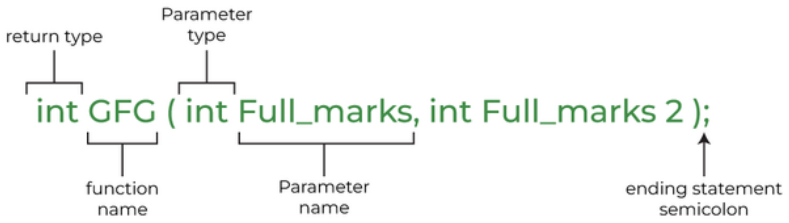
Functions in C++

A function is a set of statements that take inputs, do some specific computation, and produce output. The idea is to put some commonly or repeatedly done tasks together and make a **function** so that instead of writing the same code again and again for different inputs, we can call the function.

In simple terms, a function is a block of code that only runs when it is called.

Syntax:



Example:

```
// C++ Program to demonstrate working of a function
#include <iostream>
using namespace std;

// Following function that takes two parameters 'x' and 'y'
// as input and returns max of two input numbers
int max(int x, int y)
{
    if (x > y)
        return x;
    else
    return y;
}

// main function that doesn't receive any parameter and
// returns integer
int main()
{
```

```
    int a = 10, b = 20;

    // Calling above function to find max of 'a' and 'b'
    int m = max(a, b);

    cout << "m is " << m;
    return 0;
}
```

Output

```
m is 20
```

Explanation: - As the maximum of 10 and 20 is 20 so, m will become 20.

Why Do We Need Functions?

- Functions help us in ***reducing code redundancy***. If functionality is performed at multiple places in software, then rather than writing the same code, again and again, we create a function and call it everywhere. This also helps in maintenance as we have to change at one place if we make future changes to the functionality.
- Functions make code ***modular***. Consider a big file having many lines of code. It becomes really simple to read and use the code if the code is divided into functions.
- Functions provide ***abstraction***. For example, we can use library functions without worrying about their internal work.