C++ Functions

In programming, function refers to a segment that groups code to perform a specific task.

Depending on whether a function is predefined or created by programmer; there are two types of function:

1. Library Function
2. User-defined Function

## Library Function

Library functions are the built-in function in C++ programming.

### Example 1: Library Function

#include <iostream>

#include <cmath>

using namespace std;

int main()

{

double number, squareRoot;

cout << "Enter a number: ";

cin >> number;

// sqrt() is a library function to calculate square root

squareRoot = sqrt(number);

cout << "Square root of " << number << " = " << squareRoot;

return 0;

}

**Output**

Enter a number: 26

Square root of 26 = 5.09902

In the example above, sqrt() library function is invoked to calculate the square root of a number.

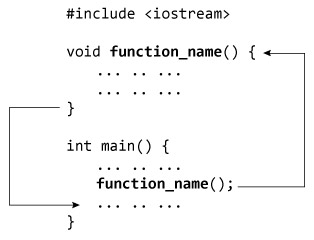
## User-defined Function

C++ allows programmer to define their own function.

A user-defined function groups code to perform a specific task and that group of code is given a name(identifier).

When the function is invoked from any part of program, it all executes the codes defined in the body of function.

### How user-defined function works in C Programming?



Consider the figure above.

When a program begins running, the system calls the main() function, that is, the system starts executing codes from main() function.

When control of the program reaches to function\_name() inside main(), it moves to void function\_name() and all codes inside void function\_name() is executed.

Then, control of the program moves back to the main function where the code after the call to the function\_name() is executed as shown in figure above.

### Example 2: User Defined Function

**C++ program to add two integers. Make a function**add()**to add integers and display sum in main() function.**

#include <iostream>

using namespace std;

// Function prototype (declaration)

int add(int, int);

int main()

{

int num1, num2, sum;

cout<<"Enters two numbers to add: ";

cin >> num1 >> num2;

// Function call

sum = add(num1, num2);

cout << "Sum = " << sum;

return 0;

}

// Function definition

int add(int a, int b)

{

int add;

add = a + b;

// Return statement

return add;

}

**Output**

Enters two integers: 8

-4

Sum = 4

### Function prototype (declaration)

If a user-defined function is defined after main() function, compiler will show error. It is because compiler is unaware of user-defined function, types of argument passed to function and return type.

In C++, function prototype is a declaration of function without its body to give compiler information about user-defined function. Function prototype in the above example is:

int add(int, int);

You can see that, there is no body of function in prototype. Also, there are only return type of arguments but no arguments. You can also declare function prototype as below but it's not necessary to write arguments.

int add(int a, int b);

**Note:** It is not necessary to define prototype if user-defined function exists before main()function.

### Function Call

To execute the codes of function body, the user-defined function needs to be invoked(called).

In the above program, add(num1,num2); inside main() function calls the user-defined function.

The function returns an integer which is stored in variable add.