C++ Functions

The function in C++ language is also known as procedure or subroutine in other programming languages.

To perform any task, we can create function. A function can be called many times. It provides modularity and code reusability.

Advantage of functions in C

There are many advantages of functions.

1) Code Reusability

By creating functions in C++, you can call it many times. So we don't need to write the same code again and again.

2) Code optimization

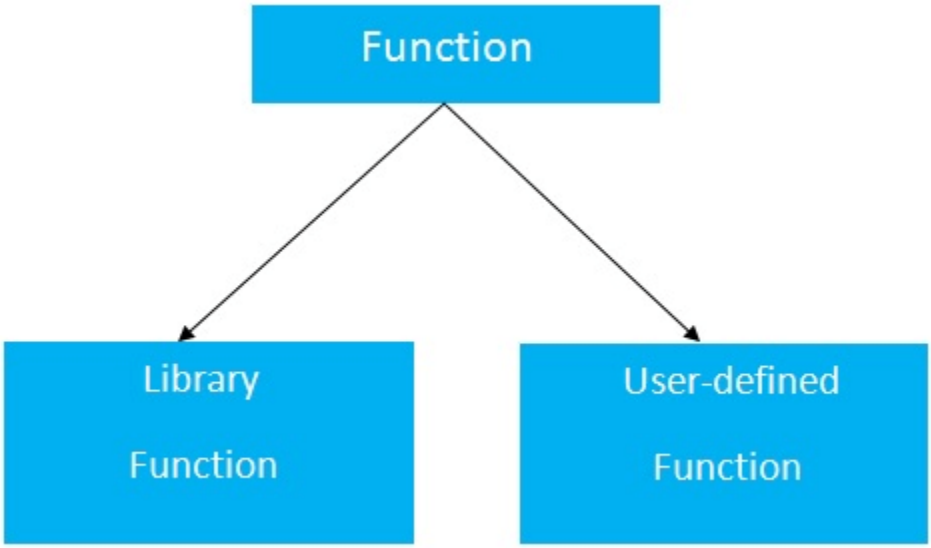
It makes the code optimized, we don't need to write much code.

Types of Functions

There are two types of functions in C programming:

1. Library Functions: are the functions which are declared in the C++ header files such as sqrt(x), cos(x), exp(x), etc.

2. User-defined functions: are the functions which are created by the C++ programmer, so that he/she can use it many times. It reduces complexity of a big program and optimizes the code.



Declaration of a function

The syntax of creating function in C++ language is given below:

```
1. return_type   function_name(data_type parameter...)
2. {
3. //code to be executed
4. }
```

C++ Function Example

Task 1: Write Simple Function.

Function0.cpp

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

// declaring a function
void greet() {
    cout << "Hello there!";
}

int main() {

    // calling the function
    greet();

    return 0;
}
```

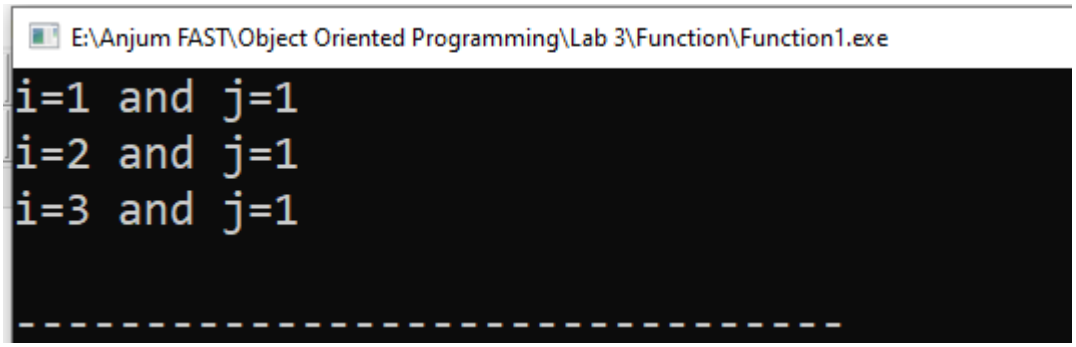
Output:

```
Hello there!
--------------------------------
```

Task 2: Let's see the another simple example of C++ function.?

```
1  #include<iostream>
2  using namespace std;
3  void func() {
4      static int i=0; //static variable
5      int j=0; //local variable
6      i++;
7      j++;
8      cout<<"i=" << i<<" and j=" <<j<<endl;
9  }
10 int main() {
11     func();
12     func();
13     func();
14     return 0;
15 }
16
```

Output:



Task 3: Write a function which will add two Numbers?

Function0.cpp | Add2Number.cpp

```cpp
// program to add two numbers using a function
#include <iostream>
using namespace std;

// declaring a function
int add(int a, int b) {
    return (a + b);
}

int main() {
    int sum;
    // calling the function and storing
    // the returned value in sum
    sum = add(100, 78);
    cout << "100 + 78 = " << sum << endl;
    return 0;
}
```

Output:

```
100 + 78 = 178
```

Task 4: Function Prototype

In C++, the code of function declaration should be before the function call. However, if we want to define a function after the function call, we need to use the function prototype. For example,

Function0.cpp | Add2Number.cpp | FunctionPrototype.cpp

```cpp
// using function definition after main() function
// function prototype is declared before main()
#include <iostream>
using namespace std;

// function prototype
int add(int, int);

int main() {
    int sum;
    // calling the function and storing
    // the returned value in sum
    sum = add(100, 78);
    cout << "100 + 78 = " << sum << endl;
    return 0;
}

// function definition
int add(int a, int b) {
    return (a + b);
}
```

```
100 + 78 = 178
```

Task 5: Write a function which will calculate Maximum Number?

Function1.cpp | CallByValue.cpp | CallByReferenc.cpp | FindMax.cpp

```cpp
#include <iostream>
using namespace std;
// function declaration
int max(int num1, int num2);
int main () {
// local variable declaration:
    int a = 100;
    int b = 200;
    int ret;
// calling a function to get max value.
    ret = max(a, b);
    cout << "Max value is : " << ret << endl;
    return 0;
}
// function returning the max between two numbers
int max(int num1, int num2) {
// local variable declaration
    int result;
    if (num1 > num2)
        result = num1;
    else
        result = num2;
    return result;
}
```

Output:

```
Max value is : 200
```

Task 6: Write a function which will calculate Square Root?

Function0.cpp | Add2Number.cpp | FunctionPrototype.cpp | FindSqrt.cpp

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

int main() {
    double number, squareRoot;

    number = 25.0;

    // sqrt() is a library function to calculate the square root
    squareRoot = sqrt(number);

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

    return 0;
}
```

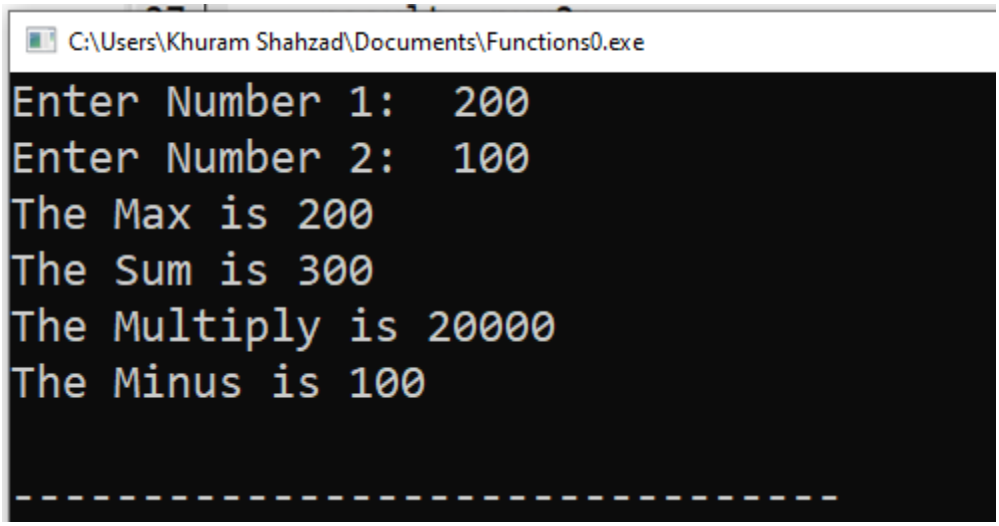
```
Square root of 25 = 5
```

Task 7: Write a function which will calculate Sum, Minus, Max and Multiply of 2 numbers?

[*] Functions0.cpp

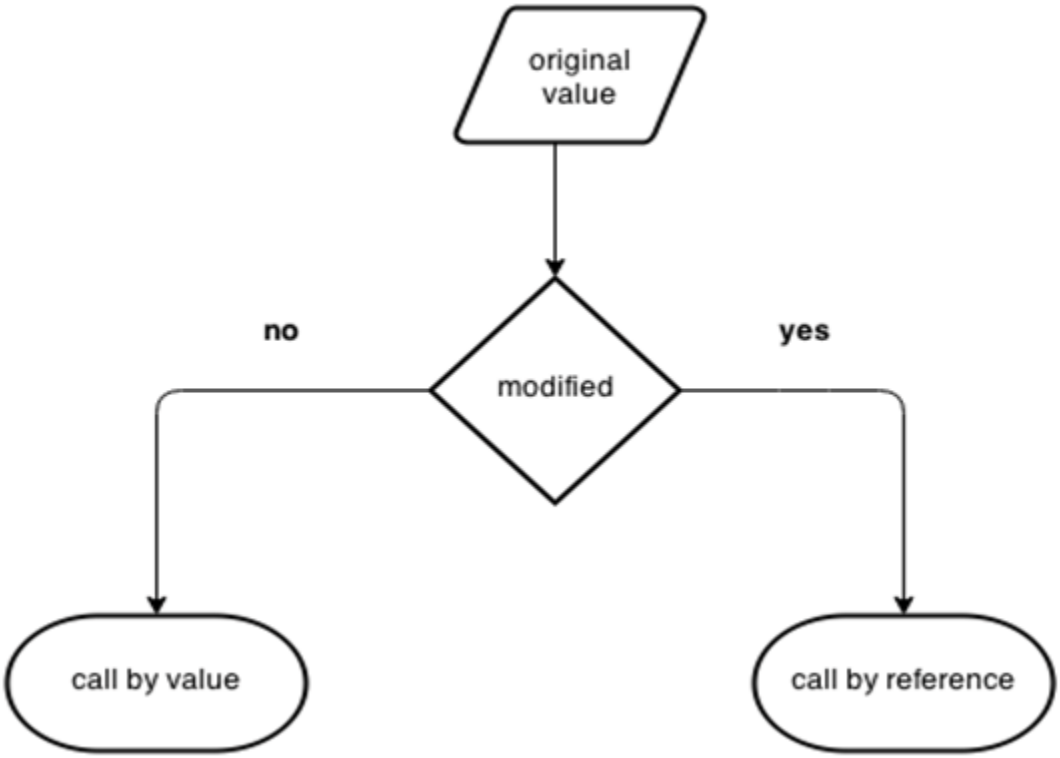
```cpp
#include <iostream>
using namespace std;
int max (int num1, int num2);
int Min (int num1, int num2);
int Multiply (int num1, int num2);
int sum (int num1, int num2);
 main ()
{
    int num1, num2;
    cout<<"Enter Number 1:  ";
    cin>>num1;

    cout<<"Enter Number 2:  ";
    cin>>num2;
    cout<<"The Max is "<< max(num1, num2)<<endl;
    cout<<"The Sum is "<< sum(num1, num2)<<endl;
    cout<<"The Multiply is "<< Multiply(num1, num2)<<endl;
    cout<<"The Minus is "<< Min(num1, num2)<<endl;

}
int max (int num1, int num2)
{
    int result=0;
    if(num1>num2)
    result =num1;
    else
    result =num2;

```

```
21  int max (int num1, int num2)
22  {
23      int result=0;
24      if(num1>num2)
25      result =num1;
26      else
27      result =num2;
28
29      return result;
30  }
31  int sum (int num1, int num2)
32  {
33      return (num1+num2);
34  }
35  int Multiply (int num1, int num2)
36  {
37      return (num1*num2);
38  }
39
40  int Min (int num1, int num2)
41  {
42      return (num1-num2);
43  }
```


C:\Users\Khuram Shahzad\Documents\Functions0.exe
Enter Number 1: 200
Enter Number 2: 100
The Max is 200
The Sum is 300
The Multiply is 20000
The Minus is 100

Call by value and call by reference in C++

There are two ways to pass value or data to function in C language: call by value and call by reference. Original value is not modified in call by value but it is modified in call by reference.



Let's understand call by value and call by reference in C++ language one by one.

Call by value in C++

In call by value, **original value is not modified.**

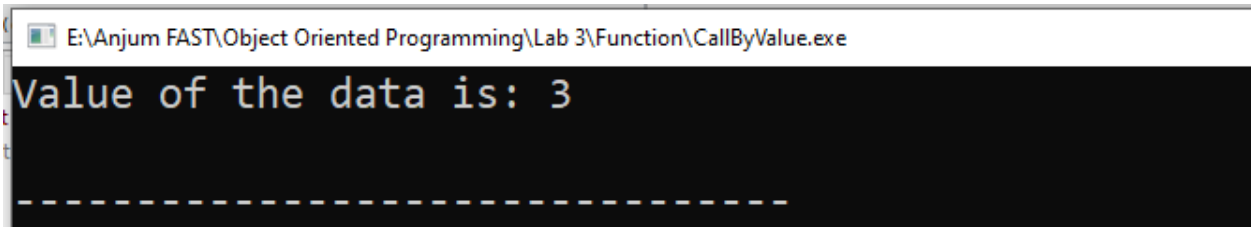
In call by value, value being passed to the function is locally stored by the function parameter in stack memory location. If you change the value of function parameter, it is changed for the current function only. It will not change the value of variable inside the caller method such as main().

Let's try to understand the concept of call by value in C++ language by the example given below:

Function1.cpp | CallByValue.cpp

```cpp
#include <iostream>
using namespace std;
void change(int data);
int main() {
    int data = 3;
    change(data);
    cout << "Value of the data is: " << data<< endl;
    return 0;
}
void change(int data) {
    data = 5;
}
```

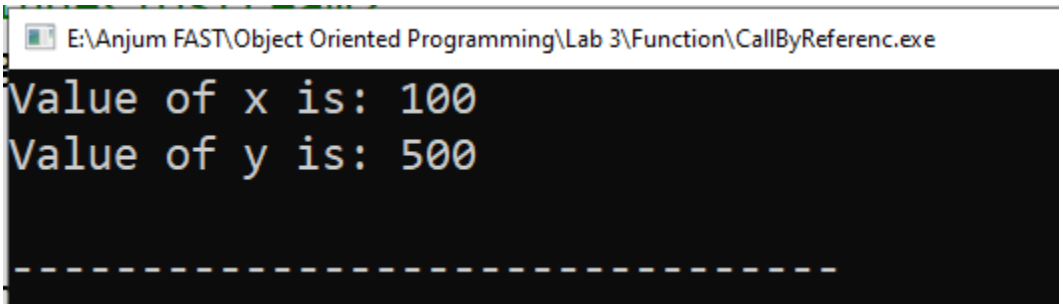
Output:



Function1.cpp | CallByValue.cpp | CallByReferenc.cpp

```cpp
#include<iostream>
using namespace std;
void swap(int *x, int *y)
{
 int swap;
 swap=*x;
 *x=*y;
 *y=swap;
}
int main()
{
 int x=500, y=100;
 swap(&x, &y);  // passing value to function
 cout<<"Value of x is: "<<x<<endl;
 cout<<"Value of y is: "<<y<<endl;
 return 0;
}
```

Output:



Difference between call by value and call by reference in C++

No.	Call by value	Call by reference
1	A copy of value is passed to the function	An address of value is passed to the function
2	Changes made inside the function is not reflected on other functions	Changes made inside the function is reflected outside the function also
3	Actual and formal arguments will be created in different memory location	Actual and formal arguments will be created in same memory location