Content 14

Function and function Prototypes C++

#### Functions in C++

Functions are the main part of top-down structured programming. We break the code into small pieces and make functions of that code. Functions help us to reuse the code easily.

#include <iostream>

using namespace std;

int sum(int a, int b)

{

    int s = a + b;

    return s;

}

int main()

{

    int n1, n2;

    cout << "Enter number 1: ";

    cin >> n1;

    cout << "Enter number 2: ";

    cin >> n2;

    cout << "\n\nThe sum of Numbers are: " << sum(n1, n2) << endl;

    return 0;

}

**Output:**

Enter number 1: 2

Enter number 2: 2

The sum of Numbers is: 4

#### Function Prototype in C++

The function prototype is the template of the function which tells the details of the function e.g(name, parameters) to the compiler. Function prototypes help us to define a function after the function call.

// Function prototype

int sum(int a, int b);

we have made a function prototype of the function “sum”, this function prototype will tell the compiler that the function “sum” is declared somewhere in the program which takes two integer parameters and returns an integer value. Some examples of acceptable and not acceptable prototypes are shown below:

* int sum(int a, int b); //Acceptable
* int sum(int a, b); // Not Acceptable
* int sum(int, int); //Acceptable

##### **Formal Parameters**

The variables which are declared in the function are called a formal parameter. For example, as shown in Code Snippet 1, the variables “a” and “b” are the formal parameters.

##### **Actual Parameters**

The values which are passed to the function are called actual parameters. For example, as shown in Code Snippet 2, the variables “num1” and “num2” are the actual parameters.

/\*For function prototype :- if we have written function a

fter main and calling i in main body then we to have say to compiler that the function

is present in the program but h(compiler) has to find it \*/

#include <iostream>

using namespace std;

int sum(int a, int b); //function prototype

int main()

{

    int n1, n2;

    cout << "Enter numder 1: ";

    cin >> n1;

    cout << "Enter numder 2: ";

    cin >> n2;

    cout << "\n\nThe sum of Numbers are: " << sum(n1, n2) << endl;

    return 0;

}

int sum(int a, int b)

{

    int s = a + b;

    return s;

}

If we don’t write the function prototype then it will throw an error:

**Output:**

Enter number 1: 2

Enter number 2: 2

The sum of Numbers is: 4