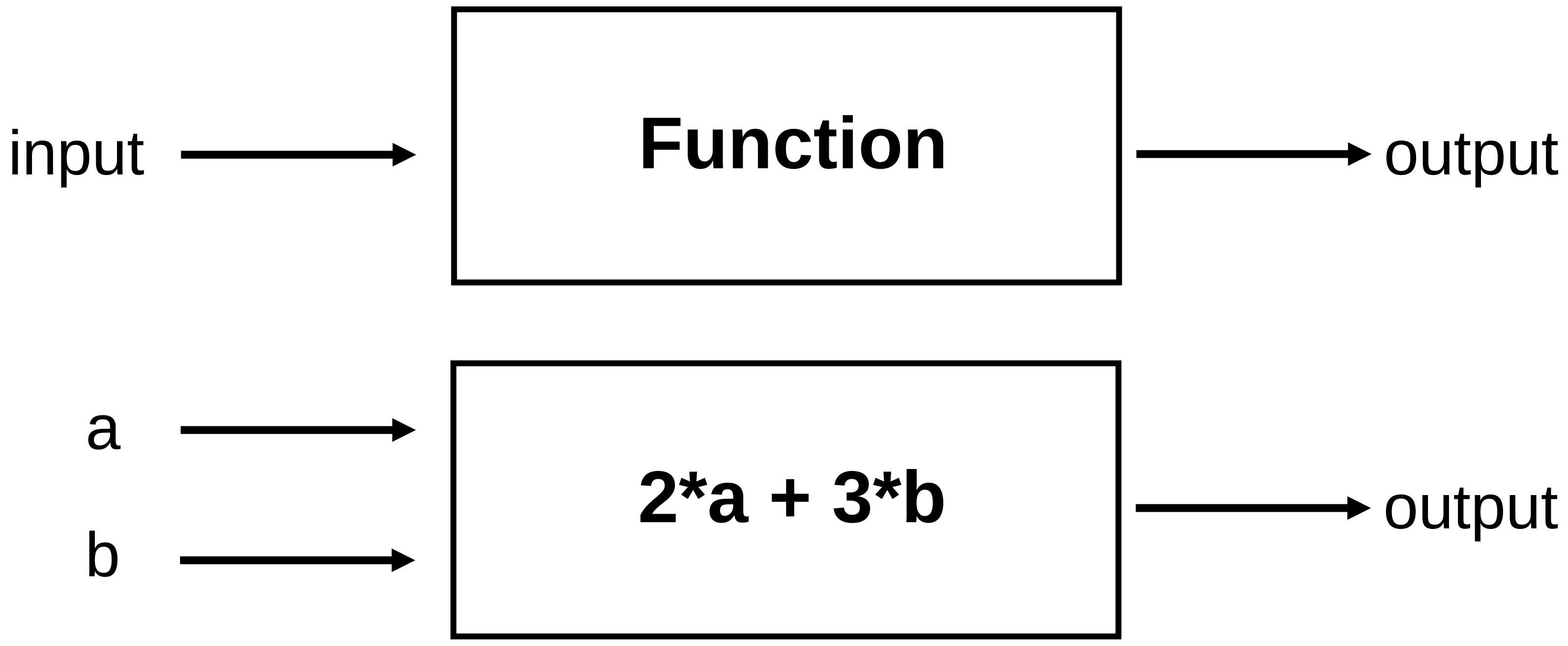
CC-301 Programming Fundamentals

Lecture 11

Engr. Waseem Ullah Khan

Functions



Output = f(a,b)

Functions

A function is a block of code which is used to performs a specific task

Used to divide a complex task into smaller tasks

Receive some information, do the processing and provide a result

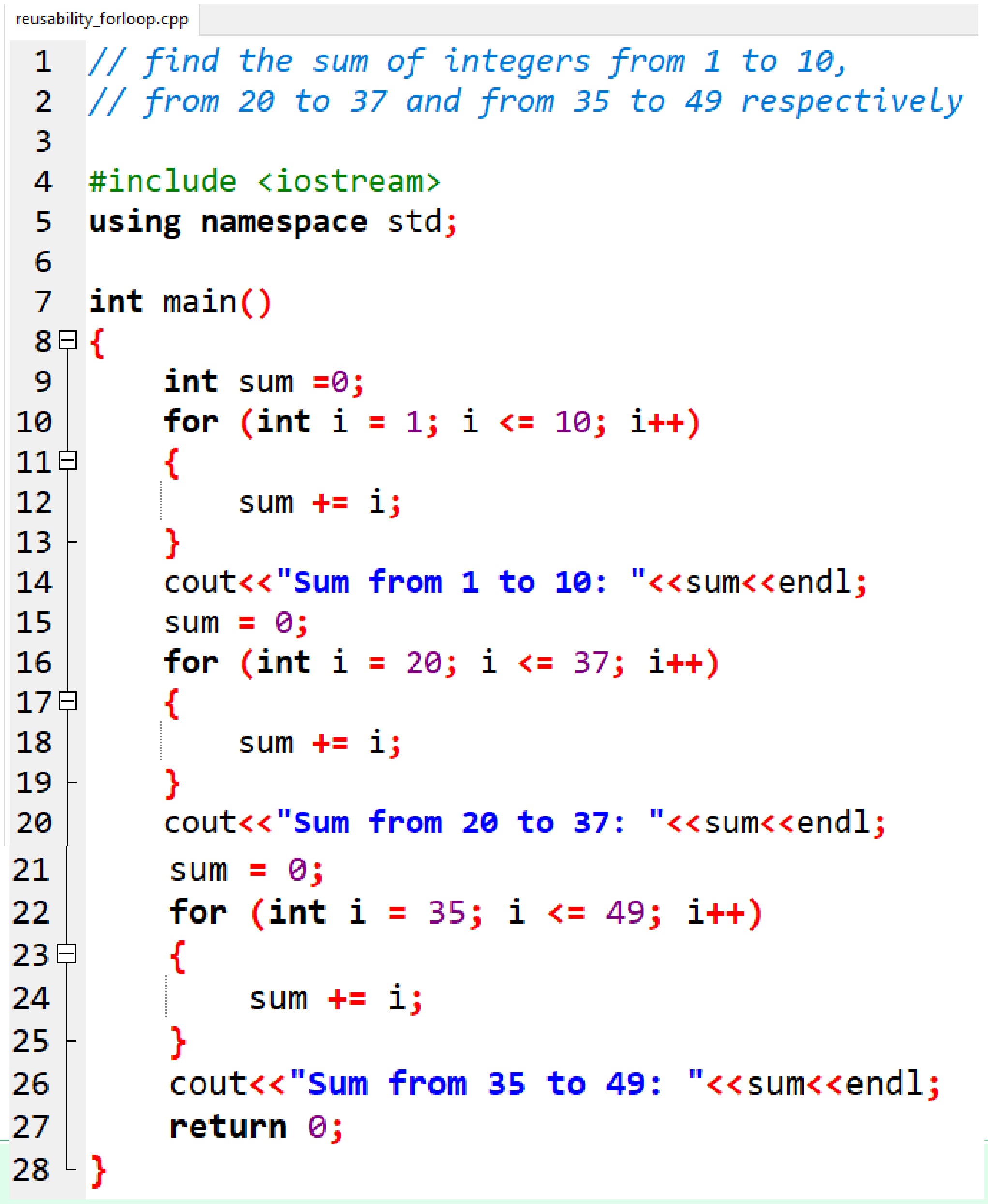
Types of functions: There are two types of function:

- **Standard Library Functions:** are the functions which are Predefined in C++
- **User-defined Function:** are the functions which are Created by users

Functions are used to provide modularity & code reusability to a program

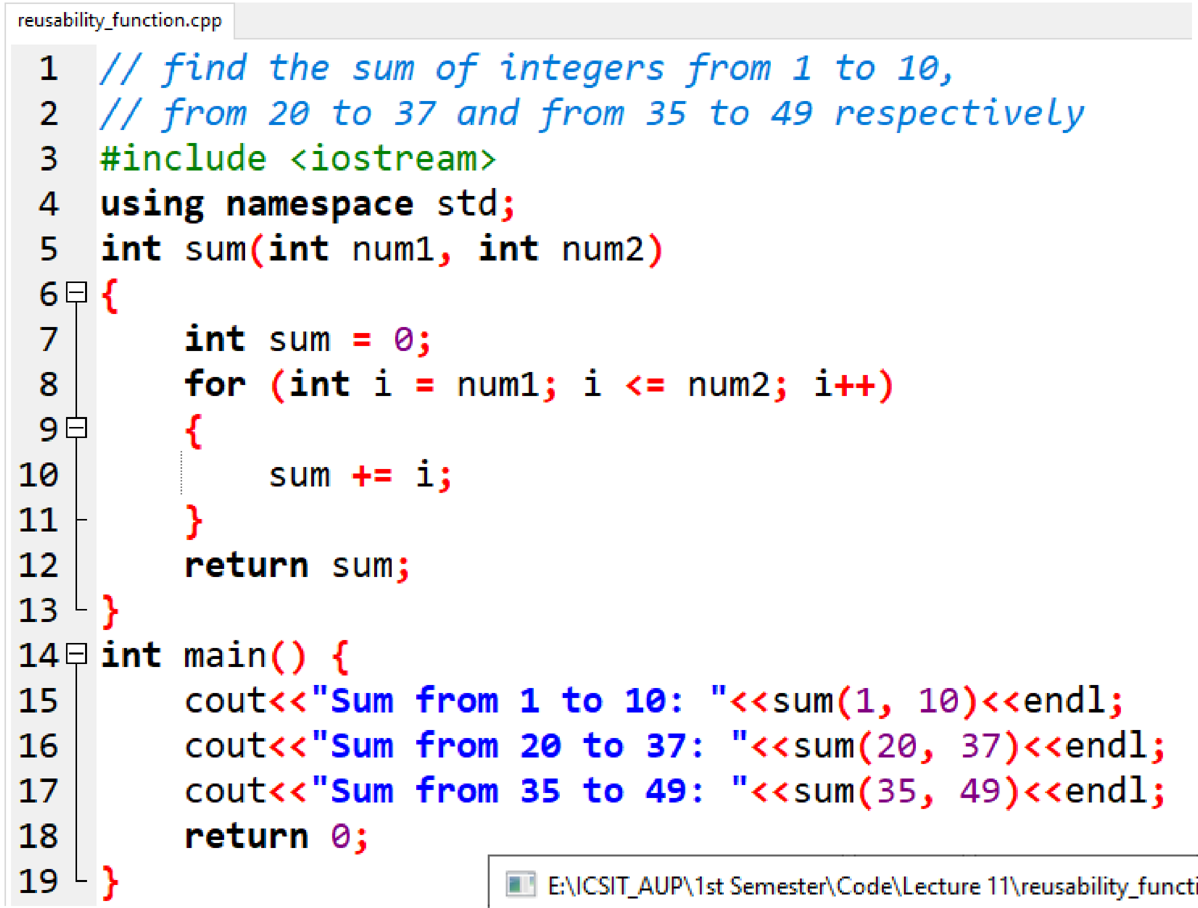
- For example: We need to find the sum of integers from 1 to 10, from 20 to 37 and from 35 to 49 respectively

Reusability - Example



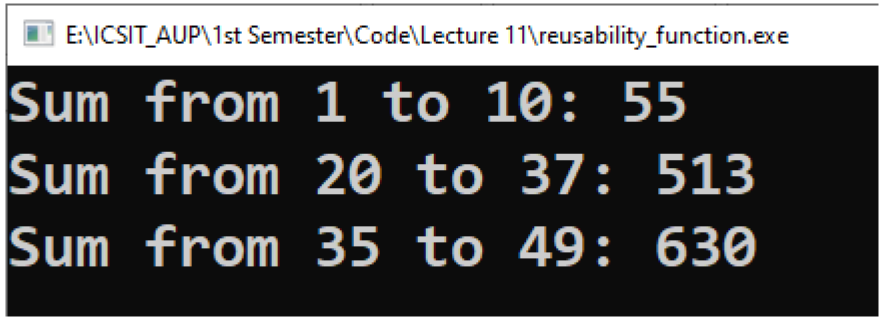
reusability_forloop.cpp

```cpp
// find the sum of integers from 1 to 10,
// from 20 to 37 and from 35 to 49 respectively

#include <iostream>
using namespace std;

int main()
{
    int sum =0;
    for (int i = 1; i <= 10; i++)
    {
        sum += i;
    }
    cout<<"Sum from 1 to 10: "<<sum<<endl;
    sum = 0;
    for (int i = 20; i <= 37; i++)
    {
        sum += i;
    }
    cout<<"Sum from 20 to 37: "<<sum<<endl;
    sum = 0;
    for (int i = 35; i <= 49; i++)
    {
        sum += i;
    }
    cout<<"Sum from 35 to 49: "<<sum<<endl;
    return 0;
}
```



reusability_function.cpp

```cpp
// find the sum of integers from 1 to 10,
// from 20 to 37 and from 35 to 49 respectively
#include <iostream>
using namespace std;
int sum(int num1, int num2)
{
    int sum = 0;
    for (int i = num1; i <= num2; i++)
    {
        sum += i;
    }
    return sum;
}
int main() {
    cout<<"Sum from 1 to 10: "<<sum(1, 10)<<endl;
    cout<<"Sum from 20 to 37: "<<sum(20, 37)<<endl;
    cout<<"Sum from 35 to 49: "<<sum(35, 49)<<endl;
    return 0;
}
```



E:\ICSIT_AUP\1st Semester\Code\Lecture 11\reusability_function.exe

```
Sum from 1 to 10: 55
Sum from 20 to 37: 513
Sum from 35 to 49: 630
```

Structure of a Function

Function Declaration

- Tells the compiler about a function name and how to call the function
- Tells about the number of parameters function takes, data-types of the parameters and the return type of the function

Calling Function

- Takes the program control to the called function

Function Definition or Function Body

- Contains the body of the function i.e. all the commands that make up the function

Structure of a Function

A function definition consists of its function name, function parameters, return value type and body

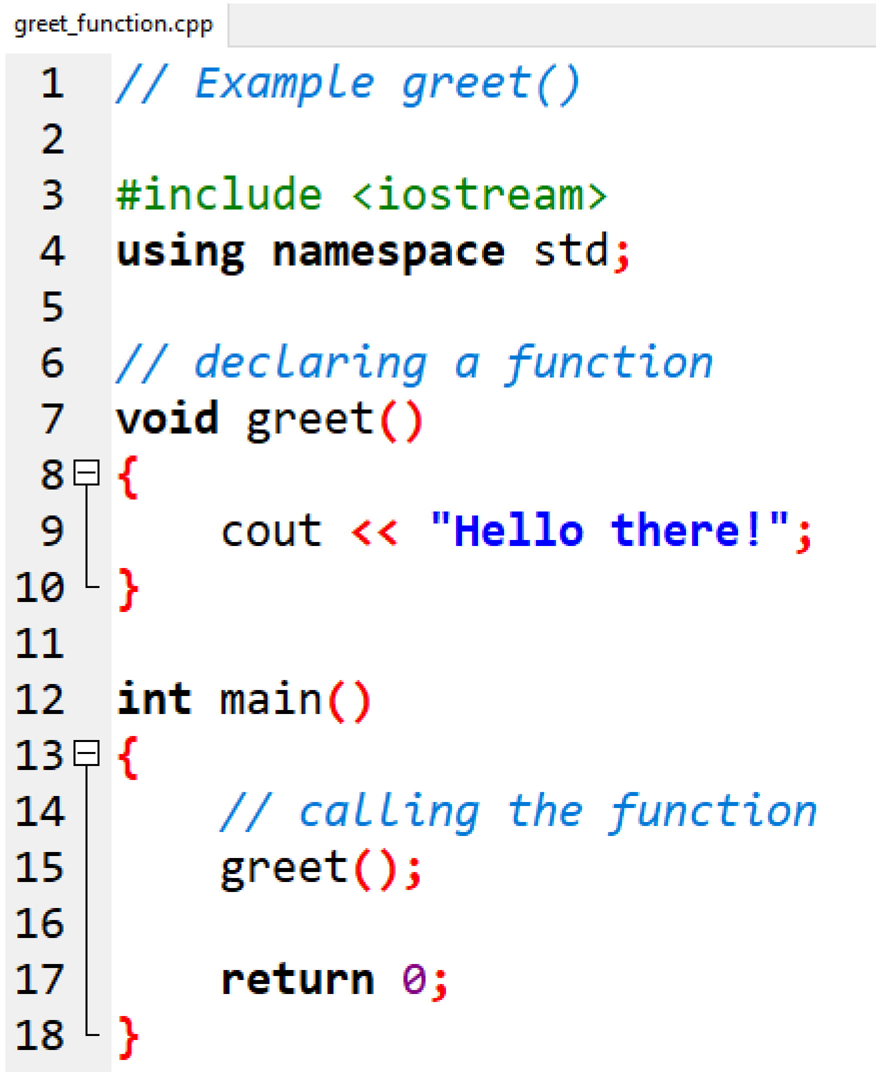
Syntax

```
// function declaration
returnType functionName (parameter1, parameter2,...)
{
    // function body
}


int main()
{
    // calling a function
    functionName();

}
```

Function Definition or Function Body

Function Call

Example



greet_function.cpp

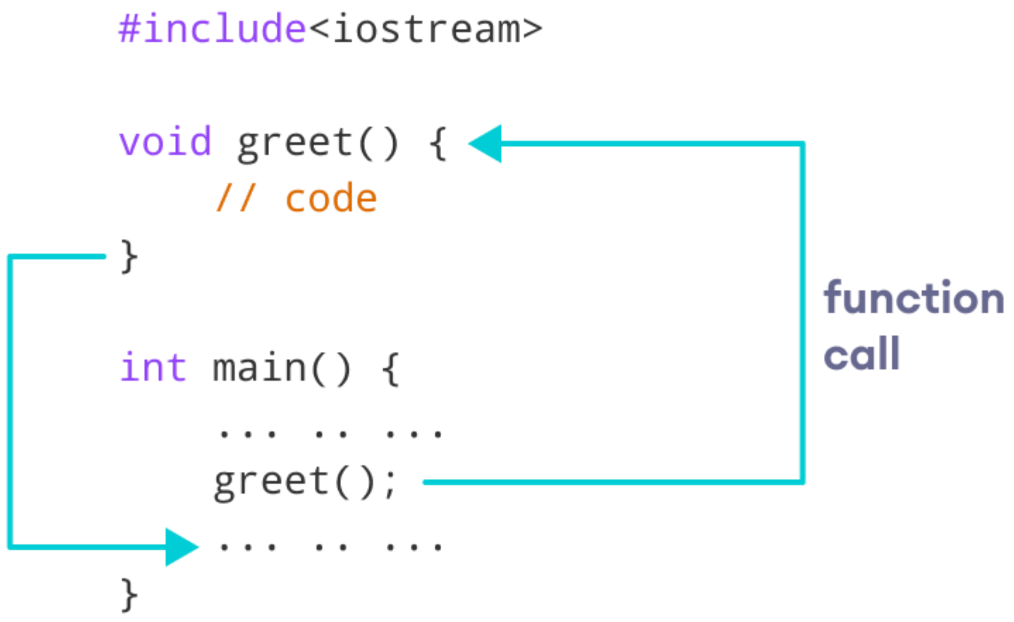
```cpp
// Example greet()

#include <iostream>
using namespace std;

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

int main()
{
    // calling the function
    greet();

    return 0;
}
```

The name of the function is greet()

The return type of the function is void

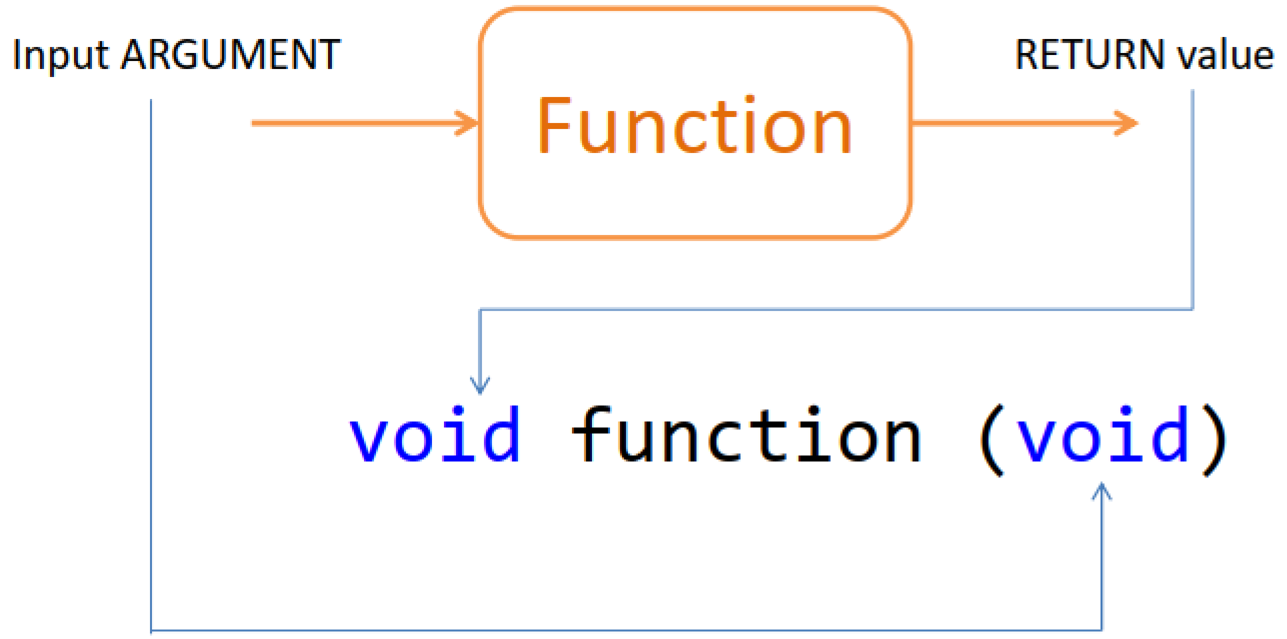
The empty parentheses mean it doesn't have any parameters

The function body is written inside {}



```cpp
#include<iostream>

void greet() {
    // code
}

int main() {
    ... .. ...
    greet();
    ... .. ...
}
```

Function



Functions with no input and no output

Functions with input but no output

Functions with no input but output

Functions with input and output

Functions with no Input and no Output

```
void function (void)
```

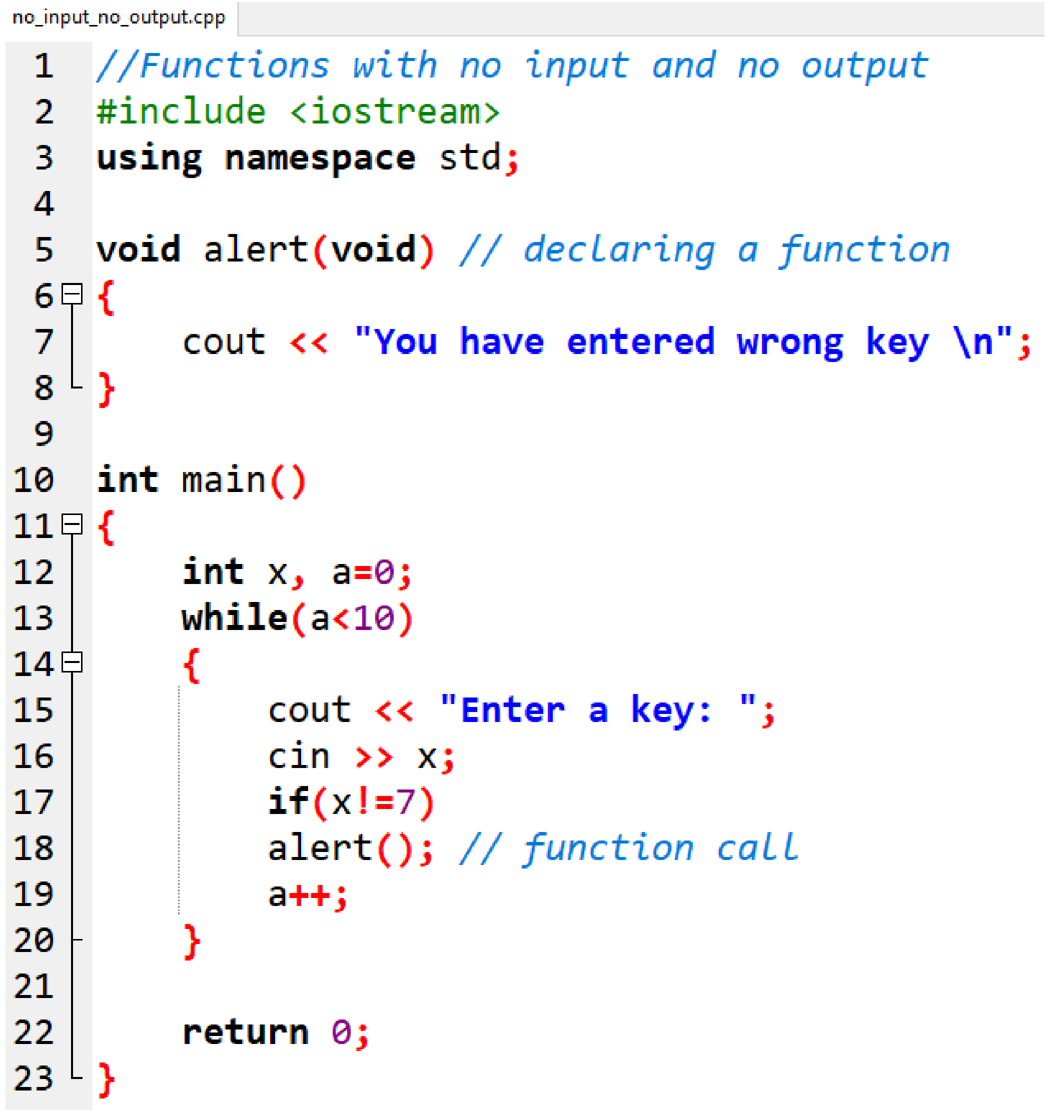
A void function does not return any value (no output)

It does not take any parameters

It performs a specific task

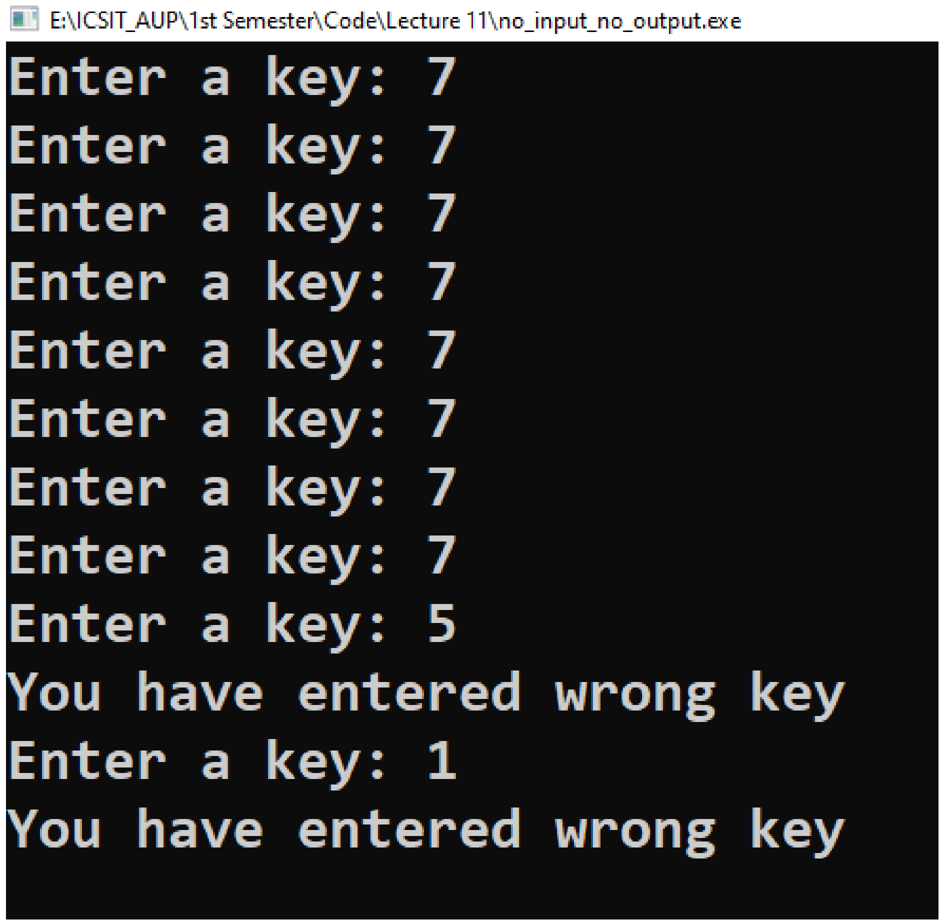
Functions with no Input and no Output

Example



no_input_no_output.cpp

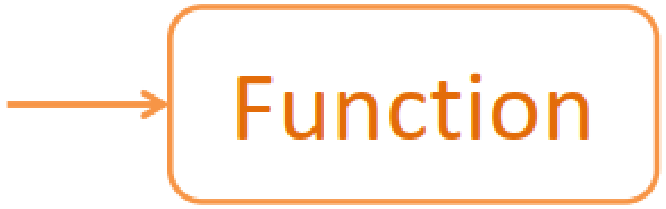
```cpp
//Functions with no input and no output
#include <iostream>
using namespace std;

void alert(void) // declaring a function
{
    cout << "You have entered wrong key \n";
}

int main()
{
    int x, a=0;
    while(a<10)
    {
        cout << "Enter a key: ";
        cin >> x;
        if(x!=7)
        alert(); // function call
        a++;
    }

    return 0;
}
```



E:\ICSIT_AUP\1st Semester\Code\Lecture 11\no_input_no_output.exe

```
Enter a key: 7
Enter a key: 7
Enter a key: 7
Enter a key: 7
Enter a key: 7
Enter a key: 7
Enter a key: 7
Enter a key: 7
Enter a key: 5
You have entered wrong key
Enter a key: 1
You have entered wrong key
```

Functions with Input but no Output



```
void function (int)
```

Such a function does not return any value

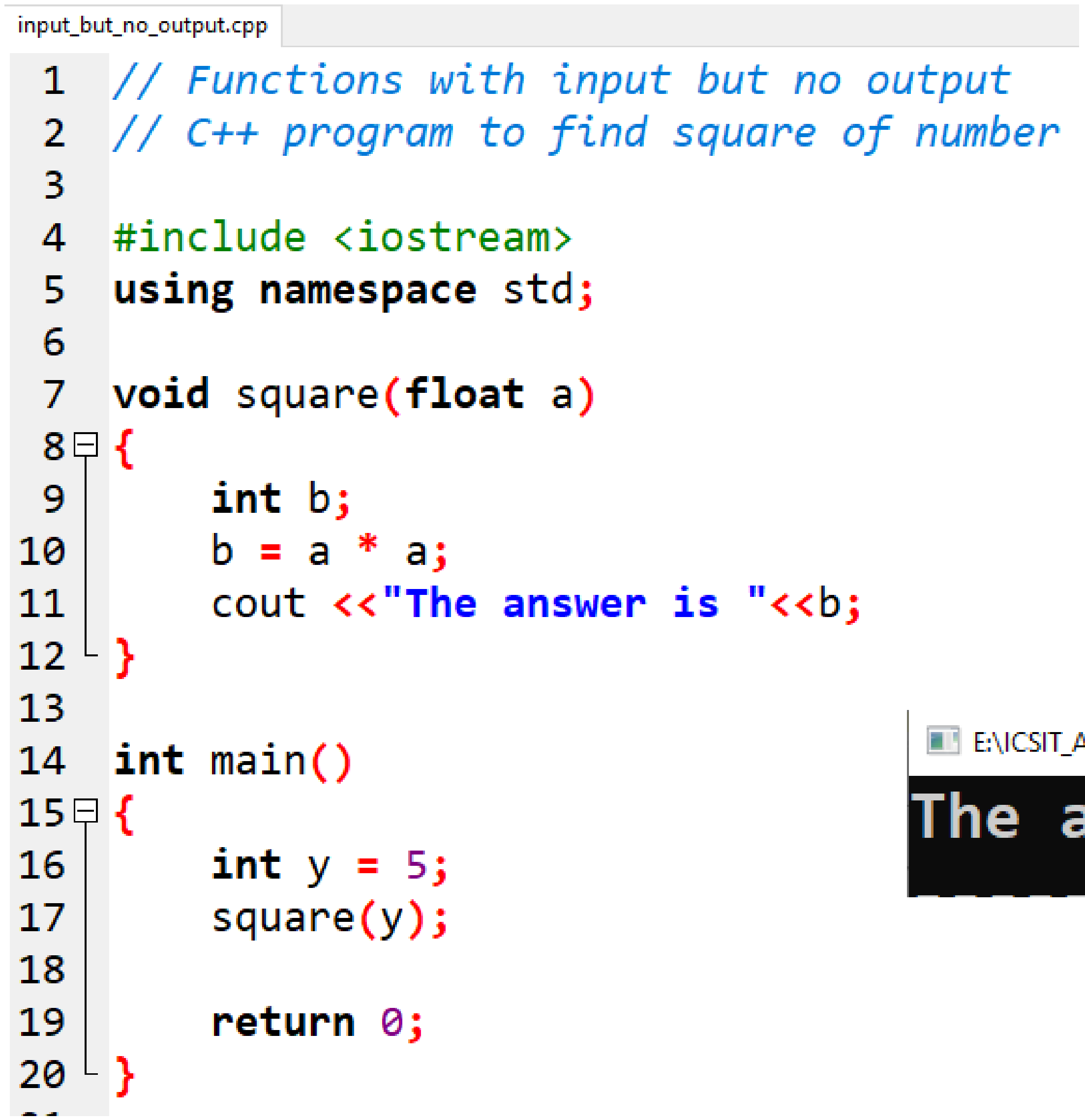
However, it takes input parameters

It performs a specific task

For example, it may perform some operation on the input data and simply display it on the screen

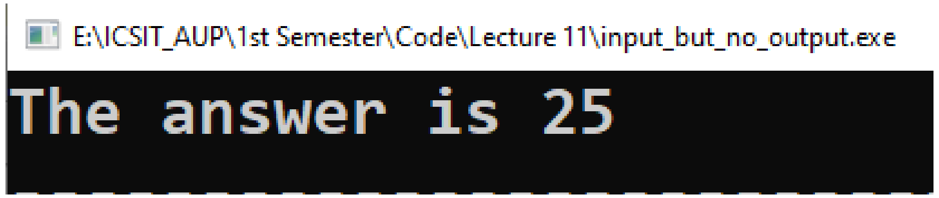
Functions with Input but no Output

Example



input_but_no_output.cpp

```cpp
// Functions with input but no output
// C++ program to find square of number

#include <iostream>
using namespace std;

void square(float a)
{
    int b;
    b = a * a;
    cout <<"The answer is "<<b;
}

int main()
{
    int y = 5;
    square(y);

    return 0;
}
```



E:\ICSIT_AUP\1st Semester\Code\Lecture 11\input_but_no_output.exe

```
The answer is 25
```

Functions with Output but no Input

```
int function (void)
```

Such a function returns a value

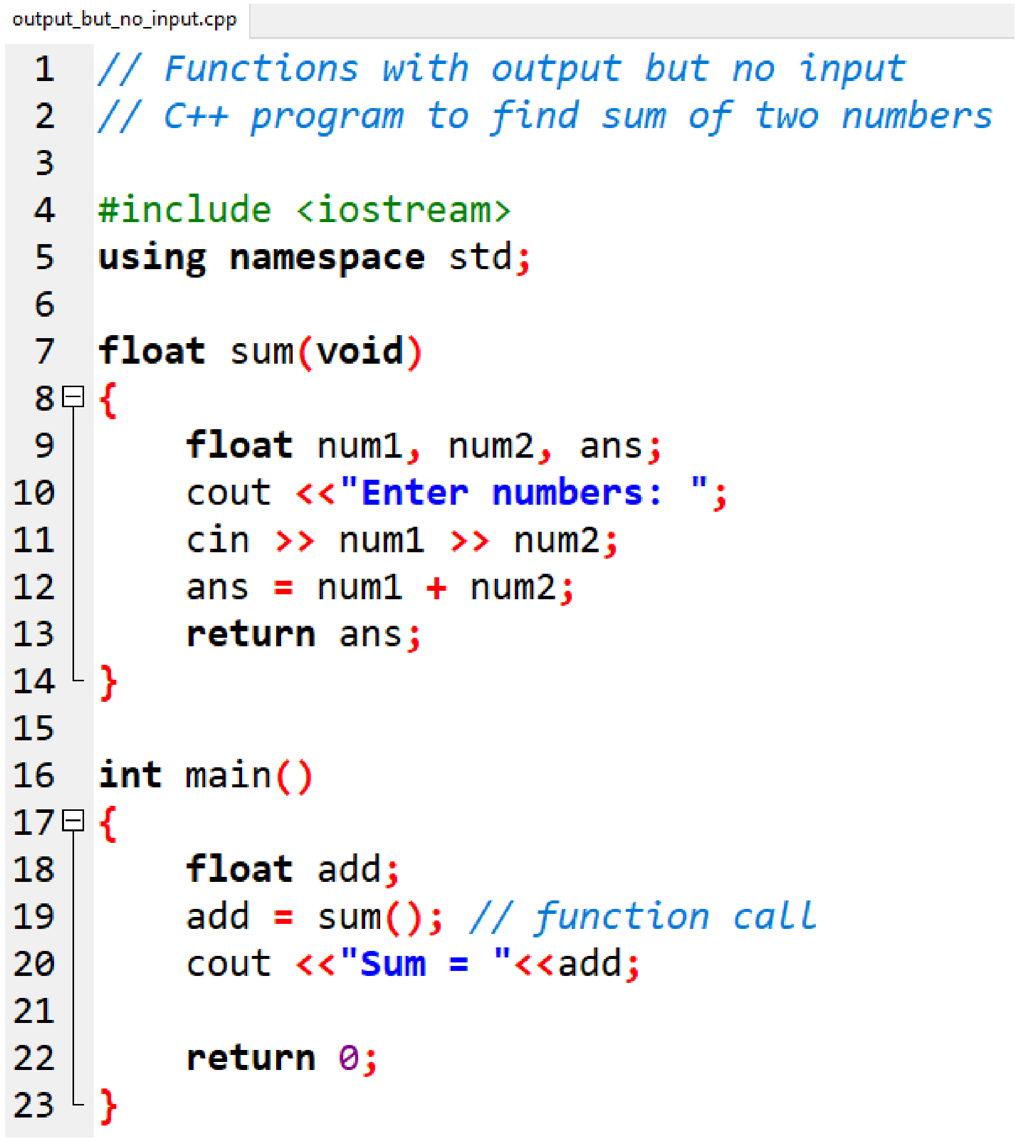
However it does not takes input parameters

It performs a specific task

For example, it may take input values from user inside the function body and return the result of the operation to main()

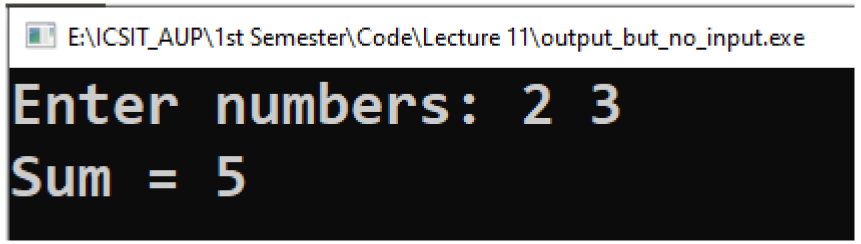
Functions with Output but no Input

Example



output_but_no_input.cpp

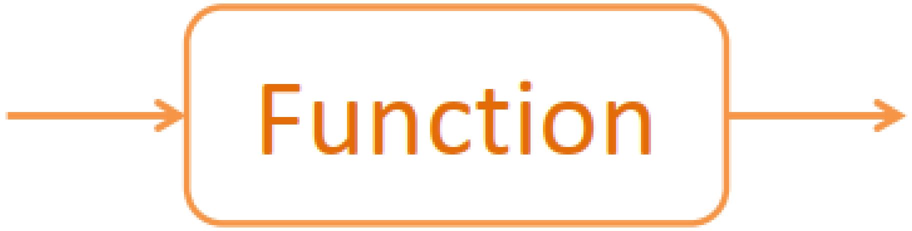
```cpp
// Functions with output but no input
// C++ program to find sum of two numbers

#include <iostream>
using namespace std;

float sum(void)
{
    float num1, num2, ans;
    cout <<"Enter numbers: ";
    cin >> num1 >> num2;
    ans = num1 + num2;
    return ans;
}

int main()
{
    float add;
    add = sum(); // function call
    cout <<"Sum = "<<add;

    return 0;
}
```



E:\ICSIT_AUP\1st Semester\Code\Lecture 11\output_but_no_input.exe

```
Enter numbers: 2 3
Sum = 5
```

Functions with both Input and Output



```
int function (int)
```

Such a function returns a value

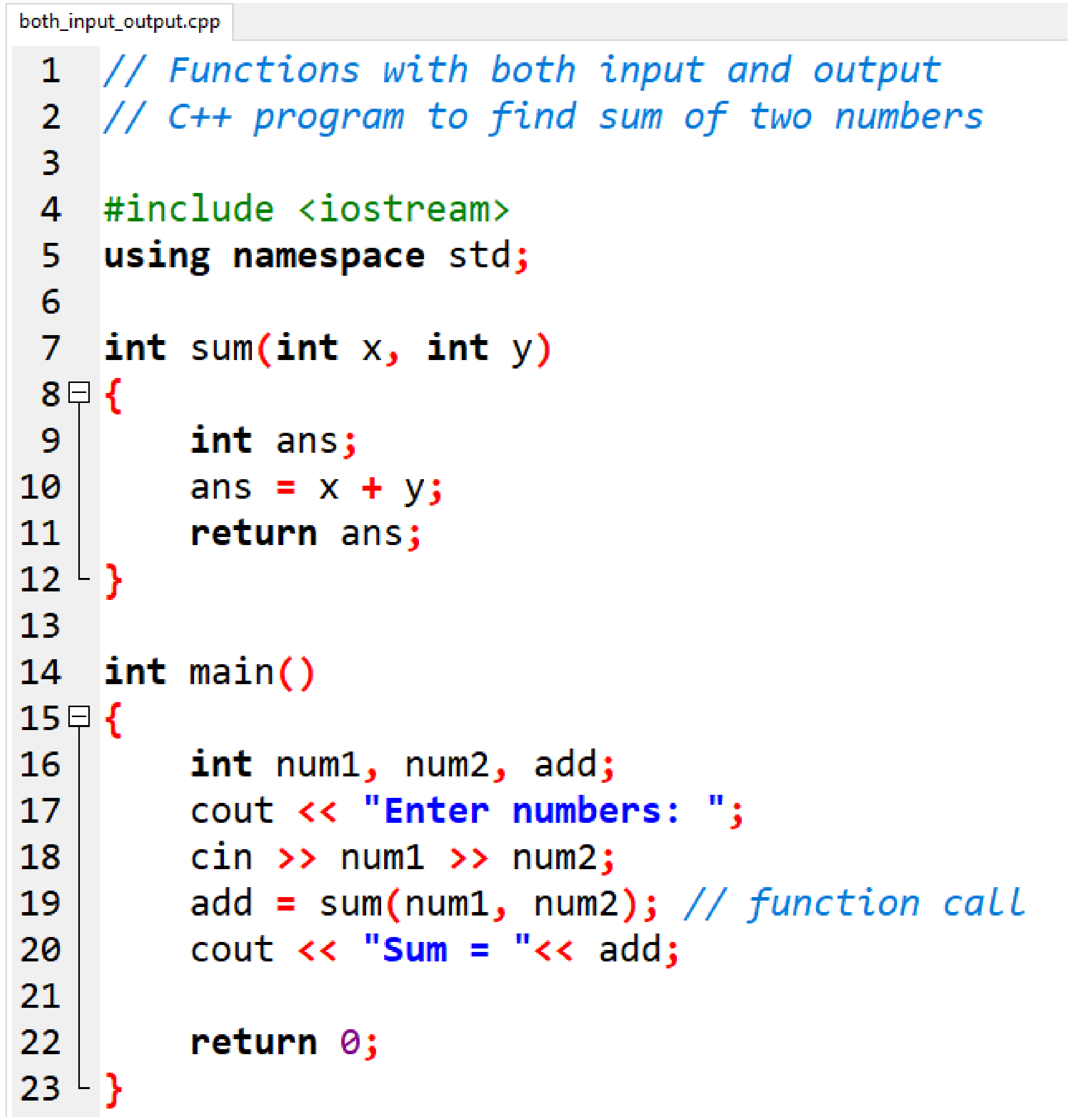
It also takes input parameter(s)

It performs a specific task

For example, it may take input values from main() and return the result of the operation to main()

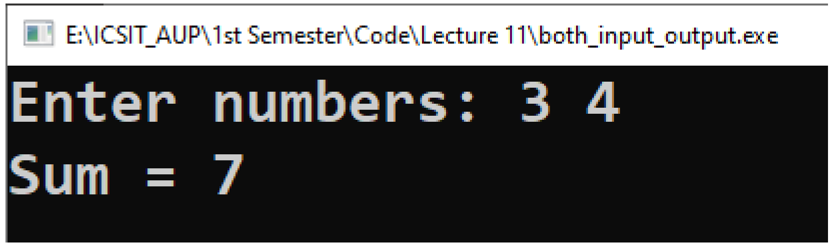
Functions with both Input and Output

Example



both_input_output.cpp

```cpp
// Functions with both input and output
// C++ program to find sum of two numbers

#include <iostream>
using namespace std;

int sum(int x, int y)
{
    int ans;
    ans = x + y;
    return ans;
}

int main()
{
    int num1, num2, add;
    cout << "Enter numbers: ";
    cin >> num1 >> num2;
    add = sum(num1, num2); // function call
    cout << "Sum = "<< add;

    return 0;
}
```



E:\ICSIT_AUP\1st Semester\Code\Lecture 11\both_input_output.exe

```
Enter numbers: 3 4
Sum = 7
```

Exercise

Task 1: Write a function which calculates & returns area of the circle. Radius should be your function parameter. Take appropriate data types.

Task 2: Write a function that takes two parameters x and y as input and returns max of two input numbers.