មហាវិទ្យាល័យវិស្វកម្ម
FACULTY OF ENGINEERING

Data Structure & Algorithm

Lecture 2
Introduction to Data Structure & Algorithm

Chhoeum Vantha, Ph.D.
Telecom & Electronic Engineering

Content

- Function in C++
- Class in C++

Function in C++

What is Function?

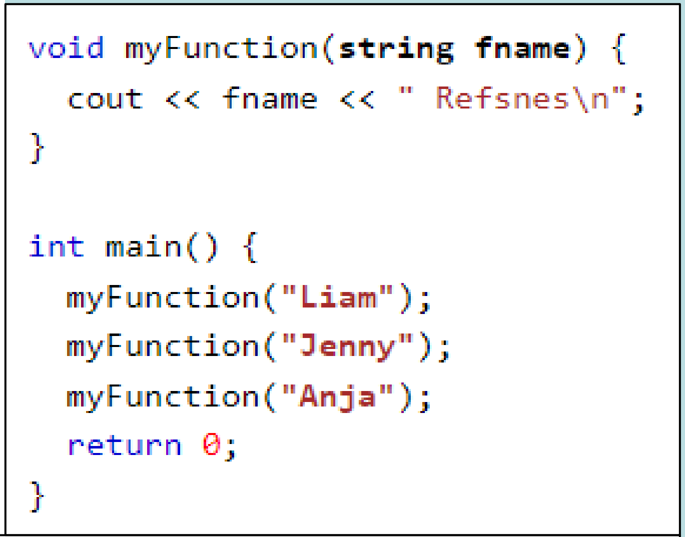
- A function is a block of code that only runs when it is called.
- You can pass data, known as parameters, into a function.
- Functions are used to perform certain actions, and they are important for reusing code:

The Keyword function in C++

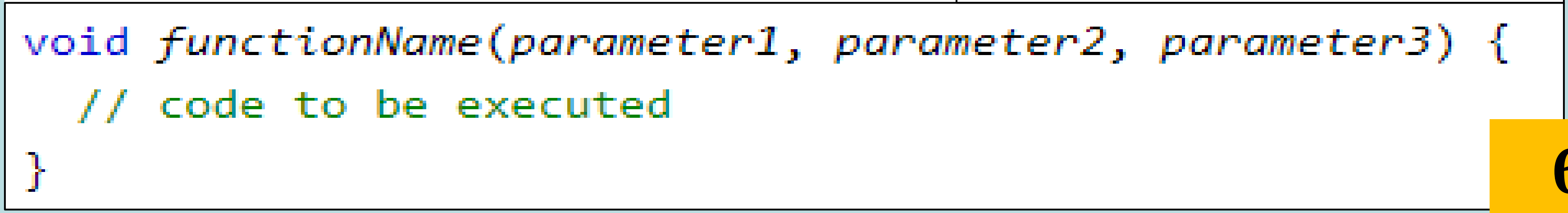
- The void keyword, used in the previous examples, indicates that the function should not return a value.
- If you want the function to return a value, you can use a data type (such as int, string, etc.) instead of void, and use the return keyword inside the function:

Function with *Parameters* in C++

- Parameters are specified after the function name, inside the parentheses.
- You can add as many parameters as you want, just separate them with a comma:



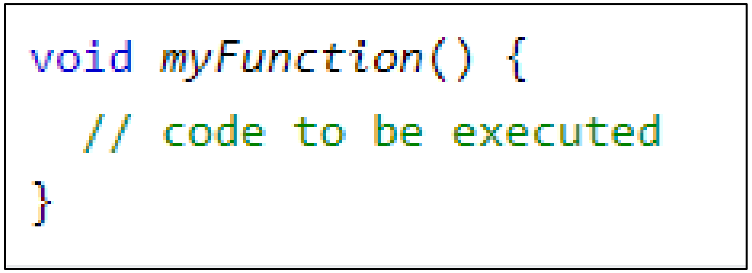
```
void myFunction(string fname) {
  cout << fname << " Refsnes\n";
}

int main() {
  myFunction("Liam");
  myFunction("Jenny");
  myFunction("Anja");
  return 0;
}
```



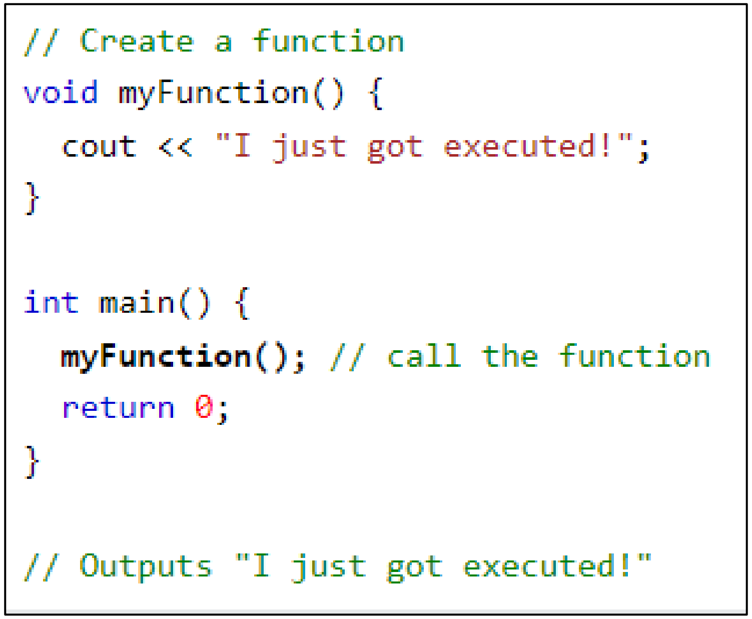
```
void functionName(parameter1, parameter2, parameter3) {
  // code to be executed
}
```

void Functions In C++

- void means that the function does not have a return value.



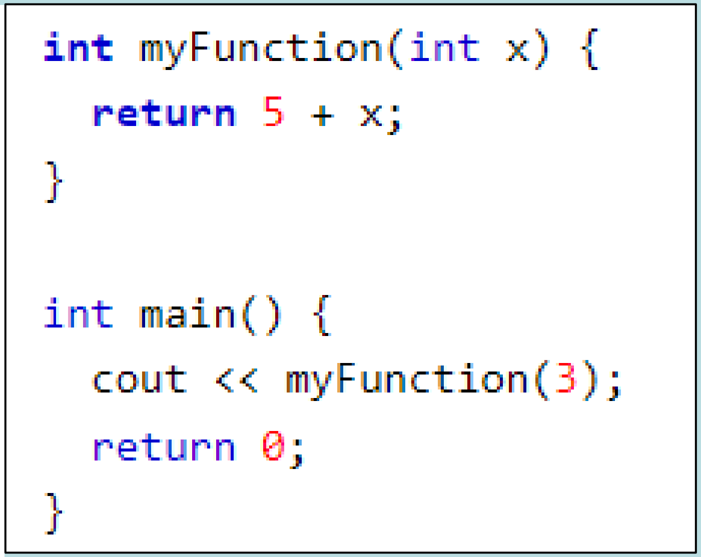
```
void myFunction() {
  // code to be executed
}
```



```
// Create a function
void myFunction() {
  cout << "I just got executed!";
}

int main() {
  myFunction(); // call the function
  return 0;
}

// Outputs "I just got executed!"
```

Function with *return value* in C++

- A function that returns a value is called a value-returning function.
- A function is value-returning if the return type is anything other than void.
- A value-returning function must return a value of that type.

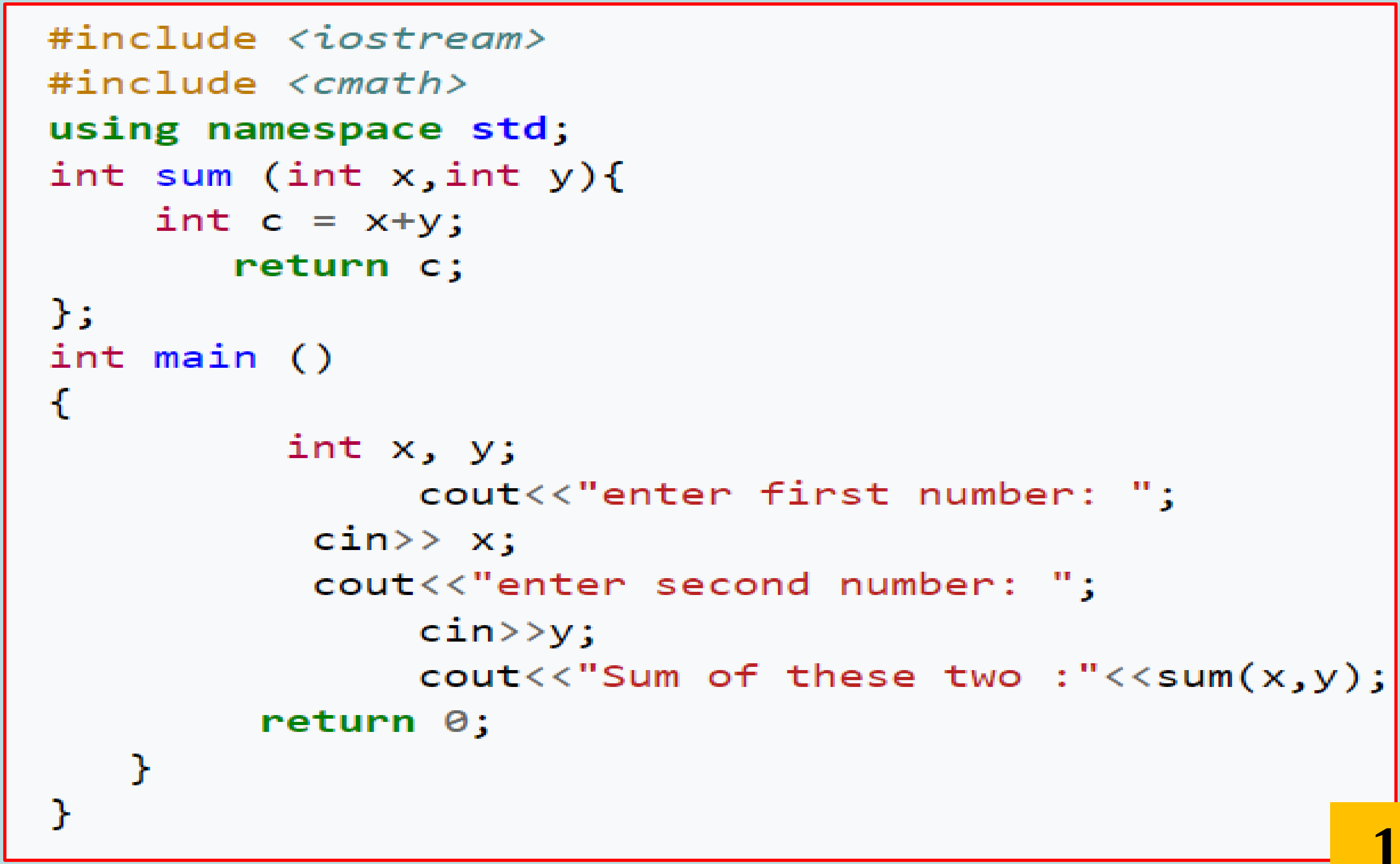


```
int myFunction(int x) {
  return 5 + x;
}

int main() {
  cout << myFunction(3);
  return 0;
}
```

Function *Calls* in C++

- A function call is an expression of the same type as the function and whose value corresponds to the return value.
- The return value is commonly passed to a suitable variable.

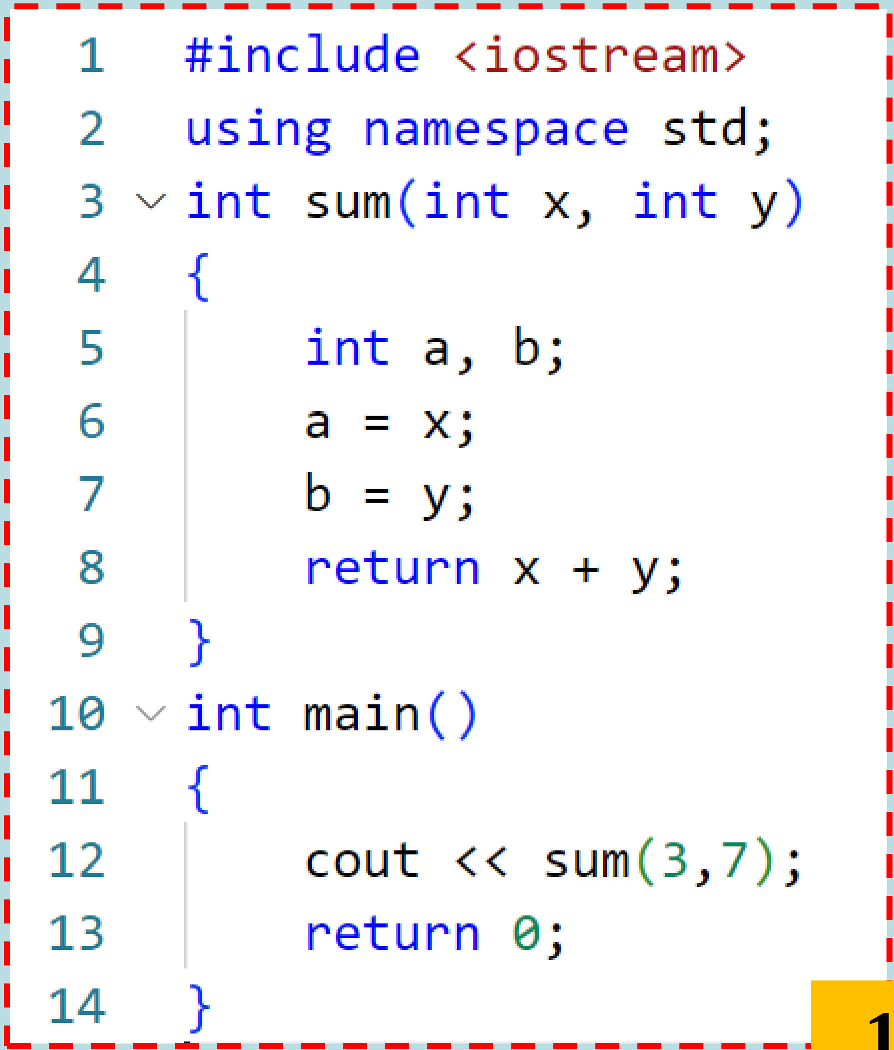
Function *Calls* in C++



```
#include <iostream>
#include <cmath>
using namespace std;
int sum (int x,int y){
    int c = x+y;
       return c;
};
int main ()
{
         int x, y;
              cout<<"enter first number: ";
          cin>> x;
          cout<<"enter second number: ";
              cin>>y;
              cout<<"Sum of these two :"<<sum(x,y);
        return 0;
   }
}
```

Position of function in C++

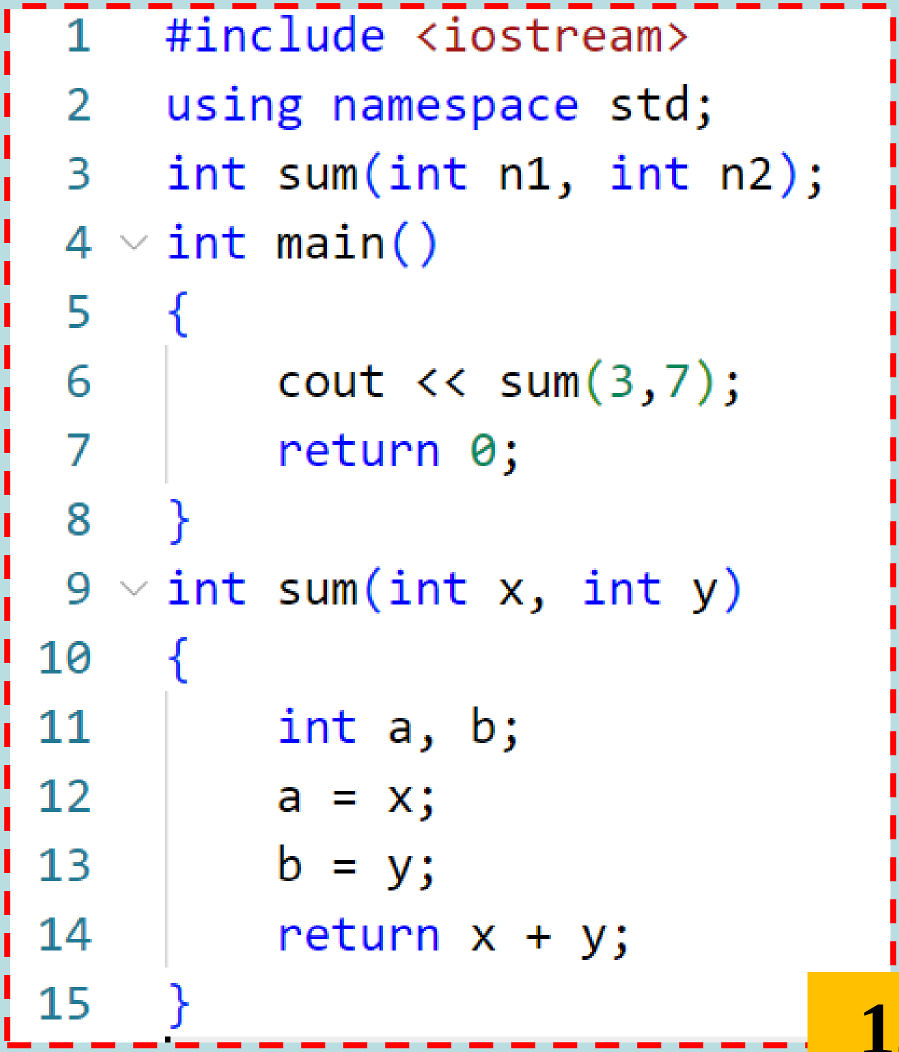
- Above *int main ()* function



```cpp
#include <iostream>
using namespace std;
int sum(int x, int y)
{
    int a, b;
    a = x;
    b = y;
    return x + y;
}
int main()
{
    cout << sum(3,7);
    return 0;
}
```

Position of function in C++

- Below *int main ()* function



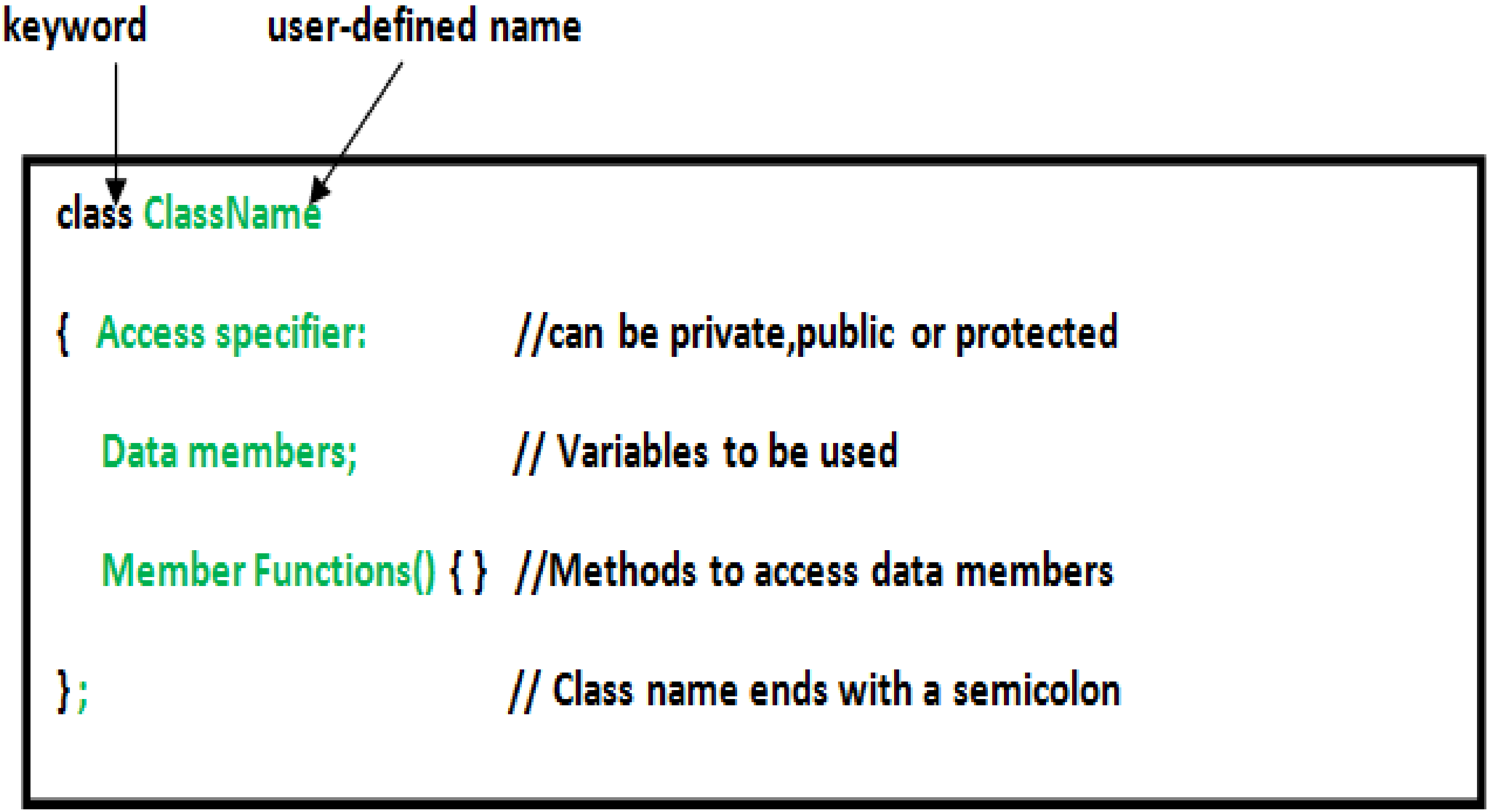
```cpp
#include <iostream>
using namespace std;
int sum(int n1, int n2);
int main()
{
    cout << sum(3,7);
    return 0;
}
int sum(int x, int y)
{
    int a, b;
    a = x;
    b = y;
    return x + y;
}
```

Class in C++

What is *class* in C++?

- A class is a user-defined data type that we can use in our program
- It works as an object constructor
- a "blueprint" for creating objects

Structure of Class in C++

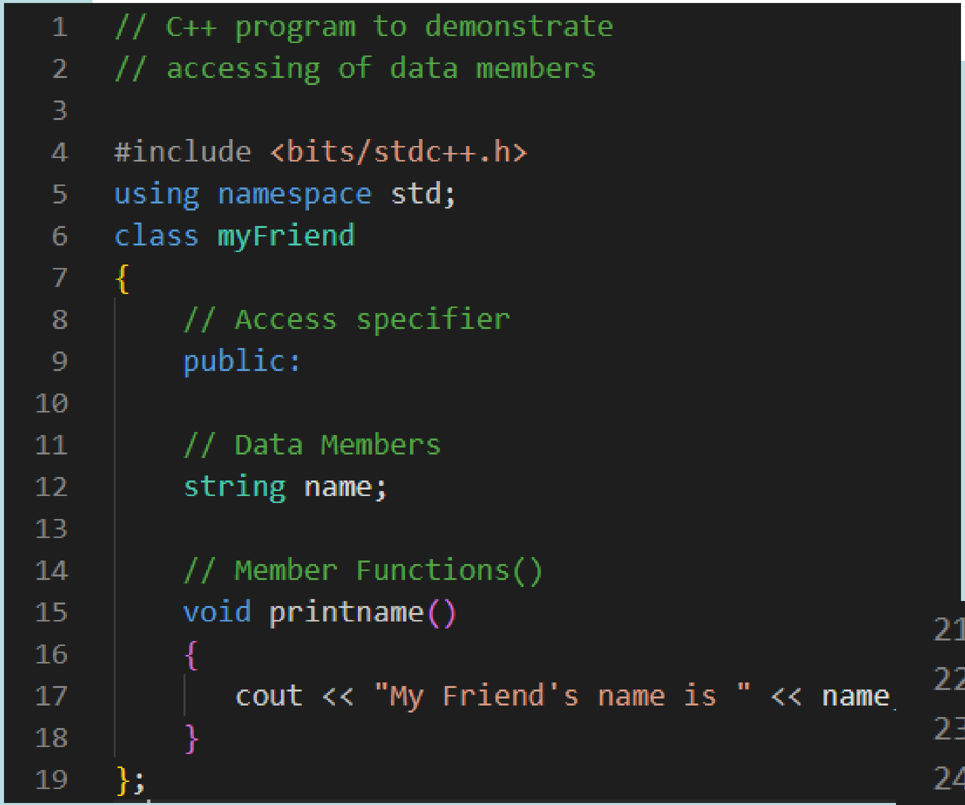


keyword

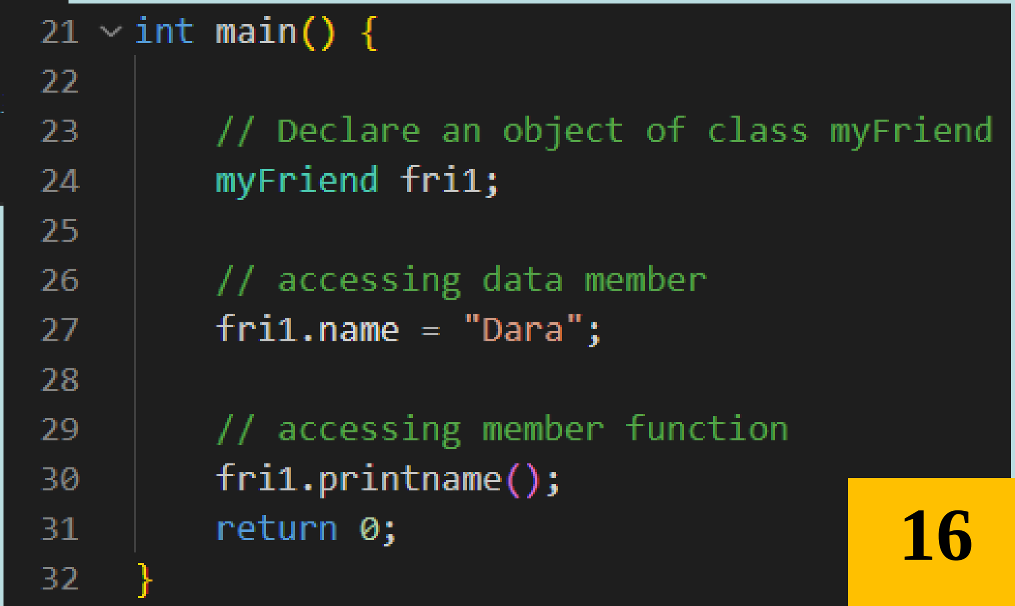
user-defined name

```
class ClassName

{  Access specifier:          //can be private,public or protected

   Data members;              // Variables to be used

   Member Functions() { }     //Methods to access data members

};                            // Class name ends with a semicolon
```

Structure of Class in C++



```
// C++ program to demonstrate
// accessing of data members

#include <bits/stdc++.h>
using namespace std;
class myFriend
{
    // Access specifier
    public:

    // Data Members
    string name;

    // Member Functions()
    void printname()
    {
        cout << "My Friend's name is " << name
    }
};
```



```
int main() {

    // Declare an object of class myFriend
    myFriend fri1;

    // accessing data member
    fri1.name = "Dara";

    // accessing member function
    fri1.printname();
    return 0;
}
```

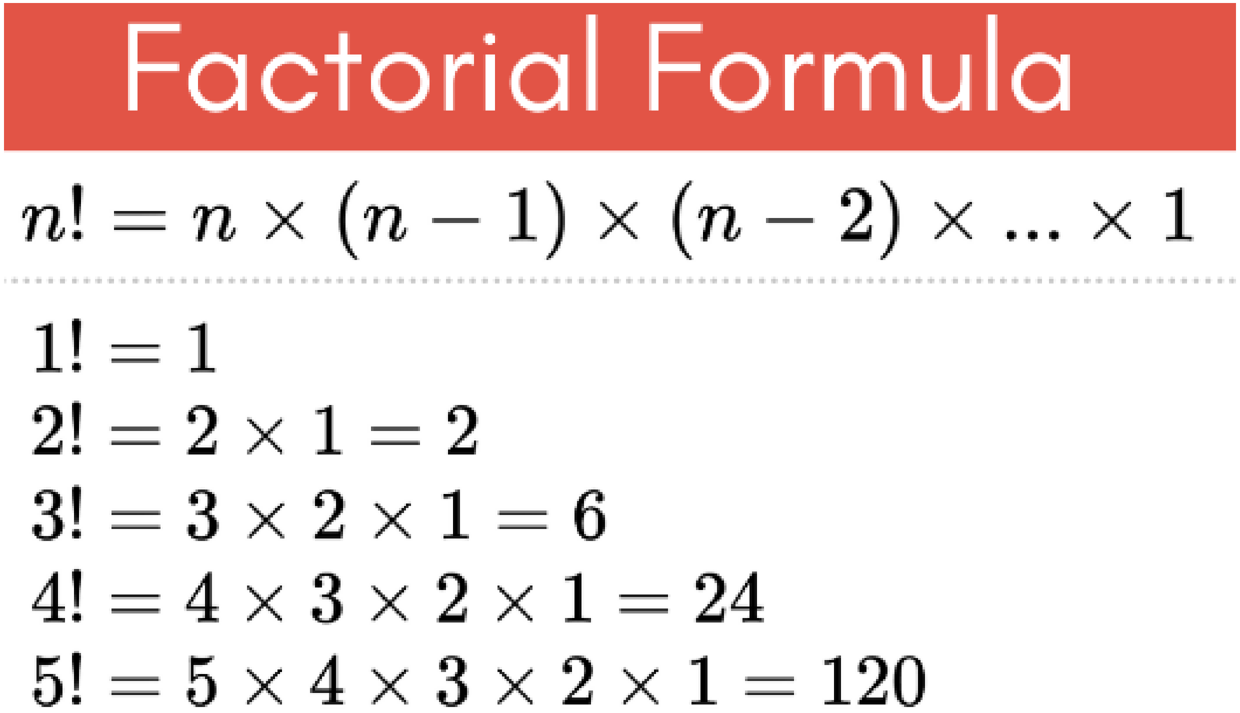
W2 – Lab 2

Ex 1

- Write a program that asks for two numbers, compares them, and shows the maximum.
- Declare a function called *max_two* that compares the numbers and returns the maximum.

Ex 2

Write a program to print the factorial of a number by defining a function named 'Factorial’.



Factorial Formula

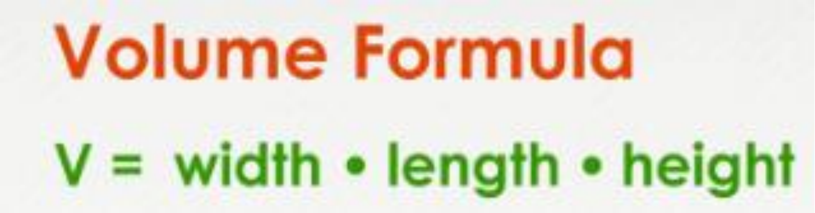
$$n! = n \times (n-1) \times (n-2) \times \ldots \times 1$$

$$1! = 1$$
$$2! = 2 \times 1 = 2$$
$$3! = 3 \times 2 \times 1 = 6$$
$$4! = 4 \times 3 \times 2 \times 1 = 24$$
$$5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$$

Ex 3

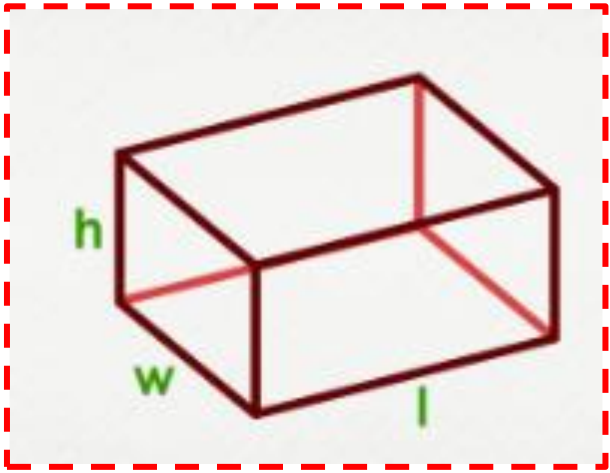
Write a class name as *classBox* having three variables and one member function which will return the volume of the box.

Example: display the volume of the box with dimensions of (3,4,5)



Volume Formula

V = width • length • height



Ex 4

Write a class name as *myFriend* that contains my friend information such as:

- Name
- Date of birth
- Place of birth
- Department

Then print the all information related to your friends at least 5 friends

Ex 5

Design and Write a class name as *infomRUPP* that contains a few pieces of information such as:

- Number of Faculty in RUPP
- Name of Faculty in RUPP
- Name of the department of FE
- Display all information-related classes of *infomRUPP*

Thanks!