

C++ PROGRAMMING

2ND LEC, FUNCTION IN C++

BY: MUQADDAR NIAZI

*Structure Programming with c++
with: Muqaddar Niazi*

FUNCTION

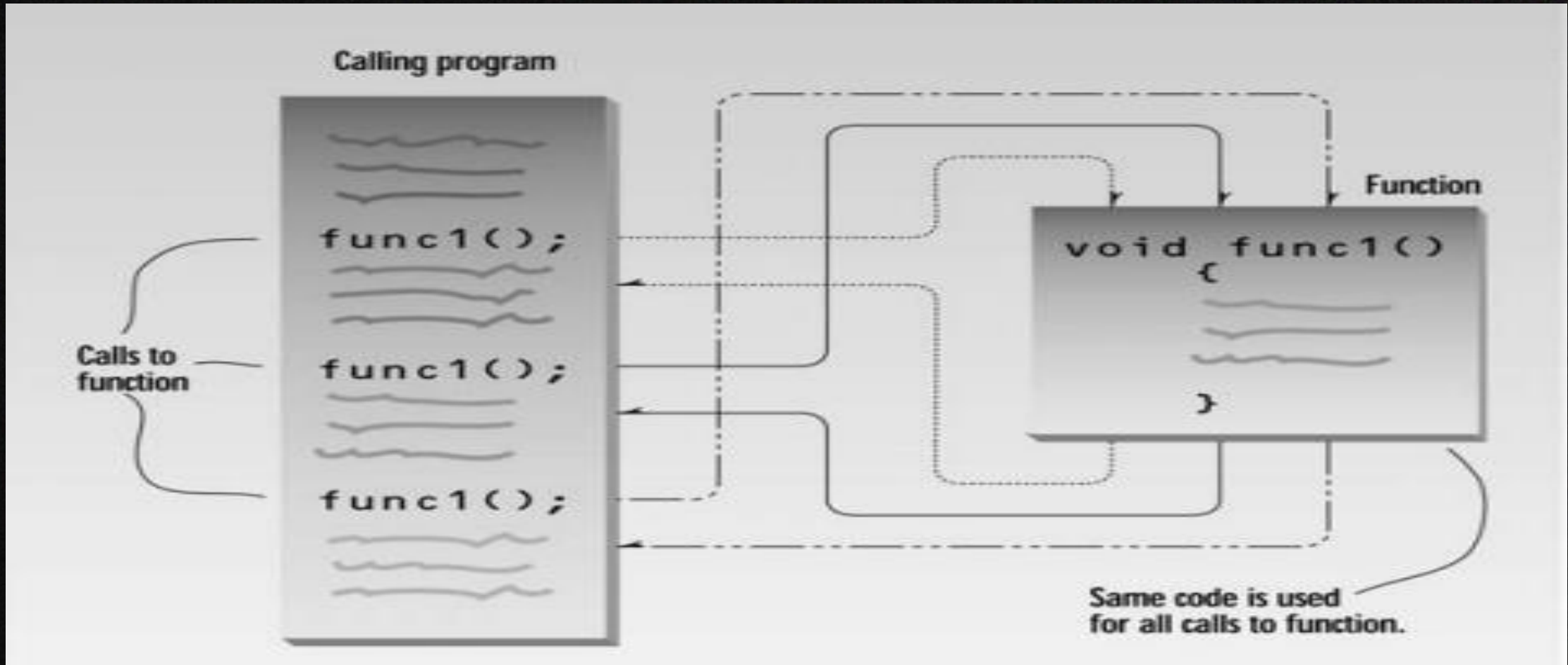
Definition: a Function is a block of code which is executes when it's called from a point of program.

A function groups a number of program statements into a unit and gives it a name. This unit can then be invoked from other parts of the program.

Like: in this picture.

*Structure Programming with c++
with: Muqaddar Niazi*

CONTINUE...



Structure Programming with c++
with: Muqaddar Niazi

CONTINUE...

- **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.**

*Structure Programming with c++
with: Muqaddar Niazi*

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 `ceil(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.

*Structure Programming with c++
with: Muqaddar Niazi*

USER DEFINED FUNCTION

A user defined function consists of the following

Function declaration

Function definition

*Structure Programming with c++
with: Muqaddar Niazi*

FUNCTION DECLARATION

Function declaration is a model of a function it is also called function prototype . It provides information to compiler about the structure of the function to be used in program .

It ends with a ; and are written at the beginning of the source file just before the main function. Function declaration consists of the following parts

Function name

Function return type

Number and type of parameters

Syntax

Return type func-name (parameters);

E.g. : Int sum(int a,int b);

*Structure Programming with c++
with: Muqaddar Niazi*

CONTINUE...

- **Return type :** it indicates the type of value that will be returned by function .if the function return no value the keyword void is used.
- **Function name:** it indicates the name of function that is to be created.
- **Parameters:** parameters are the values that are provided to a function when the function is called. parameters are given in a parenthesis . If there are many parameters these are separated by commas. if there is no parameters empty parenthesis are used are void key word is used

Structure Programming with c++

with: Muqaddar Niazi

PASSING VALUES TO A FUNCTION

Only data type of the parameters are written in prototype as

int add(int ,int)

Both data type and names of the parameters are written in prototype as

int add (int a, int b)

Void show(void)

Int cube(int)

Int add (int a,int b,int c)

Void line(char)

*Structure Programming with c++
with: Muqaddar Niazi*

FUNCTION DEFINITION

A set of statements that explains what a function does is called function definition

The function definition can be written at the following places

-before main function

-after main function

Function declaration is not required if the function is written before main function. function declaration is compulsory if the function definition is after main function.

*Structure Programming with c++
with: Muqaddar Niazi*

CONTINUE...

The function definition consists of two types

Function header

Function body

Return type function name(parameters)

{

Statements;

}

*Structure Programming with c++
with: Muqaddar Niazi*

CONTINUE...

E.g.:

```
void show()  
{  
Printf("example of function");  
}
```

*Structure Programming with c++
with: Muqaddar Niazi*

FUNCTION CALL

The statements that activates a function is known as function call .a function can be called with its name.

The following steps take place when a function is called

- **The control moves to the function that is called**
- **All statements in the function body are executed**
- **The control return back to the calling function**

*Structure Programming with c++
with: Muqaddar Niazi*

EXAMPLE OF FUNCTION

```
#include<iostream>
using namespace std;
void show();
main(){
    show();
}
void show(){
    cout<<"I'm a Function";
}
```

*Structure Programming with c++
with: Muqaddar Niazi*

CATEGORY OF FUNCTION

We are going to categorize Function into four 4 types, just for learning it easily:

- 1. Function with no return type and with no parameters**
- 2. Function with no return type and with parameters**
- 3. Function with return type and no parameters**
- 4. Function with return type and parameters**

*Structure Programming with c++
with: Muqaddar Niazi*

FUNCTION WITH NO RETURN TYPE AND WITH NO PARAMETERS

A type of Function which don't return any value and also we don't pass any value into this.

E.g.:

```
void show(void);
```

*Structure Programming with c++
with: Muqaddar Niazi*

CONTINUE...

```
#include<iostream>
using namespace std;
void show(void);
main(){
    show();
}
void show(void){
    cout<<"hello 🖐 ";
}
```

*Structure Programming with c++
with: Muqaddar Niazi*

PRACTICE

**WRITE A FUNCTION WHICH FIND SUM OF TWO
NUMBER, NOTE: FUNCTION SHOULD BE FIRST TYPE**

*Structure Programming with c++
with: Muqaddar Niazi*

FUNCTION WITH NO RETURN TYPE AND WITH PARAMETERS

A type of Functions that haven't return any value but have some parameters.

Parameters in function call are called actual parameter

Parameters in function declaration are called formal parameters.

When a function call is executed the value of actual parameters are copied to formal parameters.

E.g.:

void sum(int a , int b);

void area(int r);

In Function call we should pass that type of values.

Int r=2;

area(r);

*Structure Programming with c++
with: Muqaddar Niazi*

E.G. OF 2ND TYPE

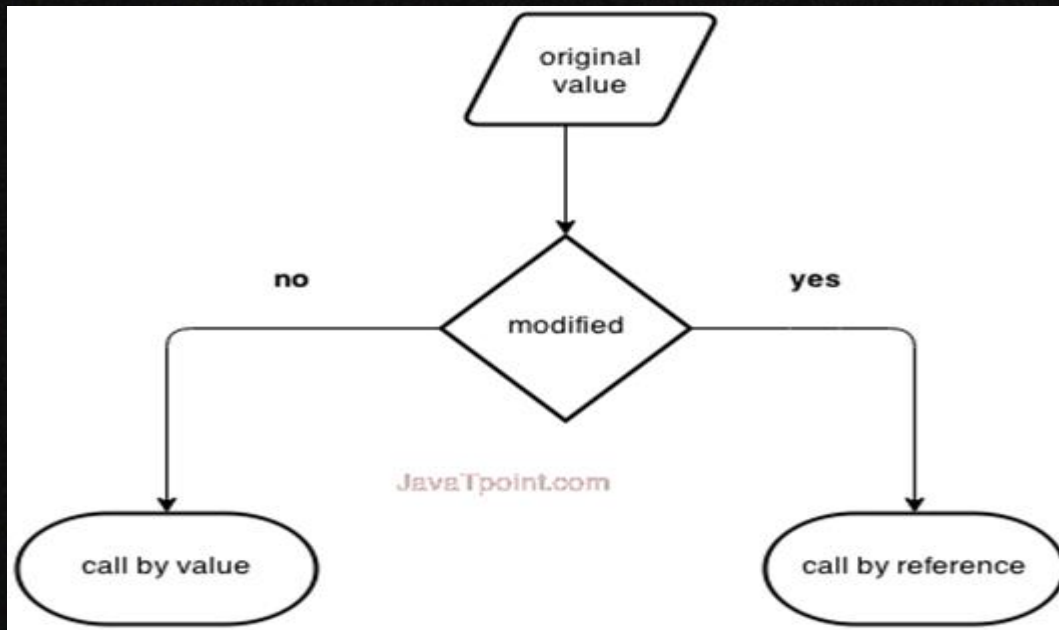
```
#include<iostream>
using namespace std;
void sum(int a,int b){
cout<<a<<"+"<<b<<"="<<a+b<<endl;
}
main(){
    int num1,num2;
    sum(num1,num2);
}
```

*Structure Programming with c++
with: Muqaddar Niazi*

PASSING VALUES TO FUNCTION

We can pass values to Function through two ✌️ ways:

1. call by value
2. call by address/ reference



Structure Programming with c++
with: Muqaddar Niazi

CALL BY VALUE

A parameter passing mechanism in which the value of actual parameters is copied to formal parameters of called function is known as pass by value. If the function makes any changes in formal parameter it does not affect the value of actual parameters .it is the default mechanism for passing parameters to functions.

```
void show(int a);
```

*Structure Programming with c++
with: Muqaddar Niazi*

CALL BY REFERENCE

In call by reference, original value is modified because we pass reference (address).

Here, address of the value is passed in the function, so actual and formal arguments share the same address space. Hence, value changed inside the function, is reflected inside as well as outside the function.

Note: To understand the call by reference, you must have the basic knowledge of pointers.

*Structure Programming with c++
with: Muqaddar Niazi*

PRACTICE

**WRITE PROGRAM FOR SWAPPING TWO NUMBERS
AND SEE DIFFERENCE BETWEEN CALL BY VALUE
AND CALL BY REFERENCE.**

*Structure Programming with c++
with: Muqaddar Niazi*

FUNCTION WITH RETURN TYPE AND NO PARAMETERS

A type of Function that return a value after calculation and don't have parameters.

Syntax:

return-type funName();

E.g.:

int sum();

*Structure Programming with c++
with: Muqaddar Niazi*

EXAMPLE

- **#include<iostream>**
- **int sum(void);**
- **main(){**
 cout<<sum();
 }
 int sum(){
 int num1 =3 ;
 int num2 = 4;
 return num1+num2;
 }

*Structure Programming with c++
with: Muqaddar Niazi*

PRACTICE

**WRITE A PROGRAM FOR CHECKING EVEN OR ODD
CHECKING WITH KEYBOARD VALUE THROUGH
FUNCTION WHICH RETURN TRUE OR FALSE.**

*Structure Programming with c++
with: Muqaddar Niazi*

FUNCTION WITH RETURN TYPE AND PARAMETERS

A type of Function that return a value and we can pass some values to it.

Syntax:

retrun-type fun_Name(parameters);

E.g.:

int sum(int a, int b);

*Structure Programming with c++
with: Muqaddar Niazi*

EXAMPLE

```
#include<iostream>
using namespace std;
int sum(int a , int b){
    return a+b;
}
main(){
    cout<<sum(3,2)<<endl;
}
```

*Structure Programming with c++
with: Muqaddar Niazi*

PRACTICE

**WRITE A PROGRAM FOR TO PASS TWO ↴ NUMBERS
TO FUNCTION , THAT FUNCTION RETURN BIG ONE.**

*Structure Programming with c++
with: Muqaddar Niazi*

Good bye 🖐️.

References:

www.javatpoint.com

Let us c by: yashwant

Next Topic:

Array in C++.

*Structure Programming with c++
with: Muqaddar Niazi*