Programming Fundamentals

Course Code: CS-111

Course Instructor: Habiba Arshad

Goals for today

- To review the basics of C++
  - Function

Learning Objectives

- Functions
- Types of Functions
- User Defined Functions
  - Function Declaration
  - Function Definition
  - Function Call
  - Scope of Function
- Passing parameters to functions
  - Pass by value
  - Pass by reference
  - Returning value from function

Learning Objectives

- Passing Array as Parameter/Argument
- Declaring function with array as parameter
- Function definition with array as parameter
- Calling function with array as parameter
- Function Overloading

CLO Covered

- CLO1: Describe fundamental problem-solving techniques and logic constructs. GA 1
- CLO2: Apply basic programming concepts. GA2
- CLO3: Analyze and solve the real-world problems by using programming constructs. GA3

Functions

- A function is a named block of code that performs some action.
- The statements written in a function are executed when it is called by its name.
- Each function has a unique name.
- Functions are the building blocks of C++ programs. They encapsulate pieces of code to perform specified operations.
- The functions perform similar kinds of task again and again without writing the same code again.
- They are used to perform the tasks that are repeated many times.

Functions

- The control moves in the function when a function is called.
- All statements of the function are executed and then the control again moves back to the point where the function was called along with possible return value.
- The functions provide a **structured programming** approach.
- It is a **modular way of writing programs.** As, the whole program logic is divided into number of smaller modules or functions.
- The main function calls these functions when they are needed to execute.

Types of Functions

- There are two types of functions
- **Built-in Functions(Standard Functions)**
  - Functions that have already been defined as a part of language.
  - These functions are stored in header files.
- **User-defined Functions**
- A type of function created by user.
- It has a unique name.
- A program may contain many user-defined functions.

User Defined Functions

- It has three parts
  - Function declaration(prototype)
  - Function definition
  - Function calling

Function Declaration

- It is a model of a function also called as function prototype.
- It tells the compiler about the structure of the function to be used in the program.
- Function prototypes are usually placed at the beginning, just before the main ( ) function.
- **Function declaration/ prototype has three parameters:**
- Function return type
- Function name
- Number and types of parameters

Function Declaration

- Function declaration is terminated by a semicolon.
- Similar to declaration of a variable and the **rules for naming the functions is also same as for naming variables**

- **Syntax**

return-Type **function-Name** (parameters if any);

Function Declaration

- Examples:
- int sum(int , int);
- void display( );
- void myFunction(void);
- void line(char);
- float sum();
- void print(int, float , char);

Function Definition

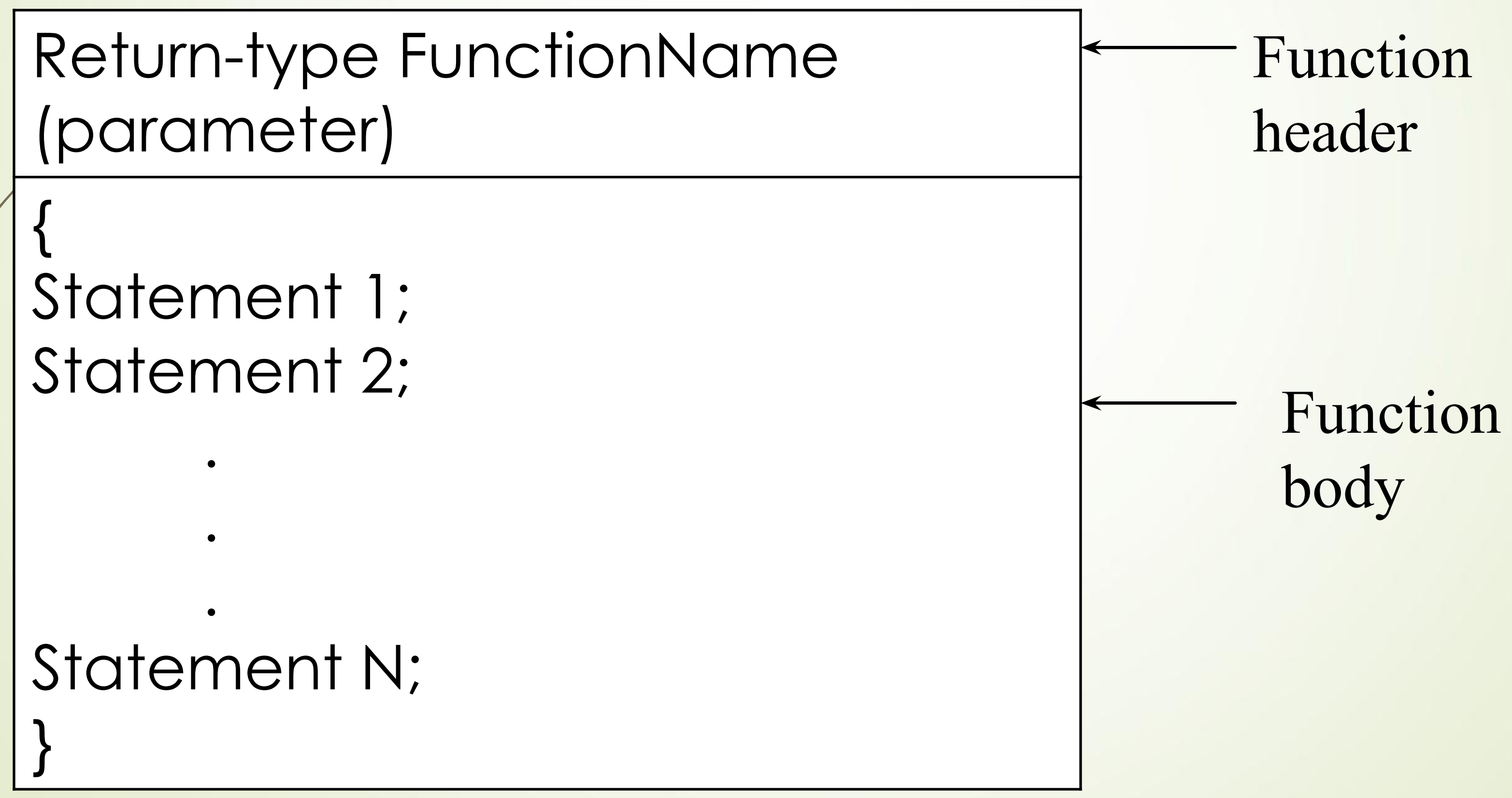
- A set of statements that explains what a function does is called function definition.
- The function definition can be written at the following places
- Before main()
- After main function()
- In a separate file
- Function declaration is
- Not required if the function definition is written before main() function.
- Is compulsory if the function definition is written after main() function.
- If function definition is written in a separate file then it can be used by including that file in the program using **#include** preprocessor directive.

Function Definition

- Function definition consists of two parts
- **Function Header**
- It is the first line of function definition also called as function declarator.
- It is not terminated with semicolon.
- The number of parameters and sequence parameters in function header and function prototype/declaration must be same.
- **Function Body**
- The set of statements which are executed inside the function and perform a specific task.
- It appears after function declarator and the statements are written in curly braces { }

Function Definition

Syntax

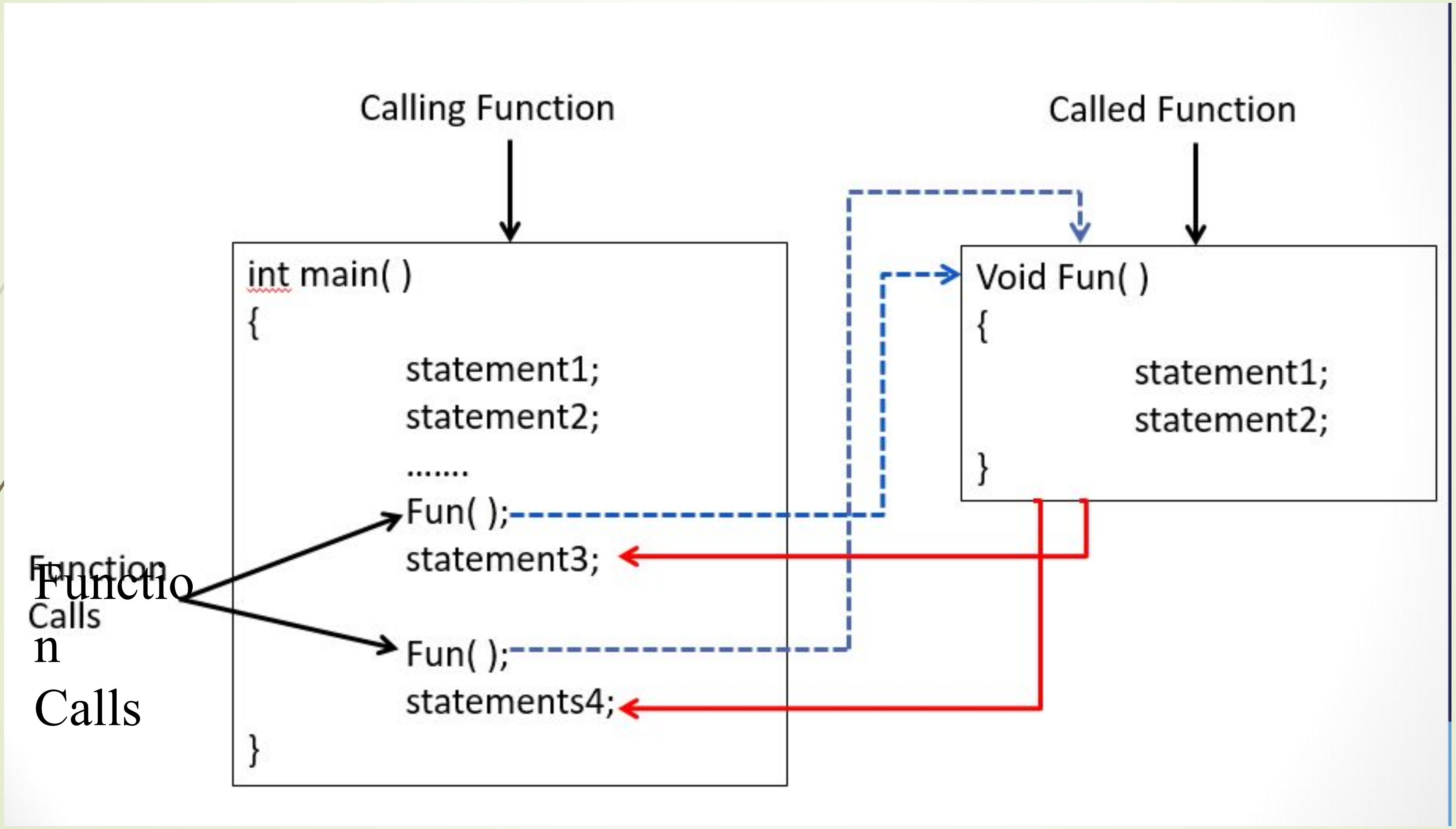


```
Return-type FunctionName
(parameter)
{
Statement 1;
Statement 2;
          .
          .
          .
Statement N;
}
```

Function Call

- The statement that activates a function is known as function call.
- A function is called with its name
- The parameters(if any) are given in the parentheses after the name of function otherwise left blank.
- **When a function is called**,
- the control shifts to the function definition.
- The statements of the body of the function are executed.
- **After execution:**
- The control returns to the calling function
- and the next statements that comes immediately after the function call is executed.

Function Call



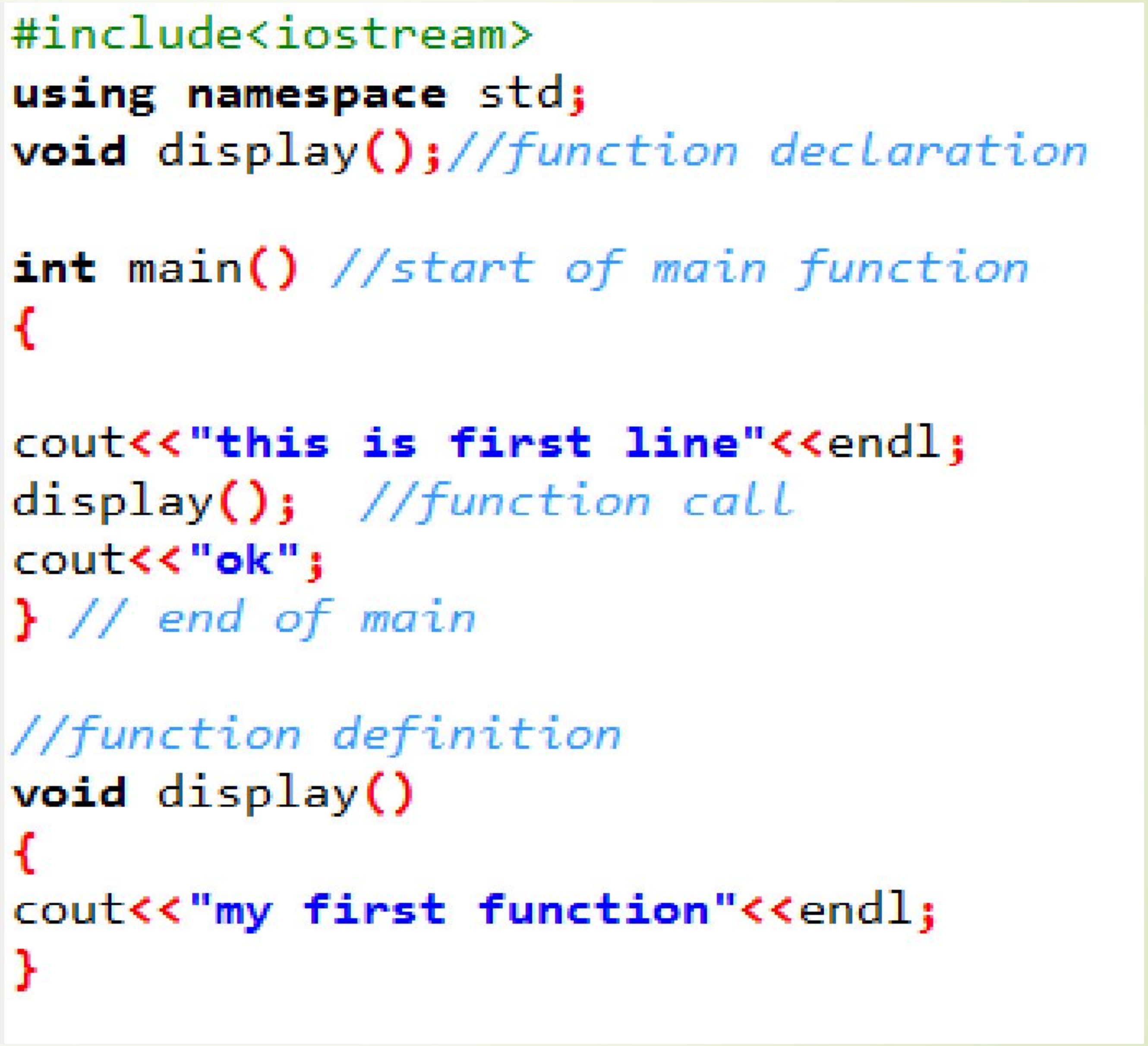
```
int main( )
{
            statement1;
            statement2;
            .......
            Fun( );
            statement3;

            Fun( );
            statements4;
}
```

```
Void Fun( )
{
            statement1;
            statement2;
}
```

Functio
n
Calls

Example



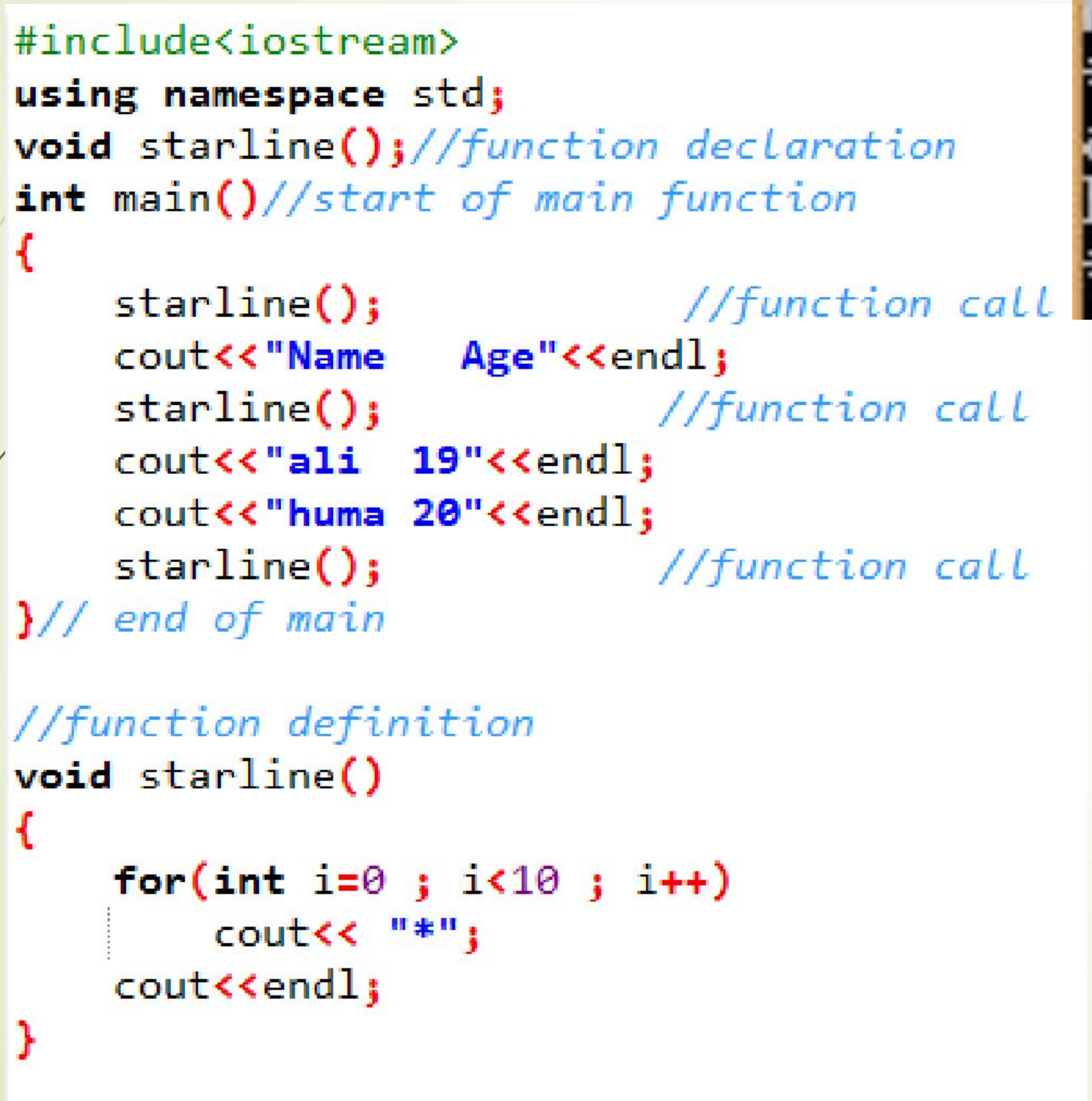
```
#include<iostream>
using namespace std;
void display();//function declaration

int main() //start of main function
{

cout<<"this is first line"<<endl;
display();  //function call
cout<<"ok";
} // end of main

//function definition
void display()
{
cout<<"my first function"<<endl;
}
```

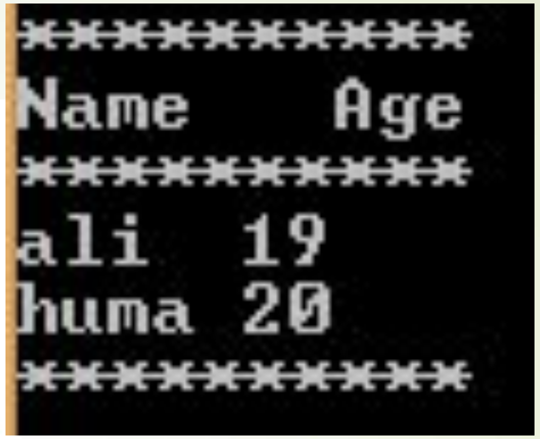
Example



```cpp
#include<iostream>
using namespace std;
void starline();//function declaration
int main()//start of main function
{
    starline();              //function call
    cout<<"Name   Age"<<endl;
    starline();             //function call
    cout<<"ali  19"<<endl;
    cout<<"huma 20"<<endl;
    starline();             //function call
}// end of main

//function definition
void starline()
{
    for(int i=0 ; i<10 ; i++)
        cout<< "*";
    cout<<endl;
}
```

Output



```
**********
Name    Age
**********
ali  19
huma 20
**********
```