Lecture (9)
" Programming Essentials in C++ "

Presented by : Dr. Nehal El Azaly,
Dr. Mohamed Abd EL Fattah

C++ Function

- **Importance of Function**
  - A program may need to repeat the same piece of code at various places.
  - The program may become very large if functions are not used.
  - The real reason for using function is to divide program into different parts.

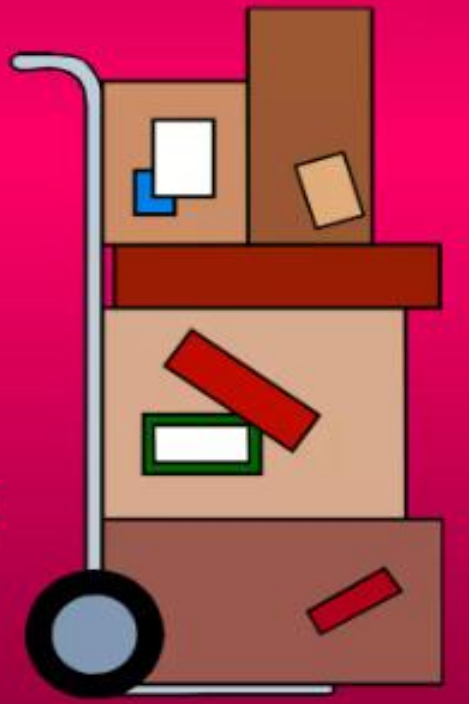
C++ Function

- C++ allows the use of both **internal** (user-defined) and external (Built IN) functions.
- External Functions are usually grouped specialized libraries (e.g., iostream, stdlib, math, etc.)

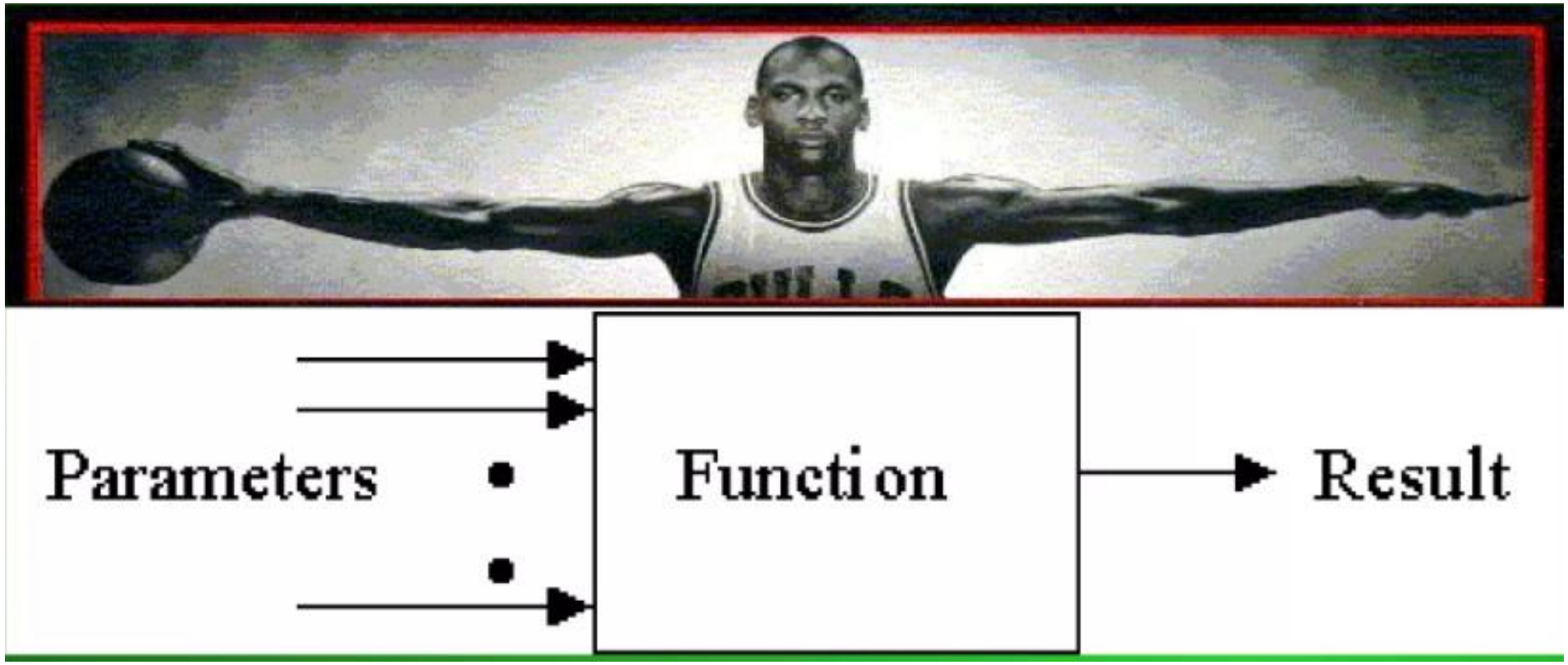
User –Defined Functions

C++ programs usually have the following form:

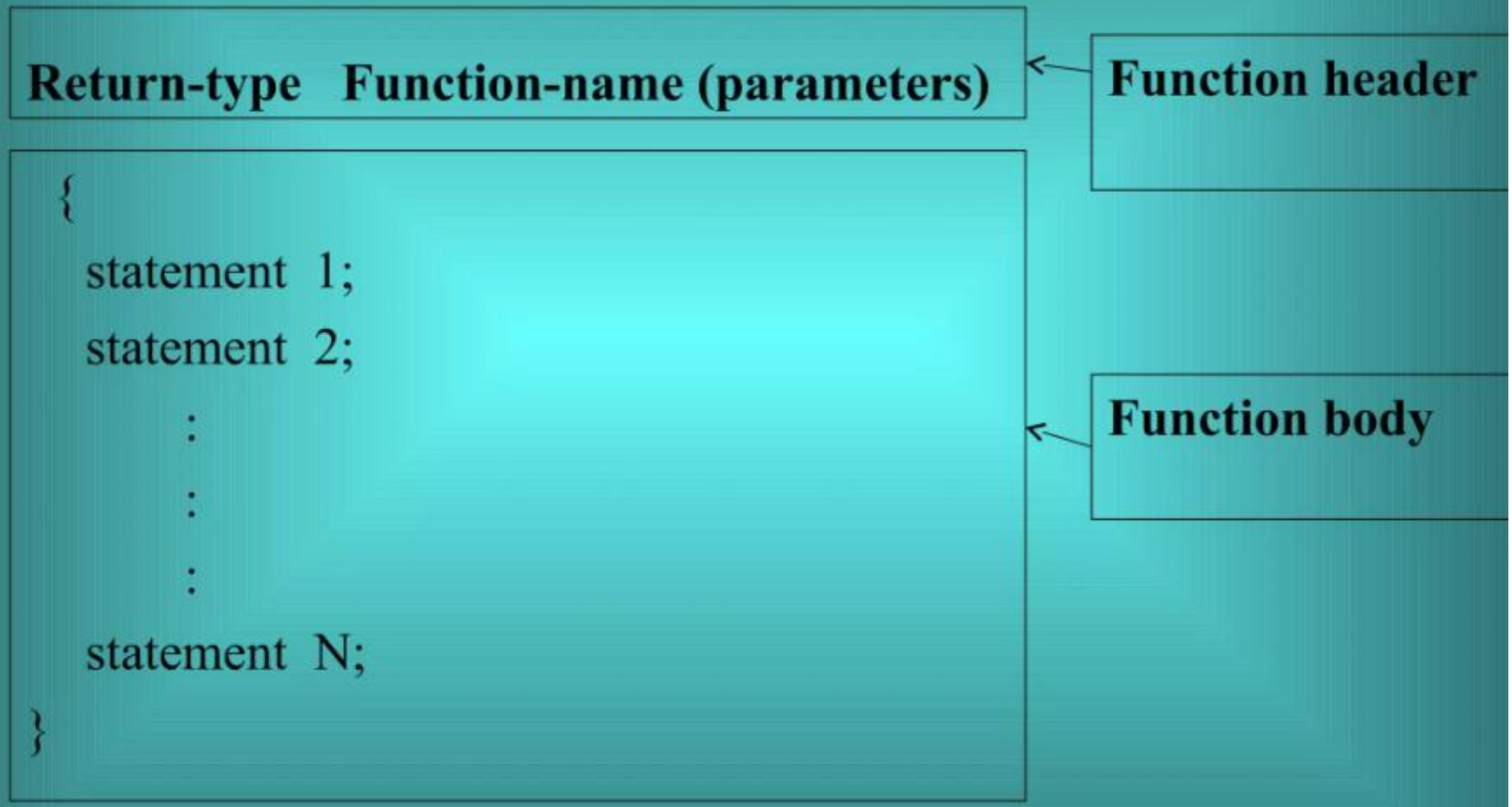
```
// include statement
// function prototype
// main() function
// function definitions
```



Function Input and Output



Syntax of Function Definition



Syntax of Function Declaration

Return_type Function Name (Parameters);

Indicates the type of the value that will be return by function

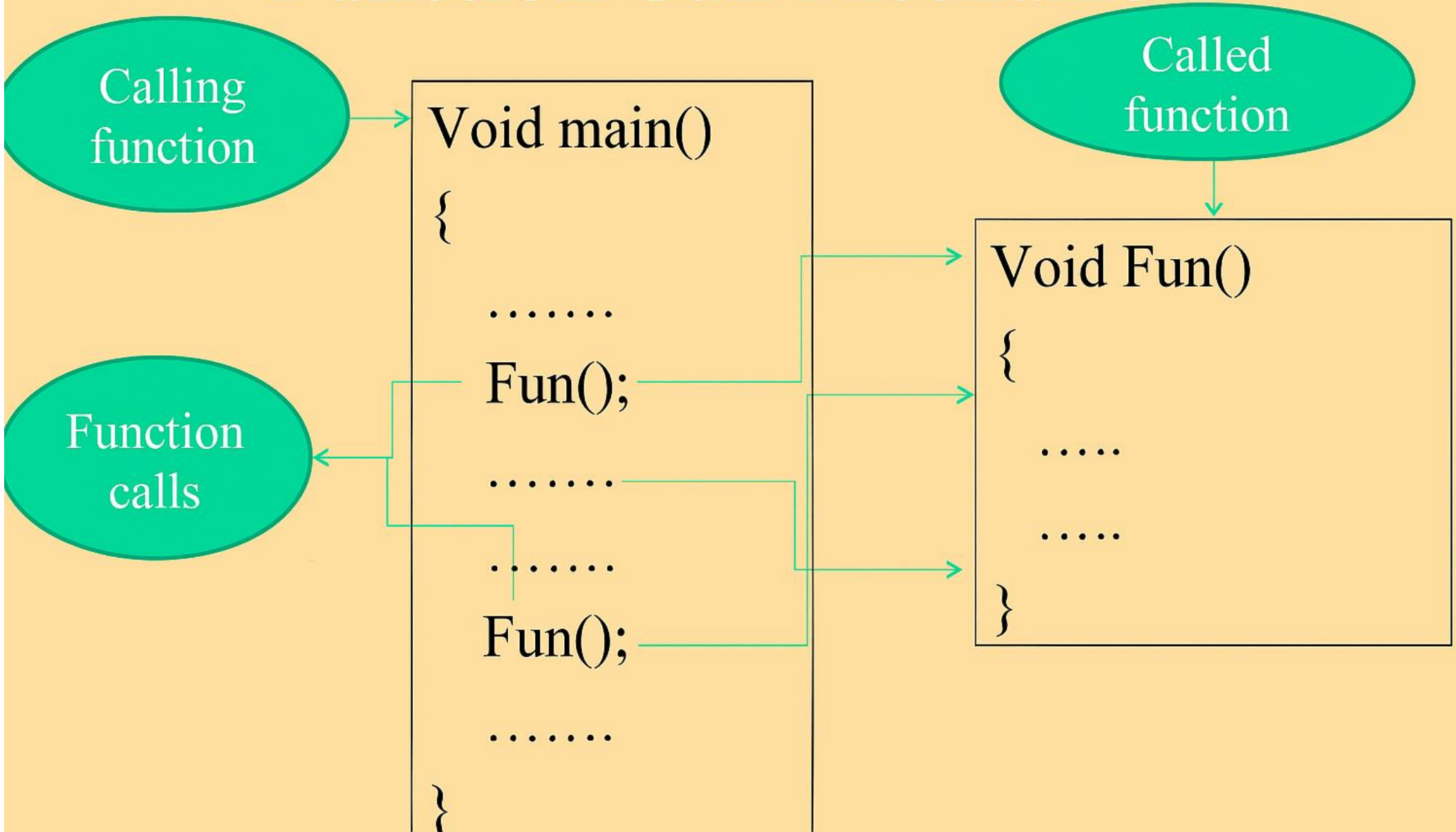
Indicates the name of function

Parameters are the values that are provided to a function when the function is called

Example of Function Declaration and Definitio

```
1   #include<iostream>
2   using namespace std;
3
4   int sum (int x , int y);          // function declaration
5   int sum(int x , int y)            // funciotn definition
6   {
7       int result;
8     result = x  + y;
9     return (result);
10  }
11  int main()
12  {
13      int x , y , output ;
14    x = 20;
15    y = 100;
16    output = sum(x,y);          /* calling a function and storing the
17                                value from funciton to variable output*/
18   cout<<output;
19
20  return 0;
21  }
```

Function Call Mechanism



Example of Function Call

```
int main()
{
    double num1, num2, maxNum;
    cout<<"Please enter a number :";
    cin>> num1;
    cout<<"Great!\nPlease enter a second number :";
    cin>> num2;
    FindMax(num1, num2, maxNum); //calling

    system ("PAUSE");
    return 0;
}
```

Example of Function Call

```
void myFunction() {
  cout << "I just got executed!\n";
}

int main() {
  myFunction();
  myFunction();
  myFunction();
  return 0;
}

// I just got executed!
// I just got executed!
// I just got executed!
```

Function Parameters and Arguments

- Information can be passed to functions as a parameter. Parameters act as variables inside the function.

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

Example

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

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

// Liam Refsnes
// Jenny Refsnes
// Anja Refsnes
```

Default Parameter Value

- You can also use a default parameter value, by using the equals sign (=).

```
void myFunction(string country = "Norway") {
  cout << country << "\n";
}

int main() {
  myFunction("Sweden");
  myFunction("India");
  myFunction();
  myFunction("USA");
  return 0;
}

// Sweden
// India
// Norway
// USA
```

Multiple Parameters

```
void myFunction(string fname, int age) {
  cout << fname << " Refsnes. " << age << " years old. \n";
}

int main() {
  myFunction("Liam", 3);
  myFunction("Jenny", 14);
  myFunction("Anja", 30);
  return 0;
}

// Liam Refsnes. 3 years old.
// Jenny Refsnes. 14 years old.
// Anja Refsnes. 30 years old.
```


Thank
you