جامعة المستقبل
AL MUSTAQBAL UNIVERSITY

قسم الامن السيبراني
Department of Cyber Security

Subject:

Structured Programming

Class:

1st stage

Lecturer:

Dr. Abdulkadhem A. Abdulkadhem

Lecture: (2)

Introduction to Functions

1. Introduction to Functions

A function is a set of statements designed to perform a specific task. In software development, constructing large programs from smaller, manageable pieces (modules) is considered the best approach. In C++, these modules are referred to as **functions**.

Functions play a crucial role in improving the readability, maintainability, and debugging of complex programs. They can be easily integrated into the main program. In C++, even the *main()* function itself is a function that calls other functions to execute various tasks.

2. Advantages of Using Functions:

- Simplifies writing small and correct code modules.
- Enhances readability, debugging, and code modification.
- Facilitates easier maintenance and updates.
- Small functions are often self-documenting and highly readable.
- Can be invoked (استدعاء) multiple times in different places with varying parameters.

3. Defining a Function

A function definition consists of a **name**, a pair of parentheses containing zero or more **parameters**, and a **body** enclosed in curly braces {}. Each parameter must have a corresponding declaration before the function body. If a parameter is not explicitly declared, it is assumed to be of type *int* by default.

4. General Syntax of a Function:

General Form of Function:

```
return-type function-name ( parameters-list )
{
    (body of function)
    statement1 ;
    statement2 ;
        :
    statement-n ;
    (return something)
}
```

The **return type** specifies the type of value the function will return (e.g., `int`, `float`, `char`, etc.). If a function does not return a value, it should be declared as `void`.

Examples:

```
void function_name();                    // Function with no
parameters
int function_name(int a, int b);        // Function with
parameters
```

Any variable declared inside a function body is considered **local** to that function. Variables that are not declared inside the function or as function parameters are considered **global** and must be defined externally.

5. Types of Functions

A. Based on Definition Type

1. Built-in Functions

C++ provides various built-in functions available in libraries such as `cmath` and `iostream`. Example:

```
#include <cmath>
#include <iostream>
using namespace std;

int main() {
    cout << "Square root of 16 is " << sqrt(16) << endl;
    return 0;
}
```

2. User-defined Functions

These are custom functions written by the programmer to perform specific tasks. Example:

```
#include <iostream>
using namespace std;

// Function to add two numbers
```

```
int add(int a, int b) {
    return a + b;
}
int main() {
    int result = add(10, 5);
    cout << "Sum: " << result << endl;
    return 0;
}
```

B. Based on Return Type

1. Function with No Return Value (`void` function)

```
void greet() {
    cout << "Hello, Welcome to C++!" << endl;
}
int main() {
    greet();
    return 0;
}
```

2. Function with a Return Value

```
int multiply(int a, int b) {
    return a * b;
}
int main() {
    int result = multiply(4, 5);
    cout << "Product: " << result << endl;
    return 0;
}
```

C. Based on Parameter Passing

1. Function with No Parameters

```
void showMessage() {
    cout << "Al-Mustaqbal University!" << endl;
}
int main() {
    showMessage();
    return 0;
}
```

2. Function with Parameters

```
void display(int num) {
    cout << "The number is: " << num << endl;
}
int main() {
    display(10);
    return 0;
}
```