CMP 1001

INTRODUCTION TO PROGRAMMING

PART 5: FUNCTIONS IN C++

by

Assist. Prof. M. Şükrü Kuran

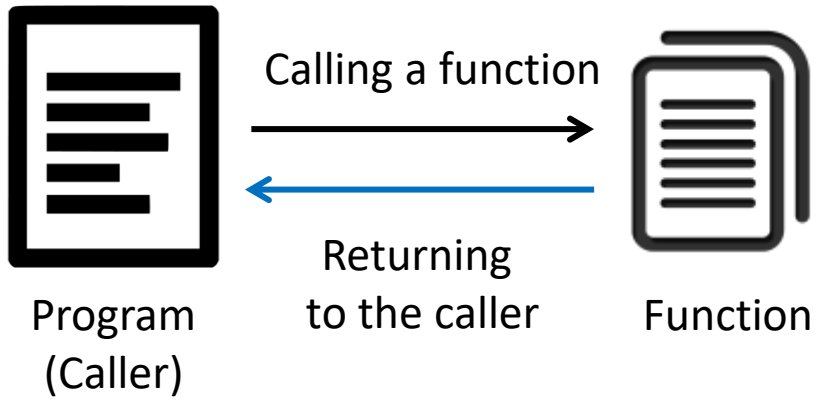
CMP 1001

Section 7: Functions

FUNCTIONS

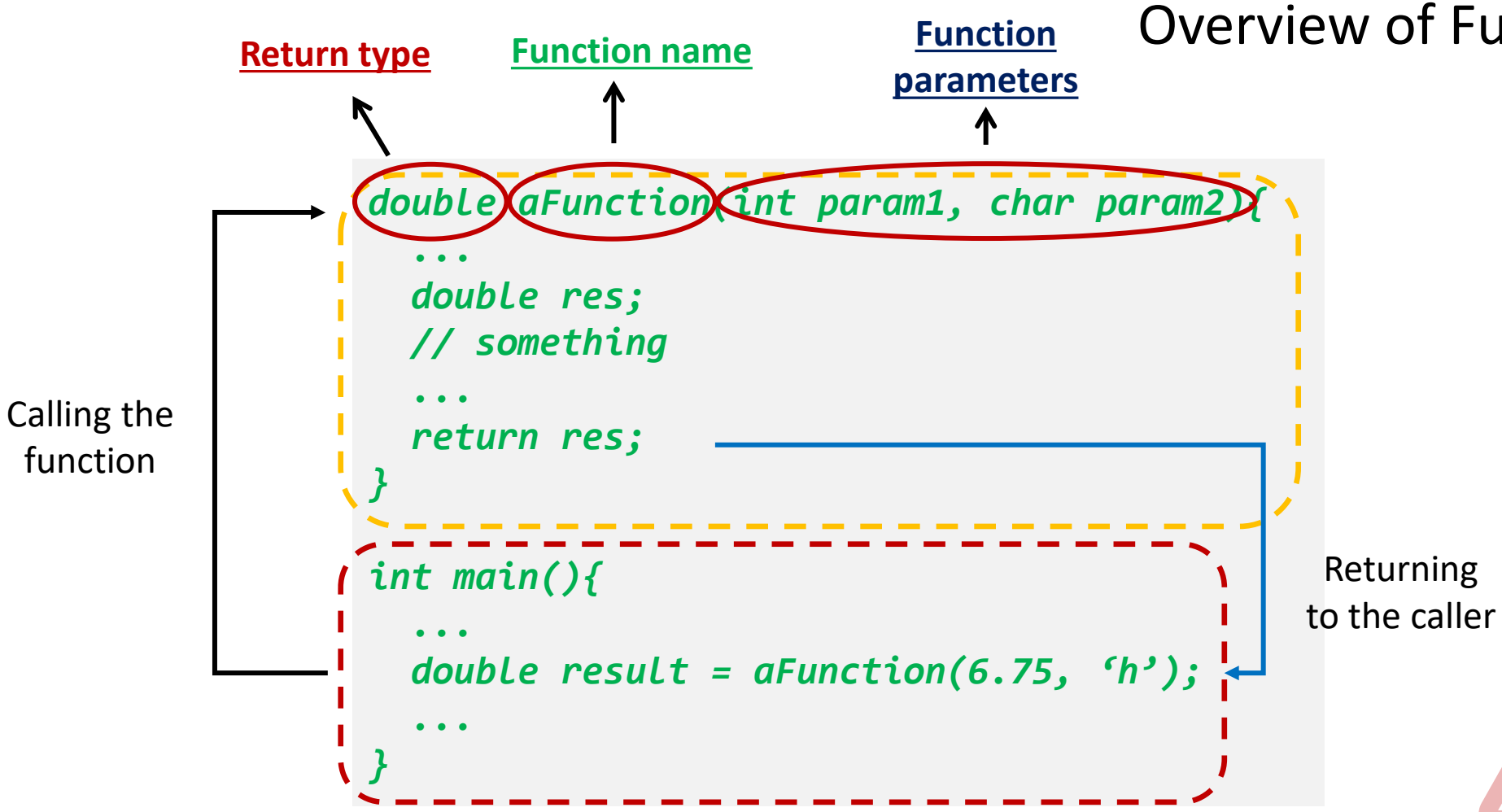
Overview of Functions

"**Functions**" (or Procedures) are part of programs that are self-contained units that achieves **ONE SINGLE WELL-DEFINED OBJECTIVE**.



FUNCTIONS

Overview of Functions



FUNCTIONS

Anatomy of a Function

Function name: The name of the function itself. This name is used when **CALLING** the method from the caller. Rules for method names are the same with variable names.

```
<return_type> <function_name>(<function_parameters>)
```

Return type: This can be **ANY** fundamental data type we have seen. This defines the variable type of the result of the function that will be **RETURNED** to the caller.

Function parameters: The parameter values that are passed to the function from the **CALLER**. There can be 0 to many parameters. Each parameter is given as below separately.

```
(<parameter1_type> <parameter1_name>, …)
```

FUNCTIONS

Example

Example:

Write a function that calculates the square of the given number:

double calcSquare(double number)

Example:

Write a function that computes kth power of n; where k and n are both parameters:

double calcPower(double number, int pow)

Example:

Write a function that takes a character and checks if the value of that character is a lower-case letter:

bool isLower(char character)

FUNCTIONS

Why Functions?

Why do we need to divide programs into smaller units?

A. **Divide-and-Conquer**: Each unit has well-defined simpler tasks. Also using methods are **VERY GOOD** in team projects.

B. **Software Reusability**: Defines code blocks that can be re-used in different parts of a program.

C. **Avoid Code Repetition**: Makes programs easier to debug and maintain.

FUNCTIONS

Example & Notes

NOTE: Functions CANNOT return more than one value.

NOTE: Functions CAN call other functions.

Example:

Write three functions that compute

1. **The factorial f(n)=n!**
2. **The permutation, P(n,k)=n!/(n-k)!,**
3. **The combination, C(n,k)=n!/(n-k)! k!,**

of two numbers n and k, where n>0, k>0, and n ≥ k.

NOTE: The main() function is a special function.
Each C++ program starts from the main() function.

VOID FUNCTIONS

Function that do not return a value

There is also a special type called “**void**”.

Functions with “**void**” return type do **NOT** return any value to the caller.

Example:
Write a function that draws a 5x5 square using the “*” character to the screen.

SCOPE OF DECLARATION

Where a variable is defined?

Each variable has a specific scope based on **WHERE** in the code it is **DEFINED**.

There are three type of variables based on where they are declared:

A. Parameter
B. Local-variable
C. Global-variable

SCOPE OF DECLARATION

```
int X;

void funcA(int param1, double param2){
   ...
}
int main(){
  int number;
  for (i=0; i<10; i++){
    int sum;
     ...
   }
   ...
}
```

Global Variable

Scope: The entire body of the file

Parameter

Scope: The function “funcA”

Local Variable

Scope: The function “main”

Local Variable

Scope: The enclosing for block

Example:

Check code on how the scope of each variable works

RECURSION

Recursive Functions

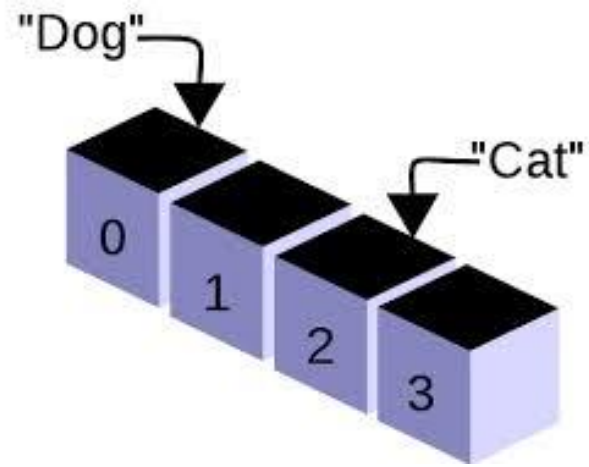
Actually, functions can also call **THEMSELVES**!

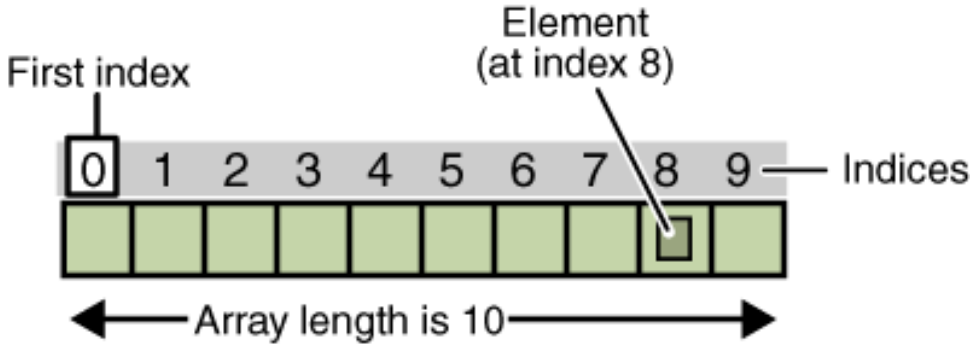
This is called "**recursion**", and these functions are called "**recursive functions**".

Example:
Write a recursive function that calculates the factorial of a given number.

COMING SOON...

Next week on CMP 1001





ARRAYS