Lecture
11_5.1

# Functions, and Strings

**Alessandro SAVINO**
**Politecnico di Torino (Italy)**
alessandro.savino@polito.it
www.testgroup.polito.it

## License Information

## Disclaimer

## *Goal*

– **This lecture presents a global overviews of problems and issues related to functions and strings**

## Prerequisites

– **Basic knowledge of C programming language**

## Homework

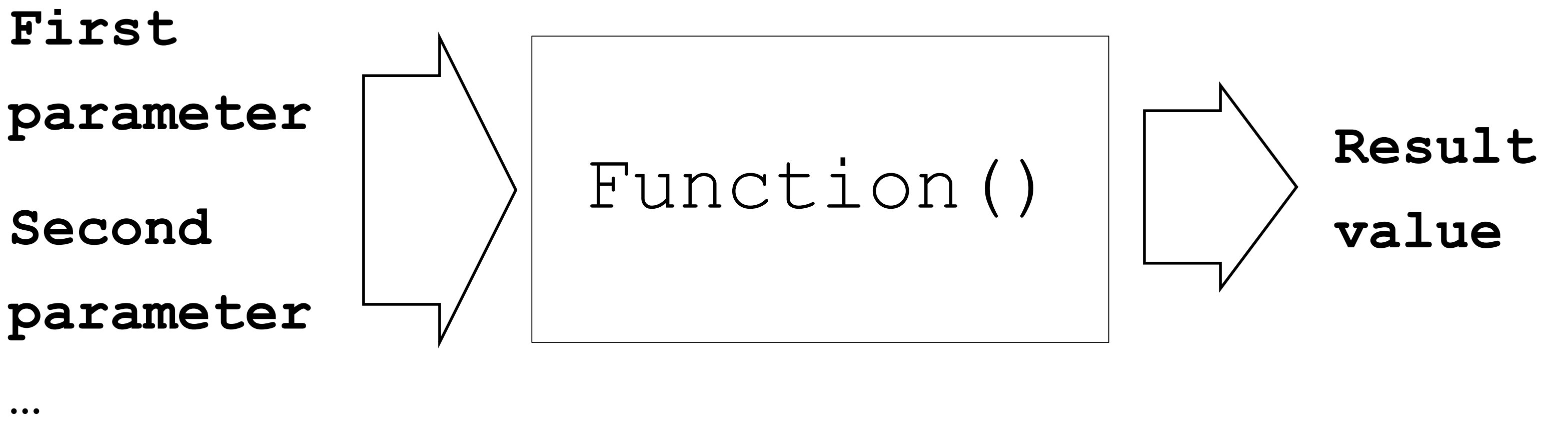
– **None**

## Outline

- **Functions**
- **String**

## Functions

- Functions allow a sequence of statements to be referred to by a name and to be parameterized.
- Calling a function causes the statements sequence to be executed according to the passed parameters and a value may be returned when the function terminates.



Rel. 20/03/2017

## Function definition

- **Establishes a name for a group of operations**
- **Syntax:**

```
<type of result> <function name> (<formal
arguments>)
    {
        <instructions>
    }
```

  - **Use `void` if the function does not return a result**
  - **In <*instructions*> must appear:**
    - `return <value>;      if not void`
    - `return;              if void`
  - **(<formal arguments>) are typed!**
    - `Example: (int a, float b)`

## Prototypes

- **It is a good practice to declare functions at the beginning of the program before their use (prototype)**
- **Syntax:**
  - **Similar to the definition, but we omit the contents (instructions) of the function**

```
int func1(int a);
...
int main ()
{
   ...
   func1(3);
}
int func1(int a)
{
…
}
```

## Default values

- **In C++, it is possible to define default values while creating a function:**

```
int func1(int a = 3);
...
int main ()
{
…
  func1();
  func1(14);
}
int func1(int a)
{
…
}
```

## Default values

- **In C++, it is possible to define default values while creating a function:**

```
int func1(int a = 3);
...
int main ()
{
…
   func1()
   func1(14);
}
int func1(int a)
{
…
}
```

int variable ***a*** is initialized by default to 3

## *Standard libraries*

– **The C++ standard libraries contain different functions. There is no need of *.h* when included.**

| **C++ std library header** | **Explanation** |
|---|---|
| <iostream> | Contains function prototypes for the C++ standard input and standard output functions. This header file replaces header file <iostream.h> |
| <cmath> | Contains function prototypes for math library functions. This header replaces header file <math.h> |
| <cstdlib> | Contains function prototypes for conversions of numbers to text, text to numbers, memory allocation, random numbers and various other utility functions. This header file replaces header file <stdlib.h> |
| <iomanip> | contains function prototypes for the stream manipulator that enable formatting of streams of data |

## Standard libraries - 2

| **C++ std library header** | **Explanation** |
|---|---|
| <ctime> | Contains function prototypes and types for manipulating the time and date. This header file replaces header file <time.h> |
| <cctype> | contains function prototypes for functions that test characters for certain properties, and function prototypes for functions that can be used to convert lowercase letters to uppercase letters and vice versa |
| <cstring> | contains function prototypes for C-style string processing functions |
| <memory> | contains classes and functions used by the standard library to allocate memory to the standard library containers |

## Standard libraries - 3

| **C++ std library header** | **Explanation** |
|---|---|
| <vector>, <list>, <deque>, <queue>, <stack>, <map>, <set>, <bitset> | These header files contain classes that implement the C++ Standard Library containers. Containers store data during a program’s execution |
| <fstream> | Contains function prototypes for functions that perform input from files on disk and output to files on disk. |
| <string> | contains the definition of class string from the standard library |
| <algorithm> | contains functions for manipulating data in the standard library containers |

## *Strings*

- Strings are objects that represent sequences of characters.
- The standard string class provides support for such objects with an interface similar to that of a standard container of bytes, but adding features specifically designed to operate with strings of single-byte characters.
- The string class is an instantiation of the `basic_string` class template that uses char (i.e., bytes) as its character type (see basic_string for more info on the template).

## *Strings – 2*

- A first difference with fundamental data types is that in order to declare and use objects (variables) of this type we need to include an additional header file in our source code: `<string>` and have access to the std namespace.

```
#include <iostream>
#include <string>
using namespace std;
int main () {
   string mystring = "This is a string";
   cout << mystring;
   return 0;
}
```

## Strings – 2

- **A first difference with fundamental data types is that in order to declare and use objects (variables) of this type we need to include an additional header file in our source code: `<string>` and have access to the std namespace.**

```
#include <iostream>
#include <string>
using namespace std;
int main () {
   string mystring = "This is a string";
   cout << mystring;
   return 0;
}
```

## Strings – 2

- **A first difference with fundamental data types is that in order to declare and use objects (variables) of this type we need to include an additional header file in our source code: `<string>` and have access to the std namespace**

```
#include <iostream>
#include <string>

int main () {
   std::string mystring = "This is a string";
   std::cout << mystring;
   return 0;
}
```

## Strings – 3

- Copy a string into another:
  - `destination_str = source_str;`
  - `destination_str.assign(source_str);`
- Concatenates s2 to s1:
  - `s1 += s2;`
  - `s1.append(s2)`
- Comparison: return 0 if s1 is equal to s2
  - `int result = s1.compare(s2);`
  - `int result = s1.compare(0,1,s2,3,2);`
  - `s1==s2` or `s1>=s2` or `s1<s2;`
- Returns the length of s
  - `int result = s1.length();`

Малые Автюхи, Калинковичский район, Республики Беларусь