



# Before we start:

If you feel ill, go home

Keep your distance to others

Wash or sanitize your hands

Disinfect table and chair

Respect guidelines and restrictions

# 02393 Programming in C++

## Module 1: C++ Language Features

**Lecturer:**  
**Alceste Scalas**

(Slides based on previous versions by Andrea Vandin, Alberto Lluch Lafuente, Sebastian Mödersheim)

8 September 2020

# Lecture Plan

#	Date	Topic	Book chapter *
1	01.09	Introduction	
2	08.09	Basic C++	1
3	15.09	Data Types  Libraries and Interfaces	2
4	22.09		
5	29.09		3
6	06.10	Classes and Objects	4.1, 4.2 and 9.1, 9.2
<i>Autumn break</i>			
7	20.10	Templates	4.1, 11.1
8	27.10	LAB DAY	Old exams
9	03.11	Inheritance	14.3, 14.4, 14.5
10	10.11	Recursive Programming	5
11	17.11	Linked Lists	10.5
12	24.11	Trees	13
13	01.12	Exercises & Summary	
	07.12	Exam	

\* Recall that the book uses sometimes ad-hoc libraries that are slightly different with respect to the standard libraries (e.g., strings and vectors).

# Disclaimer

## Remarks on live programming:

These lecture slides **do not spell out** all points covered and discussed during the live programming sessions!

- The slides summarise keywords and concepts
- The final lived-coded program will be on DTU Inside
- We refer to the book chapters as study materials

**If you miss a live programming session**, please make sure that you understand the material in detail, and ask questions to the TAs or in the next lecture!

# Outline

- ① **Functions**
- ② Live Programming
- ③ Exercises and CodeJudge

# Functions

Live programming session today will cover some of:

- Basic data types and conversions;
- Local variables, parameters;
- Several functions;
- Function prototypes;
- Namespaces.

Book chapter 1, especially section 1.6.

# Functions

## An Abstract View

- A function is a sequence of *statements* (instructions) that have been collected together and given a name
- **A bit like in mathematics**, a C++ function:
  - ★ takes some arguments (zero or more)
  - ★ returns a result
- **Unlike in mathematics**, a C++ function is a *procedure*:
  - ★ can have **side effects**, e.g.: print on screen, set **global variables**
    - ▶ two calls with same arguments may return different results!
  - ★ may **not return a result at all**
    - ▶ if return type is void
  - ★ can **modify its arguments** (“call by reference,” later in the course)
- **Scope:** function arguments and **local variables** are only accessible within the body of the function

Bottom line: **a tool to break down a big problem into smaller ones**

# Functions

## A Technical View

Running programs have a **call stack** (a.k.a. execution stack):

- Keeps track of arbitrarily nested function calls
- Function arguments and results are **copied**\* across calls
  - ★ the local variables of the calling function are not affected
- The call stack has a maximum size, and it can **overflow**
  - ★ when performing too many nested function calls
  - ★ when using big data structures as parameters or local variables

---

\*When using “call by value” (we’ll get back to it later in the course)



# Outline

- ① Functions
- ② Live Programming**
- ③ Exercises and CodeJudge

# Live Programming

We will see several examples (see FileSharing file live02) like ...

- $\binom{n}{k}$ : number of combinations to choose  $k$  out of  $n$  values.
  - ★ Example: lottery with 36 balls and we pick 7
- How to compute it?
- For which values of  $n$  and  $k$  is this actually defined?
- What sub-problem do we need to solve?

Formula:  $\binom{n}{k} = \frac{n!}{k! \cdot (n-k)!}$       Example:  $\binom{5}{3} = \frac{5 \cdot 4 \cdot 3 \cdot 2}{(3 \cdot 2) \cdot (2)}$

Can we find a more efficient method?

Other examples: are these equations true in C++?

- $x - y + z = x + z - y$
- $(x + y)/2 = x/2 + y/2$

It depends on the data types we use! Be aware of their limits!

# Outline

- ① Functions
- ② Live Programming
- ③ Exercises and CodeJudge**

# Exercises and CodeJudge

- There is an exercise sheet on Inside File Sharing
- Hand-in via CodeJudge before the next lecture.



# Before we leave:

Disinfect table and chair

Maintain your distance to others

Wash or sanitize your hands

Respect guidelines and restrictions outside