02393 Programming in C++
Module 2: C++ language features

Teacher: Alberto Lluch Lafuente

Sebastian Mödersheim (slides author, course responsible)

February 8, 2016

Lecture Plan

#	Date	Topic	Chapter
1	1.2	Introduction	1
2	8.2	Basic C++	1
3	15.2	Data Types	2
4	22.2	Data Types / Libraries and Interfaces	2 / 3
5	29.2	Libraries and Interfaces	3
6	7.3	Classes and Objects I	4,9
7	14.3	Classes and Objects II	4,9
		Påskesferie	
8	4.4	Classes and Objects III	4,9
9	11.4	Recursive Programming	5-7
10	18.4	Lists and Trees	10.5, 11, 13.1
11	25.4	Trees	13
12	2.5	Graphs	16
13	9.5	Summary	
	17.5	Exam	

Outline

Disclaimer

General note on live programming:
On these lecture slides, we will not spell out all points covered and discussed in live programming sessions!

- We give the key words of the covered concepts
- We put the final version of the developed program on campusnet
- We refer to the chapters in the Stanford reader that cover the material

Especially 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!

Functions

Live programming session today will cover some of:

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

Stanford reader chapter 1, especially section 1.6.

Functions

An Abstract View

- A bit like in mathematics:
  - ★ give an argument/several arguments
  - ★ get a result
- Differences—it is actually a procedure
  - ★ it can have side effects like printing on the screen
  - ★ it can depend on/change global variables
  - ★ thus: two calls with same arguments may produce different results
  - ★ there may not be a result at all:
    if return type is `void`
  - ★ Later: call by reference
- Scope: arguments and local variables are declared only for the body of the procedure

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

Functions

A Technical View

The construction with the stack:

- Allows arbitrarily nested sub-routine calls—up to the size of the stack. (Note: stack-overflows!)
- Also parameters and local variables are handled on the stack!
- When using huge data structures as local variables or parameters, we get into trouble.
- Arguments and results are copied (when using call by value as we did so far): the local variables of the calling procedure are not affected!

Outline

Live Programming

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

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

Live Programming

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

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

Formula:

$$\binom{n}{k} = \frac{n!}{k! \cdot (n-k)!}$$

Live Programming

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

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

Formula:

$$\binom{n}{k} = \frac{n!}{k! \cdot (n-k)!}$$

Maybe not be the best way of computing the function. Can we find an alternative?

Live Programming

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

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

Formula:

$$\binom{n}{k} = \frac{n!}{k! \cdot (n-k)!}$$

Maybe not be the best way of computing the function. Can we find an alternative?

Other examples:

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

These equations may not always hold when working with C++ data types.

Live Programming

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

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

Formula:

$$\binom{n}{k} = \frac{n!}{k! \cdot (n-k)!}$$

Maybe not be the best way of computing the function. Can we find an alternative?

Other examples:

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

These equations may not always hold when working with C++ data types.

Bottom line: Be aware of the limits of the used data types!

Outline

Exercises and CodeJudge

- There is an exercise sheet on campusnet filesharing
- Hand-in via CodeJudge until next Monday before the lecture.