

02393 Programming in C++

Module 1: Introduction

Sebastian Mödersheim

August 29, 2016

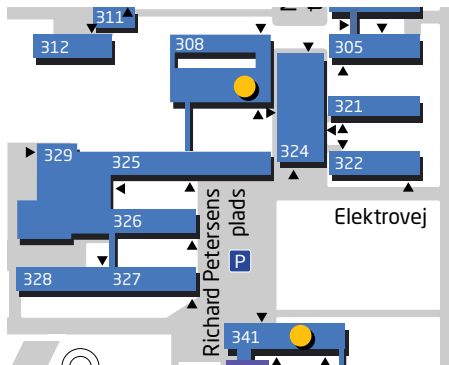
Course Materials by
Sebastian Mödersheim and Alberto Lluch Lafuente

The Team

- Teacher: Sebastian A. Mödersheim (samo@dtu.dk)
Building 324, Room 160. (Please make an appointment.)
- Teaching Assistants:
 - ★ Fabian Goerge s150957@student.dtu.dk
 - ★ Riccardo Miccini s137345@student.dtu.dk
 - ★ N.N.
- Course Responsibles: Sebastian
(and Alberto Lluch Lafuente albl@dtu.dk)
- Help with CodeJudge: Anders Roy Christiansen

See *List of participants* on the course's campusnet page

Location



- Lecture: 5pm - 7pm in building 308, room 12
- Exercises: 7pm - 9pm in [E-Databars 109, 117, 127](#) building 308, ground floor

Evaluation

The course is pass/fail.

- Assignments
 - ★ to be handed in via *CodeJudge*
 - ★ automatically tests your code and gives you the chance to fix bugs
 - ★ please contact the TAs first, if you have trouble getting your exercises accepted by CodeJudge.
- Exam
 - ★ Date: 5.12.2016
 - ★ Duration: 4 hours
 - ★ Electronic exam using CodeJudge

Materials / Literature on C++

- The course slides and programming examples from lecture will be made available on campusnet.
- There are many good books on C/C++ for instance
 - ★ Bjarne Stroustrup: *The C++ Programming Language*
 - ★ Koenig and Moo: *Accelerated C++*
 - ★ KR: *THE* reference for C:
Kernighan and Ritchie: *The C Programming Language*
 - ★ Stanford Course Reader by S. Roberts, J. Zelenski:
Programming Abstractions in C++ (Available from CampusNet)
- Websites like <http://cplusplus.com>, code.org, [techrocket](http://techrocket.com), etc.
- . . .

Lecture Plan

The course consists of 3 major blocks

- Basic C++
- Object-oriented programming in C++
- Advanced topics

Each block covers approximately 4 weeks

Lecture Plan

#	Date	Topic
1	29.8.	Introduction
2	5.9.	Basic C++
3	12.9.	Data Types, Pointers Libraries and Interfaces; Containers
4	19.9.	
5	26.9.	
6	3.10.	Classes and Objects I
7	10.10.	Classes and Objects II
		<i>Efterårsferie</i>
8	24.10.	Classes and Objects III
9	31.10.	Recursive Programming
10	7.11.	Lists
11	14.11.	Trees
12	21.11.	Novel C++ features
13	28.11.	Summary
	5.12.	Exam

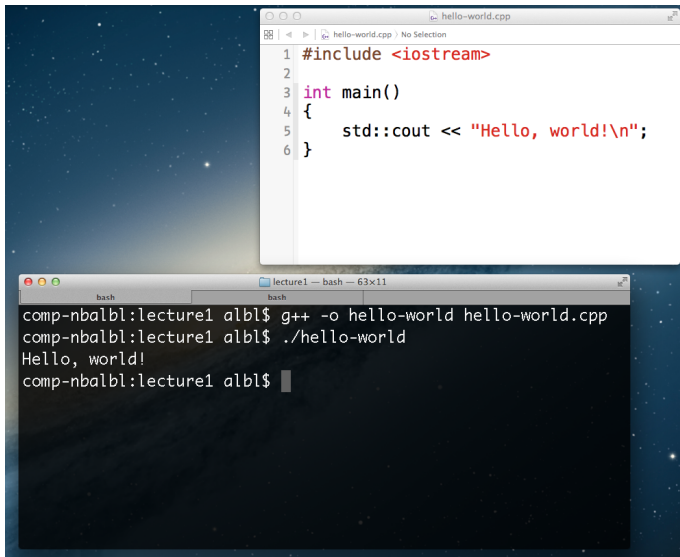
Challenges

- Large class (≥ 100 registered!)
- Late in the evening
- Students with different backgrounds: Mat, CSE, ITCT, MatMod, ManagEng, ElecEng, GeoPh, Ph, NanoTech, ...
- Programming can hardly be learned from a lecture

Ideas for an Effective Course: Live Programming

- Live programming
 - ★ Not much code on slides.
 - ★ Instead: developing a program/example during the lecture
 - ★ We may make smaller exercises together in the lecture
 - ▶ please bring your laptops to the lecture

Live Programming



The screenshot displays a live programming environment with a dark blue, starry background. In the upper right, a code editor window titled 'hello-world.cpp' shows the following C++ code:

```
1 #include <iostream>
2
3 int main()
4 {
5     std::cout << "Hello, world!\n";
6 }
```


Below the code editor is a terminal window titled 'lecture1 — bash — 63x11'. It shows the compilation and execution of the program:

```
comp-nbalbl:lecture1 albl$ g++ -o hello-world hello-world.cpp
comp-nbalbl:lecture1 albl$ ./hello-world
Hello, world!
comp-nbalbl:lecture1 albl$
```

Ideas for an Effective Course: CodeJudge

- Exercises using *CodeJudge*
 - ★ One needs to practice!
 - ★ Immediate feedback from *CodeJudge* which tests your code

CodeJudge



CodeJudge
 Automated Code Judging

Programming in C++ (02393-E14)

[Assignments](#)
[Help](#)

[Logout \(SSO\)](#)
[Alberto Lluch Lafuente](#)

[Assignments](#) » [Exercises Set 2](#) » Gaussian Sum

Exercise

Write a program that computes for a given positive integer n the sum:

$$1+2+\dots+n$$

For example, for $n=100$, the result is 5050.

You should read n from cin, and write the result to cout.

[Sample test data - click here.](#)

The due date has passed! You can still submit solutions, but these will **not** count in the evaluation.

Submit Solution

Language: C++ (g++ 4.8.2)

Comment:


File Source Code

Source Code:


Submit

PresentationExample

TestingOneVariable

Your attempts: 1 Solved by: 62 

MultipleVariables

Your attempts: 0 Solved by: 57 

Your Statistics

Solved Exercises	10%
Submissions	3

Course Statistics

Active users	121
Submissions	2753
Average solved	7.2

Ideas for an Effective Course: Game Programming

- Larger Example of a **small computer game**

Our hope:

- ★ it's fun
- ★ tickle your creative side
- ★ a bit of “real world”:
 - ▶ defining interfaces, using abstraction
 - ▶ combine code from people that have never worked together
 - ▶ find your way in a huge poorly-documented library
 - ▶ sometimes dirty hacks necessary :-)
- Based on a classic 2D computer game series:
Esprit/Oxyd/Enigma
 - ★ By now open source, you may get inspired there!

C/C++ History

C

- Designed and implemented in early 1970 at Bell Labs.
- Used mainly for systems programming (of Unix).
- Ability to access the underlying machine at both a very high level of abstraction and at a primitive and direct level.
- C was designed so that it should be easy for a compiler to generate good assembly code.

C++

Started as a collection of macros and library routines for the C language in order to get Classes/Abstract Data Types:

[Bjarne Stroustrup: *Classes: An Abstract Data Type Facility for the C Language* \(1982\)](#)

Development over many years to integrate features that *make sense for the working programmer*.

Programming Paradigms

- ① Procedural or Imperative Programming
- ② Modular programming
- ③ Object-Oriented Programming
- ④ Generic Programming
- ⑤ Functional Programming
- ⑥ Logic Programming

C++ supports the first 4.

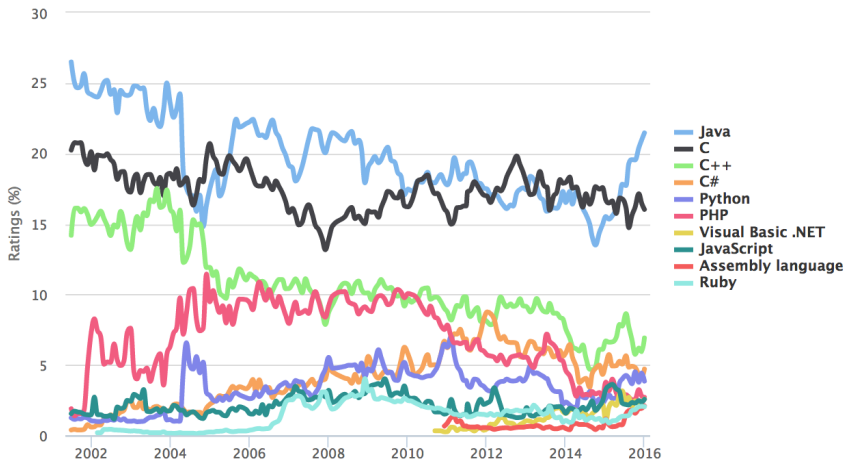
C++ and Efficiency

- C++ allows you to do things that many programming languages forbid
- Advantage: you can program close-to-the-machine and optimize code to the last bit
- Ideal for computer games, hardware drivers, and in general fast processing of data.
- But speed is not everything, there are other interesting properties of code:
 - ★ Free from errors (or at least not full of them)
 - ★ Easy to read/understand and modify
 - ★ General/re-usable
 - ★ Portable
- Do not lose time and code quality by “stupid” optimizations!

C++ popularity

TIOBE Programming Community Index

Source: www.tiobe.com



Hello World

It is custom to first program “Hello World”.
Live programming examples

Useful Tools: Make

- A [Makefile](#) describe how to compile a complex program from a set of source files
- Paths, compiler options, input/output files
- When re-compiling after changes, compile only what is affected by the change
- Standard invocation: change to a directory with the [Makefile](#) and run the command [make](#)
- For the game project we will later give out a [Makefile](#)
- This tool is part of most Linux and Mac installations.
Windows/cygwin: select package [gmake](#).

Useful Tools: Subversion (svn)

Classical problems to keep things in synch:

- After a major change the program does not work anymore.
- Using several different computers.
- Team of programmers sending each other new versions of files per email.

Avoid the mess. Create a repository, e.g. at

<https://repos.gbar.dtu.dk/>

- to store all the files and all versions of your project
- works well with all kinds of text-files!
- More modern (and complicated) alternative to subversion: [git](#)

Useful Tools: Subversion

Working process:

- ① **Talk** with your team members first!
- ② **Check out/update**: get a copy of the latest version.
- ③ Work on the files
- ④ **Check in/commit**: try to get your changes registered as new version in the repository
 - ★ Fails if someone else has committed in the meantime (between step (2) and (4)) on a file you have modified
 - ▶ If the changes are in different parts of the file, the changes get **merged** into your copy and you can try to commit again.
 - ▶ Otherwise you get a **conflict**: you must resolve the conflict manually!

This course has a repository use

`svn checkout svn://repos.gbar.dtu.dk/samo/cpp2016/`

with username `student` and password `yvyebbnnq532ej3b`

Exercises

- **No exercises today.** We start next week.
- If you are using your own laptop, you can already start installing software