

Generic programming and library development

Description:

The purpose of this course is to provide a deep understanding of the C++ programming language and its standard library. Advanced programming techniques are described, and it is shown how these techniques are used in modern library development.

Course team:

- Kenny Erleben <kenny@diku.dk>
- Knud Henriksen <kaiip@diku.dk>
- Jyrki Katajainen <jyrki@diku.dk> (course manager)

Course home page:

[http://www.diku.dk/forskning/performance-engineering/
Generic-programming/](http://www.diku.dk/forskning/performance-engineering/Generic-programming/)

Generic programming

The term **generic programming** has been used in at least four different but related meanings:

1. programming with *generic parameters*,
2. programming by abstracting from concrete types,
3. programming with parametrized components, and
4. programming method based on finding the most abstract representation of efficient algorithms.

[Vandevoorde and Josuttis 2003, §14.5]

Course elements

Cycle: Lecture-Assignment-Discussion-Reading

Lectures: 5 Tuesdays; 2 Fridays

Assignments: 5 in all; given about one week before the deadline

Discussion sessions: 5 Fridays; 1 Tuesday

Reading:

David Vandevorde and Nicolai M. Josuttis, *C++ Templates: The Complete Guide*, Addison-Wesley (2003)

Mini-project: given one month before the deadline 19 June 2007

Oral exam: 26 June 2007 (and 29 June 2007); re-exam in August

Grading

Assignments: 20–40 points

Presentation(s): 10 points

Mini-project: 0–20 points

Oral exam: 0–30 points

Points	Grade
50–57 points	6
58–65 points	7
66–73 points	8
74–81 points	9
82–89 points	10
90–100 points	11,13

Lectures

Time: Tu 24 April; Tu 8 May; Tu 15 May; Tu 22 May; Tu 29 May;
Fri 8 June; Fri 15 June

Place: Store UP1 at DIKU on Tuesdays; D317 at HCØ on Fridays

Contents: We follow the textbook quite closely; a detailed plan is available on course home page.

Keywords: template basics, static polymorphism, dynamic polymorphism, policy classes, template metaprogramming, expression templates, standard-library concepts, smart pointers, function objects, constraint polymorphism

15 June: also course evaluation

Assignments

Handouts: via course home page

Deadlines: Templates Fri 27 April; Tools Fri 4 May; Polymorphism
Fri 11 May; Metaprogramming Fri 18 May; Review Tu 12 June

Group size: 1–3 people

Answer length: 2–4 pages

Answer format: L^AT_EX DIKU-article style available via the CPH STL
home page <http://www.cphstl.dk/WWW/tools.html>

Handing in: electronically via ISIS

No late assignments, please!

Discussion sessions

Time: Fri 27 April; Fri 11 May; Fri 18 May; Fri 25 May; Fri 1 June;
Tu 12 June

Place: D317 at HCØ on Fridays; Store UP1 at DIKU on Tuesdays

Presenters: 2–3 groups present their results of the assignment; 2–3 groups present software development tools

Opponents: 2–4 other groups comment the presentations

Mini-project

Handout: via course home page

Problem formulation: will be made available as a CPH STL report on 20 May 2007

Supporting code: will be made available together with the problem formulation

Group size: 1–3 people

Answer length: about 12 pages

Answer format: L^AT_EX DIKU-article style available via the CPH STL home page <http://www.cphstl.dk/WWW/tools.html>

Handing in: electronically via ISIS by 19 June 2007 at 9.15

Exam

Time: 26 June 2007 (and 29 June 2007); re-exam in August

Place: *nordfløjen* at DIKU

Form: individual, oral, 30 minutes, no preparation (internal examiners)

Pensum: mini-project and the material covered by the lectures

Main changes from the last year

Last year's course evaluation: This is a super course!

Reading: We dropped the other book:

Björn Karlsson, *Beyond the C++ Standard Library: An Introduction to Boost*, Addison-Wesley (2005)

Lectures: We reduced their number by 1.

Assignments: We reduced the number of essays from 2 to 1.

Presentations: We added the presentations on software tools.

Mini-project: We made the project period one week longer.

Exam: We clarified the grading system.