

# Course Analysis: SF2565, Program Construction in C++ for Scientific Computing, 2016

- Course Data**
- Program Construction in C++ for Scientific Computing, SF2565, 7.5 ECTS
  - Period 1/2, 2016/2017
  - Responsibility: Michael Hanke
  - Teaching hours:
    - Lectures/exercises: 24+8 h
    - Computer labs: 2 h
  - Registered students: 33 + 2 PhD students
  - Literature: Lippman/Lajoie/Moo, C++ Primer, 5th ed., Skansholm, C++ direkt, 3:e uppl, lecture slides
  - Credits:
    - homework: 3.5 ECTS
    - Written examination: 4 ECTS
  - Performance index (according to VIS): 62%
  - Examination index (according to VIS): 48%

**Aim** The course provides an introduction to the C++ language both for users and developers of classes with a special emphasis on problems in Scientific Computing. Special care is put on efficient programming. Most of the language features are developed using examples from the numerical solution of partial differential equations.

**Changes compared to the last year** A number of typos in the course material has been fixed.

**Conclusions** This year, the number of answers to the course questionnaire was very large (more than 50% of active students). The course was estimated as having just the right difficulty. It was considered very interesting and meaningful. The homeworks came to the point and their level was just right.

The numerical parts (structured grids and finite difference operators on structured grids) are usually considered as hard to understand. Not only that they went far beyond a basic course in numerical analysis, but the C++ tools to implement them were nontrivial. Therefore, special exercises have been spent to handle them in more detail. These measures gave positive results: It became more understandable, and the hardest parts of the course are now the real advanced features of C++.

In a programming course, where many code snippets are shown, it is unavoidable to use slides extensively. They will be commented on heavily by the teacher. Many students take the lecture slides as they are as their main source of information. However, this is not what they are intended for: They should be considered as a skeleton for the notes taken by the students during the lectures. Therefore, they are published well in advance.

**Teaching** The teaching was done by lectures, exercises, and one computer lab. The latter was intended for students not comfortable with the Linux operating system and the GNU Compiler Suite to get started. Homeworks have been evaluated during lectures or exercises.

**Examination** The examination based on homework problems and a written examination. A successfully solved project 4 gave bonus credits for the written examination.

**Prerequisites** With the exception of certain programming skills, no problem. This concerns, in particular, experiences with developing more complex programs. One student mentioned lack of mathematical background.

**Planned changes** More explicit references to the course literature. Reworking the homeworks.

**Grading** No problems.