Course Analysis: SF2565, Program Construction in C++ for Scientific Computing, 2015

Course Data

- Program Construction in C++ for Scientific Computing, SF2565, 7.5 ECTS
- Period 1/2, 2015/2016
- Responsibility: Michael Hanke
- Teaching hours:
  - Lectures/exercises: 24+8 h
  - Computer labs: 2 h
- Registered students: 33 + 3 PhD students
- Literature: Wilkinson/Allen, Parallel programming 2nd ed., lecture slides
- Credits:
  - homework: 3.5 ECTS
  - Written examination: 4 ECTS
- Performance index (according to VIS): 83%
- Examination index (according to VIS): 79%

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 There was the requirement to come to the “real thing” of developing a complete solution for a pde solver. Since there was no time left in the compulsory project work, an elective project 4 was introduced covering the final step. The latter provides bonus credits for the written examination.

A number of typos in the course material has been fixed.

Conclusions The difficulty of the course received widely varying estimations: from rather easy to very hard. 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) were considered as hard to understand. Not only that they went far beyond a basic corse in numerical analysis, but the C++ tools to implement them were nontrivial. Therefore, special exercises have been spent to handle them in more detail.

I would like to add that I got the impression that some students lack simple skills in programming. In a programming oriented course like the present, this may become a problem. The students were using many different operating system and development platforms. In particular, IDEs were rather different. In order to be able to concentrate on the language, this freedom was intended. On the bad side, it made it impossible to demonstrate the software development process in more detail. I assumed that such aspects have been taken in other courses before.

Teaching The teaching was done by lectures, excercises, and computer labs.

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 programming skills, no problem.

Planned changes More live demonstrations during the lectures.

Grading No problems.