Course Analysis: SF2565, Program Construction in C++ for Scientific Computing, 2018

Course Data

- Program Construction in C++ for Scientific Computing, SF2565, 7.5 ECTS
- Period 1/2, 2019/2020
- Responsibility: Michael Hanke
- Teaching hours:
  - Lectures/exercises: 24+8 h
  - Computer labs: 2 h
- Registered students: 46 + 11 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
- Graduation rate (undergraduate students): 72.1

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. The language features are developed using examples from the numerical solution of partial differential equations.

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

Conclusions The questionnaire has been answered by 13 out of 57 participants.

Most of the students who answered to the questionnaire were present at more than 80% of the lectures. 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. Some students asked for more frequent, but smaller, homeworks. The interest was rather equally distributed over all topics (with th exception of Basics and I/O). Interestingly, the more advanced a lecture the more interest it attained.

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 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. It it was emphasized that the running example was a very good motivation to show the benefits of C++ and the benefits of using it.

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. The lecture slides shall be considered as a skeleton for the notes taken by the students during the lectures. Therefore, they are published well in advance such that they can be written out. This was well appreciated. However, for pedagogical reasons, the information on them was not exhaustive. The intention here was to encourage the use of other sources. Not surprisingly, the internet was the main source of information besides the lecture slides. Even if occasionally done, more "live" presentation of certain C++ features were required.

Teaching The teaching was done by lectures, excercises, and one computer lab. The latter was intended for students not compfortable with the linux operatinmg system and the GNU Compiler Suite to get started. Homeworks have been evaluated during lectures or excercises. According to the answers, the course activities where definitely of help to reach the learning outcomes.

Examination The examination based on homework problems and a written examination. A successfully solved project 4 gave bonus credits for the written examination. According to the students' questionnaire, homework and examination reflected the course's goals very well. The level of the homeworks was estimated very different: ranging from too simple in the beginneing to too advanced in the later papers. This indicates that the level of programming skills in the beginning was rather wide.

The graduation rate is above the average this year.

Prerequisites With the exception of certain programming skills, no problem. This concerns, in particular, experiences with developing more complex programs. It is only one answer which mentioned that his/her prerequisites were not sufficient for this course.

Planned changes Reworking the homeworks. Editing of the lecture slides according to the students' proposals.

Grading No problems.