Course Analysis: SF2565, Program Construction in C++ for Scientific Computing, 2020

Course Data

- Program Construction in C++ for Scientific Computing, SF2565, 7.5 ECTS
- Period 1/2, 2020/2021
- Responsibility: Michael Hanke
- Teaching hours:
  - Lectures/exercises: 24+8 h
  - Computer labs: 2 h
- Registered students: 28 + 7 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): 78.3%

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 The most important change has been that the course was given completely remotely because of the Covid-19 pandemic. This includes even exercises and lab sessions.

Conclusions The questionnaire has been answered by 9 out of 34 participants.

The most important observation in the answers compared to the last years is that the shift to remote lectures and exercises changed the kind the students are comprehending the information. The lectures and exercises where previously mainly organised (at least in important parts) as having a dialogue during the sessions. The remote situation, however, made all of them mostly to a one-way kind of communication. Even if live quizzes are used at certain points, a real discussion was almost impossible to start. As a lecturer, I was missing live feedback. This can clearly seen in the estimation of the lectures and the lecture slides which went down from very good to good-acceptable. Accordingly, the percentage of listeners to lectures went down. This is in sharp contrast to live lectures which were highly estimated and heartily recommended by the former participants.

Despite the more requiring situation for both students and teacher, the examination rate is above the long-term average.

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 the most advanced parts, a moderate fluency in programming was required which led to widely varying quality of the implementations.

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 and discussed with the students. 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 downloaded for making notes. However, for pedagogical reasons, the information on them was not exhaustive. As indicated previously, in a remote situation, these ideas do not seem to be sufficient. Since there is no course book in the traditional sense available, comprehensive lecture notes should be provided. For the time being, the internet is the most important source of information.

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 beginning to too advanced in the later papers. This indicates that the level of programming skills was rather wide.

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 For the future, lecture notes should be prepared.

Grading No problems.