

Course Analysis: SF2565, Program Construction in C++ for Scientific Computing, 2017

- Course Data**
- Program Construction in C++ for Scientific Computing, SF2565, 7.5 ECTS
 - Period 1/2, 2017/2018
 - Responsibility: Michael Hanke
 - Teaching hours:
 - Lectures/exercises: 24+8 h
 - Computer labs: 2 h
 - Registered students: 29 + 4 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): 79%
 - Examination index (according to VIS): 69%

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 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). 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 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. This measures gave positive results: It became more

understandable, and the hardest parts of the course are now in par with 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. 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. Nor surprisingly, the internet was the main source of information besides the lecture slides. The overwhelming recommendation by the students to their classmates was, consequently, Attend the lectures!

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. According to the answers, the course activities were 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.

Prerequisites With the exception of certain programming skills, no problem. This concerns, in particular, experiences with developing more complex programs.

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

Grading No problems.

Here are some answers to selected free-text questions:

- What do you think about the course book?
 - *Used online content instead such as learncpp.com.*
 - *It was, however, perhaps a bit too fundamental at times. Even I who have only taken a handful of programming courses thought it explained many basic things too extensively, but the extra detail was appreciated for the not-so-basic stuff.*
- What do you think about the lecture slides?

- *I think the lecture slides were in general really good and described the concepts well. There was an exhortation in the beginning of the course that you should print out the lecture slides before so you could take notes so this is avoidable, however, When studying to the exam there were a few questions in the slides that was not provided with an answer in writing. The answer was of course given during the lecture but i would be good to have it in the slides as well in case you misunderstood or just missed to answer that question. (Though the left out answer was a good incentive to try to figure it out yourself!)*
- *Lecture slides were good and well structured! A possible improvement could be even more examples and a more 'birds-eye-view of some problems'. ... In a sense, the exam was structured more in this way when it said something like : "construct a class such that the following operations makes sense" (where the operations was spelled out as they would be in the main program) - I think this is a good way of introducing concepts.*
- **What do you think about the lectures from a pedagogical point of view?**
 - *I think the teacher was really good and seemed genuinely interested in getting the students to understand. He was happy provide help and explain the topics further if you did not understand.*
 - *Very, very well organised course. Michael was super clear, knowledgeable and encouraged questions.*
- **Comments about the homeworks**
 - *The homework was at a good level and the difficulty increased at a good pace.*
 - *I think the hard part for me was the numerical analysis part. The C part were not that difficult. This is for scientific computing so I guess the labs suits the topic though I think it would have been nice to have more homework though "simpler" in terms of scope.*
 - *The first one was a good warm-up for C programming. But the two later ones were more rewarding.*
 - *... the third homework made us organise a lot of information ...*
- **Which advice would you like to give to future participants?**
 - *Don't miss the lectures, and especially exercises.*
 - *If not familiar with C before. Focus the first couple of weeks to get familiar with the language and syntax so that you later only have to focus theory.*
 - *Attend the lectures. Dont be afraid to ask questions, you'll for sure get good answers and help from the teacher.*

- *Make sure to have a basic knowledge of simple examples early on. They are easy to find online and can be very informative when thinking about concepts.*