

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, 2018/2019
 - Responsibility: Michael Hanke
 - Teaching hours:
 - Lectures/exercises: 24+8 h
 - Computer labs: 2 h
 - Registered students: 28 + 1 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 own statistics): 77%
 - Examination index (according to own statistics): 63%

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 8 out of 30 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 most interesting chapter was the one about writing of efficient programs.

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 measure 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++. However, occasionally, basic facts from Calculus did not seem to be present.

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. The overwhelming recommendation by the students to their classmates was, consequently, Attend the lectures!

A few participants had problem in debugging programs. In the present course, such skills are assumed. So it could only be touched shortly.

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?
 - *One only improvement I might suggest is that sometimes there are questions to highlight some particular behavior or aspect. These questions are answered in class but not in paper, and I found my notes to be very confusing so I would maybe write the clear answer there as well!*
 - *They were great. I always appreciate teachers that, after introducing a concept, can also present a clear example applying the same concept.*
- Did you get the help you need?
 - *I felt that I was always welcome to ask questions and possibly could get help anytime I needed.*
- Comments about the homeworks
 - *They are just right because I guess some people will struggle with the numerical analysis while others suffer debugging C .*
 - *Some parts was challenging, personally for me it was a bit hard to start with the homework 3. The idea of algebraic grid generation is not hard but it was pretty tedious to grasp how to implement it in code. Luckily, I started really early with the assignement and completed it just in time. So take message home is to start doing assignments as early as possible, at least read the instructions early so you can think about it for a longer period.*
- Which advice would you like to give to future participants?
 - *Do the homeworks on time and ask whenever you dont understand!*
 - *Read the homework instructions from the beginning. Yes, you might not understand how to do them but at least you'll have more time to think about them and you'll definately notice if the concept that you need to implement will be in the lecture slides.*
 - *Try to have nothing that occupies the time the lectures are (like work or another course with same time slots)*
 - *I would recommend to learn C from tutorials which are available in the youtube in an easy way with easy examples in first weeks of the course then everything will be much easier.*