

Kursuskode og navn	DM560 - Introduction to Programming C++
Semester ( Efterår/forår og årstal)	Efterår 2017
Undervisningsansvarlig	Marco Chiarandini
Øvrige undervisere	Troels Risum Vigsøe Frimer (instructor)
Antal tilmeldte studerende til kurset	13
Antal studerende, som har deltaget i evalueringen	6 ~ 50%
På hvilke studieretninger og semestre indgår kurset	The course is elective in the second year of the Physics and Chemistry curriculum.
Hvilken evalueringsform har været anvendt	Electronic questionnaire plus Delphi method in class
Har der været foretaget en midtvejsevaluering på kurset?	Yes.

**Beskriv evalueringens resultater** — *f.eks. indenfor: Kursets opbygning og placering, emner, undervisningsformer, fordeling af øvelsestimer, forelæsninger og e-timer mv, de studerendes arbejdsbelastning, undervisningsmaterialet, delprøver og forudsætningsprøver, sammenhæng med studiet, de enkelte læreres undervisning:*

The course run this year for the first time. It was scheduled to run through one month and a half only. I was assigned to the course after the design of it had already been done, that is, content, book and schedule were already decided when I took up the course.

The course gives 5 ECTS and it is supposed to be an introductory course in Programming assuming no previous knowledge on this topic by the students. The choice of the programming language C++ has been taken by the FKF department. The language is suitable for scientific computing due to its presumed efficiency and safeness against possible bugs. On the other hand it is not easy as a first programming language.

I held 14 introductory classes of two hours and 8 training sessions (some of which of 4 hours). I flipped some introductory classes asking to discuss the slides and the chapter assigned from the book. Moreover, I helped the students resolving several installation issues. In the training classes, students had to work at a few exercises that were given in advance to let them prepare at home on them. It never happened that an exercise was solved before the class. In addition, I gave the option to submit two assignments during the course for feedback. In particular, the second assignment built up on a previous exercise for the training session that resulted to be out of reach for most of the participants. Hence, the second assignment had to be dropped.

The final assessment is pass/fail by the teacher only. It was given after the end of the course. It consists in implementing one of three programs. The three programs were chosen with the intention of meeting the different levels of the students.

All material is available at: <http://www.imada.sdu.dk/~marco/DM560/>.

The course has followed the main indications from the research community around C++ of presenting the language in a modern way, that is, leaving memory management at the last and using vector from the standard library as the main type to avoid issues with memory management.

Several activities were thought as group work in particular work in pairs.

There has been three choices on the tools that deviated too much the attention away from the main content:

- The use of Visual Studio as Integrated Development Environment.
- The inclusion of a graphical library among the list of contents.
- The adoption of git as a way of exchanging files.

The first two are taken from the text book. The last one was an experiment by the teacher.

From the students' evaluations it arises that the majority thinks the following:

- The course has been challenging and the students are overall not satisfied with the final outcome.
- The teacher was not good in understanding where the students might have difficulties.
- The set up of the course was not in order.
- The exercises were not well chosen.
- The pensum was not well suited to the premises of the students and too large.
- The introductory classes were not relevant, most of the learning occurred during the laboratory hours where students were asked to program.
- The introductory classes were much ahead with respect to the exercises. Or the course was too fast.
- The progression has been too steep, with a low start and then a fast increase in difficulty and amount of work.
- The second hand in was too difficult.

On the positive side:

- There were good possibilities to give feedback to the teacher.
- The structure and the expectations have been overall made clear enough and the course page was usable and satisfactory.
- Overall the atmosphere during the course was polite and welcoming.
- The instructor was very helpful.
- The activities in group were helpful.
- The laboratory classes were good.
- The slides were good.
- It was appreciated that the teaching form changed during the course to meet the students status.

**Giver evalueringen anledning til justering af undervisning mv. Hvis ja, beskriv hvilke:**

There are several changes that I would do if I had to teach the course again. Most of these changes were discussed during the last session with the students and encountered their approval.

- Firstly, I would investigate the possibility to spread the course throughout the whole semester rather than throughout one month and a half only.
- I would try to go less in depth and in the details of how things actually work during the intro classes and restrict myself to the practical aspects and to actually make code working rather than understanding technical details. I thought students valued knowing what happens when a program runs but I was wrong. It was a surprise but an important learning for me.
- I would not use Visual Studio and the Windows environment. Instead, I would choose a very basic Integrated Development Environment, such as Note++ or Atom, and use Linux Ubuntu on Windows for building programs. Way too much time was lost in setting up Visual Studio, which is definitely worthwhile in an advanced setting but which is something students can learn afterwards.
- I would ensure that there are more practical exercises with coding and working code from the very beginning. This should be a consequence of the previous point.
- I would leave more time to practical exercises asking to prepare the classes and reducing introductions to a minimum. Assuming preparation has occurred before the class I can use the introductory hours to show how to solve exercises by means of live coding.
- I would try to find a better progression during the course. This implies a revision of the book since I followed closely the book in the sequence of topics. In particular, the calculator example that occupied two chapters turned out to exhibit a quite steep learning curve. However, it was not clear among the students whether this was the issue and they seem to appreciate the book.
- I would give simpler exercises and more focused and make clearer which exercises are incremental in such a way that they can be actually carried out. Some confusion arose on this.
- I do not have a clear idea yet on whether I should skip the introduction of a graphical library or not. Students were also unsure about this. It took a lot of time but it is enticing to work with graphics. I might give it another try because once removed Visual Studio and introduced to Linux command line also this part might become easier.
- I would use a bit more time at the beginning to ensure that everybody can easily set up a basic program and make it working.
- Similarly, I might give another try to git. But this has the lowest priority.
- I would try to do more live coding, in which I show how to solve an exercise from scratch thus giving an example of the whole work flow.
- I would try to give very small self assessment tests in class.
- Finally, I must remember to ensure there are the breaks during the classes.

---

**Giver evalueringen anledning til ændring i kursusbeskrivelsen? Hvis ja; beskriv hvilke:**

I will consider another exam form, for example, written exam in class on more basic tasks. I do not plan changes on the contents.

---

Behandlet af undervisningsudvalget på;

Dato