

Statistics

Results: Introduction to Compilers (OV)

Statistics Skema A

148 could answer this evaluation form

29 have answered this evaluation form

19.59 % answer percentage: $(29/(148)) * 100$

Skema A: 7.5 ECTS credit course

1	My average weekly workload on this course was (incl. lessons, preparation, written work, etc.):		(27 answers)
	Less than 10 hours	0	0.00
	10-15 hours	2	7.41
	15-20 hours	9	33.33
	20-25 hours	7	25.93
	More than 25 hours	9	33.33

Skema A: Course Evaluation

1	Given my background, the academic level of the course is:		
	Far too low	0	0.00
		1	3.57
		15	53.57
		10	35.71
	Far too high	2	7.14
2	In my opinion, the workload on the course is:		
	Far too small	0	0.00
		1	3.57
		7	25.00
		11	39.29
		9	32.14
	Far too large		
3	I believe that I have acquired the competencies described in the course objectives		
	Totally disagree	0	0.00
		2	7.14
		8	28.57
		8	28.57
	Totally agree	10	35.71
4	In my opinion, the individual subelements (lectures, exercises, etc.) of the course were logically connected		
	Totally disagree	0	0.00
		4	14.29
		6	21.43
		10	35.71
	Totally agree	8	28.57
5	In my opinion, the teaching material was relevant to the course		
	Totally disagree	0	0.00
		2	7.14
		3	10.71
		11	39.29
	Totally agree	12	42.86
6	In my opinion, I have received relevant academic feedback on my oral and written work on the course		
	Totally disagree	0	0.00
		2	7.14
		4	14.29
		7	25.00
	Totally agree	15	53.57
7	In my opinion, I have had access to the necessary information about the course		
	Totally disagree	0	0.00
		2	7.14
		4	14.29
		12	42.86
	Totally agree	10	35.71
8	Overall, I find that the course has been useful		
	Totally disagree	0	0.00
		2	7.14
		3	10.71
		9	32.14
	Totally agree	14	50.00

9 What was good about the course? Why?

(29 answers)

- That you are allowed to make the G-assignments on your own, meaning that you never have to show up at the university. Also, the W-assignments are reasonable unlike to those of "Algorithms & Data Structures".
- Cosmin Eugen Oancea was very inspiring and helpful.
- Generelt er det et rigtig godt kursus med god sammenhæng mellem forelæsninger, øvelser og projekt.
- The instructor (Kristoffer) was good. So was both teachers. It was good with the extra officehours!
- G-opgaven! Det har været fedt at arbejde på et større projekt, og jeg tror det er første gang i min tid på DIKU hvor jeg er blevet sat til at arbejde videre på en eksisterende kodebase.

• The non-stop help from the professors and instructors.
Support: ku@arcanicsupport.dk +45 4525 7272

- The logical order of introducing concepts, the elaborate slides and good material, which helps a lot when studying the subject.
- Spændende. Kunne godt lide at vi fra starten havde overblik over kursets indhold og trinvis gik videre til hvert skridt i oversættelsesprocessen.
- I like that we were both introduced to the theoretical aspect in the lectures and the exercises, and the practical aspect in the groupwork. I think it is very important to work with such heavy theory in practise. That ensures a much better understanding, and is somehow (in my opinion) more important than the theory itself most of the time. The much needed help for the groupwork were available through extra office hours, and the lecturers were generally wellprepared and helpful.
- I liked that there were office hours every friday. It was nice that the teachers were always checking on how far we were the the group assignment.
- Kurset er velplanlagt og godt struktureret
- Cosmin & Jost did an excellent job teaching, by far some of the best teachers i've had so far at DIKU.
The group project was a nice practical exercise! I like the difference between G and W assignments.
- It's nice that the material is written for the course and the book is fine. The teachers did a good job in the lectures. Good choice of languages.
- It is really usefull to see how a compiler is implemented. Also the parsing and lexing part seems useful in other applications as well.
- I think the course overall was great!
- The lectures and the office hours since they where most helpfull in the heaviest parts of the course
- it was very well taught, and the material, although advanced, was understandable. I had a very dedicated instructor, who gave useful feedback.
- Nice with code heavy course. Really nice support from TA's and Professors(Which was needed to keep the workload realistic)
- The G-assignment gave a really good insight into compilers in general and got me felt prepared for the exam.
- Committed teachers especially Cosmin. Detailed explanations.

10 **I would like to propose the following improvements**

(29 answers)

- Make it easier to find Simon Shine's guide "An Introduction to ML-Lex (for Moscow ML)". It is your *ONLY* decent mosmllex guide. Even now, I've forgotten how to find it again.

Your testfile logical.pal can never possibly return "true".

The fibRec and fibWhile tests calculate fib(read()+1) instead of fib(read()). fibArray does it correctly.

If you make a single spelling error in mosmllex, the .output file will be empty.

In top-level sml, you can't run more than one .pal file without closing again first. It seems that some compiler stage *SAVES* the main procedure, and simply adds the procedures to those already there.

In the G-assignment, it is incorrectly stated that further up a %left, %right or %nonassoc precedence rule is written, the higher the precedence. It is, in fact, the opposite.

Make another text field in the evaluation for comments that are neither praise nor suggestions.

Make another evaluation after the exam.

- less ambitious assignments. I felt that the group assignment was very ambitious and too complex given that the short length of the course.
- Det var nødvendigt at bruge væsentligt mere end de 20 timer for at nå det hele. Det er især de to første ugentlige afleveringer, som krævede for mange timers arbejde.
- Make the first two-three weekly assignments a bit easier.
- At Jost og Cosmin prøver at lave W-opgaverne mere ens - der var meget stor forskel, alt efter hvem der havde lavet dem, på både omfang og format.
- Slow a bit down on the individual assignments and maybe some introduction to git(version control).
- The amount of work it takes to understand the SML code given is way out of balance with learning the principles of compilers. I spend more time reading SML than I did on reading compiler theory, which I see as a problem for the course. Indeed, it is certainly a great language to build a compiler with, but I would consider rethinking the c ode part a bit. Not to say that it should be eliminated, but it should definitely be made easier.
- Ikke så god sammenhæng mellem den teoretiske og praktiske del af kurset. De individuelle ugeopgaver var kedelige og virkede ikke særlig relevante for ens forståelse. Forelæserne laver nogle gange deres egen udgave af fremgangsmetoderne beskrevet i bogen, og det er forvirrende at skulle skelne mellem de to. Øvelsestimerne virker lidt meningsløse. I de sidste par uger sad vi bare og arbejdede på G-opgaven, mens instruktoren ventede på spørgsmål.
- The one big problem in this course is that the students had to spend way to much time:
 1. Catching up with SML
 2. Getting familiar with the structure of the handed out compiler. (This was the big one)
 I spend way to much time trying to figure out the complex structure of the different files which form the compiler. The lecturers and/or the instructors should have thoro ughly introduced and talked us through the structure, that would have saved a lot of time.
- The documentation on the already implemented parts of the compiler, like RegAlloc.sml, could have been better.
- I personally felt a need for more actual code examples and I think it's quite natural. I know that there is a focus on the principles and not the actual coding, but to be able to pass the exam and finish the group assignment it is crucial that we're able to code in SML and MIPS. I think it's little too optimistic to think we're able to code in SML out of the blue since most of us haven't used it for a whole year. A lot of the time a felt that teachers and TA's didn't know or couldn't understand that there was a huge gap between what we knew/could remember from SML and what they expected of us.
- the g assignment should be individuel
- Desværre bruger man for lang tid på ugeopgaverne til, man kan nå at lægge den nødvendige tid i G-opgaven
- In the beginning of the course the G-assignment seemed way out of proportions, this is due to the midway hand-in, where we had the first 3 weeks to do it, but didn't learn the necessary material until just a week before it was due which seemed stressful.
- The book was shit.
- Make the G-assignment code base a bit shorter if possible and perhaps have the students implement more of the basic stuff themselves, possibly in earliers exercises. I'm still thinking that there must be some prettier and nicer lexer/parser generators than the mosml{lex,yac} ones, which would be an improvement. Keep the language functional though!
- In the classrooms we almost never did the assignments because we were talking about the w or g assignment, which is perfectly fine, but I think that the extra exersices could have been cut down, or more exiting then.

I don't think it makes sense to put this survey before the course is finished.

- Re-evaluate the workload of the week-assignment. It was nowhere near the estimated "3-4 hours".
- Workload is waaay to heavy!
- Instead of getting a partially finished compiler, i would prefer starting from scratch.
- A bit more focus on how the parts of the compiler works together would have been appreciated, it was not completely clear.
- The workload in the beginning was far too large.
- I miss an overview of the compiler we're working with. I don't think it helped much to see the code snippets at lectures. I think workload should be adjusted downwards.
- I found it a bit stressful with the G-assignment colliding with some of the W-assignments. It would have been better with easier W-assignments while the G-assignment was due. It were especially the two first W-assignment there was bit too hardcore in my opinion.
- The workload of the weekly individual assignments are way to high, especially the first weeks. Of course it's important to know the weekly pensum, but some parts of the assignments where just about writing a lot of information down into tables, several times. Which wasn't always necessary. Furthermore there where to much code on the slides in the lecture. It's should be up to the student to look at detailed code at home, and only showing the necessary information in the slides.
- I was a very good and informative course but the workload was very high. Both the W and G assignment took longer than the described time. Also sometimes there was errors in the assignments that the instructors hadn't noticed because the had not worked through the assignments themselves.