



Kursanalys - KTH¹

Formulär för kursansvarig.

Kursanalysen utförs under kursens gång.

Nomenklatur: F – föreläsning, Ö – övning, R – räknestuga, L – laboration, S – seminarium)

KURSDATA Obligatorisk del ²

Kursens namn

Scientific Programming
(Programmeringsteknik)

Kursnummer

EL2310

Kurspoäng och poäng fördelat på exam-former

7,5hp

När kursen genomfördes

period 1, ht2008

- LAB1 2,5hp (matlab)
- LAB2 2,5hp (C)
- LAB3 2,5hp (C++)

Kursansvarig och övriga lärare

Patric Jensfelt (ansvarig)

Kristoffer Sjö

Undervisningstimmar, fördelat på F, Ö, R, L, S

F: 30h, R: 12h

F: 2h

Antal registrerade studenter 29

Prestationsgrad efter 1:a examenstillfället, i % 52%

Examinationsgrad efter 1:a examenstillfället, i % 41%

MÅL

Ange övergripande målen för kursen

Kursens övergripande mål är att ge deltagarna grundläggande kunskaper inom programmering och då speciellt MATLAB, C och C++ samt att hur programmering kan användas för att lösa olika problem. Det senare målet innebär att kursen även har ett inslag av problemlösning.

Ange hur kursen är utformad för att uppfylla målen

¹ Instruktioner till kursanalysformulär sist i dokumentet

² Rektors beslut: <http://www.kth.se/info/kth-handboken/II/12/1.html>

Kursen är uppdelad i tre delar. Det inleds med MATLAB där grundläggande programmering varvas med en introduktion till MATLAB som ett verktyg. Därefter behandlas C och sist C++. Varje lektion varvar repetition, ny information med praktiska exempel. En (frivillig) laboration genomförs för varje del och examinationen är ett individuellt projekt inom varje språk.

Programmering är någon man lär sig genom att göra. Man kan lära ut grundläggande ideer och försöka inspirera till god programmeringstil men i slutändan handlar det om att öva, öva, öva. Det är därför som kursen examineras genom ett projekt i varje språk. Nivån på programmeringskunskaperna som krävs är sådana att någon som redan kan språket har relativt lätt att lösa uppgiften ur ett programmeringstekniskt hänseende. Det som återstår är övningen att översätta ett matematiskt, mekaniskt, etc problem till ett programmeringsproblem.

Eventuellt deltagande i länkmöte före kursstart

Synpunkter från detta

Kursens pedagogiska utveckling I

Beskriv de förändringar som gjorts sedan förra kursomgången. (Berätta även för studenterna vid kursstart)

De största ändringarna från förra året är att i) inga labbar genomfördes i labsal denna gång eftersom studenterna inte utnyttjade dessa förra gången ii) MATLAB-projektet som ansågs vara för enkelt förra året nu gjorts aningen mer utmanande. En del av det som gjordes förra året i den labben är kvar, det som testade enklare beräkningar och plottning men en helt ny del som handlar om scan-matching har lagts till. Föreläsningarna hölls mer eller mindre som första gången.

Kontakt med studenterna under kursens gång

Studenter i årets kurs-nämnd:

Namn

E-post (lämnas blank vid webbpublicering)

Resultat av formativ mittkursenkät

Resultat av kursmöten

Kontakt med övriga lärare under kursens gång

Kommentarer

Kursenkät; teknologernas synpunkter Obligatorisk del ³

Att komma ihåg:

- 1) Uppmana, mha kursnämnden, till ifyllande av kursenkät i anslutning till / just efter slutexaminationen
- 2) Delge kursnämnden enkäten
- 3) Publicera enkäten under en kortare tid

Period, då enkäten var aktiv Juni 2009

Frågor, som adderades till standardfrågorna Alla frågor och svar bifogas denna analys

Svarsfrekvens 11 studenter av 29 som registrerade sig varav 21 får anses ha för avsikt att slutföra kursen.

³ Rektors beslut: <http://www.kth.se/info/kth-handboken/II/12/1.html>

Förändringar sedan förra genomförandet	
Helhetsintryck	På det hela taget tycker jag att studenterna ger en positiv bild av kursen genom enkäten. Det är en tuff krs för de som inte haft någon kontakt med programmering alls innan men det är inte helt orimligt tycker jag. Det medges att det är en svår balansgång mellan att vara för svår för de som inte kan programmera och totalt ointressant för de som redan kan.
Relevanta webb-länkar	
Kursansvarigs tolkning av enkät	
Positiva synpunkter	Majoriteten tycks vara nöjda med kursen och föreläsningarna. Det är glädjande att projekten är intressanta även för de studenter som redan kan programmera.
Negativa synpunkter	De studenter som inte kan programmera alls tycker att kursen är mycket tuff. En del av det som krävs i C-projektet kräver lite för mycket extrastudier.
Var kursen relevant i förhållande till kursmålen?	Det är min bedömning ja.
Syn på förkunskaperna	
Syn på undervisningsformen	De flesta är nöjda med den även om labbarna har en lite oklar roll eftersom så få gör dem.
Syn på kurslitt/kursmaterial	Det finns ingen kursbok och de flesta tycks vara nöjda med det.
Syn på examinationen	De flesta verkar nöjda även om en del tycker att det är svårt.
Speciellt intressanta kommentarer	Att de delar av C-projektet som behandlar sockets och kommunikation måste tas bort eftersom det r för komplicerat.
Synpunkter från övriga lärare efter avslutad kurs	
Vad fungerade bra	
Vad fungerade mindre bra	
Resultat av kursnämndsmöte efter examination	
Studenternas sammanfattn.	
Förslag till förändringar	
Länk till kursnämndsprot.	
Kursansvarigs sammanfattande berättelse	
Helhetsintryck	Det utan tvekan största problemet med kursen är att folk inte blir klara med den i tid. Detta beror till stor del på att i) jag varit alldeles för snäll med att skjuta deadlines och inte ha någon slutlig deadline efter vilket man får vänta till nästa år ii) att studenterna i SCR har 3 obligatoriska kurser och denna får stå åt sidan för de andra två. Detta är nu åtgärdat i programmet inför nästa år.
Positiva synpunkter	Det är trevligt att även de som kan programmera finner kursen intressant.

Negativa synpunkter	Kursen har väldigt låg prioritet bland studenterna och det är väldigt svårt att få dem att göra saker i tid. Detta trots att de flesta inte spenderar mer än sammanlagt 40h på alla projekt tillsammans.
Syn på förkunskaperna	Det finns få ämnen där spännvidden är så stor i förkunskaper som i programmering och detta märks tydligt.
Syn på undervisningsformen	Programering lär man sig genom att göra. Labbar och projekt tycks för mig vara det enda sättet att få studenterna att göra.
Syn på kurslitteratur/kursmaterial	De flesta tycks vara nöjda med att det inte finns någon kursbok och jag ser ingen anledning att ändra på det.
Syn på examinationen	På tok för långsamt
Kursens pedagogiska utveckling II Obligatorisk del ⁴	
Hur förändringarna till denna kursomgång fungerade	Jag tror att MATLAB-projektet blev aningen för svårt eftersom det inte var tillräckligt tydligt att syftet var att lösa det rent programmeringstekniskt och inte att göra den perfekta lösningen på scan-matching.
Förändringar som bör göras inför nästa kursomgång	Öka kraven på att leverera projekten i tid.
Övrigt	
Kommentarer	

⁴ Rektors beslut: <http://www.kth.se/info/kth-handboken/II/12/1.html>

EL2310 SCR intro

Results from the course evaluation

Some questions to answer

1. What is your overall impression of the course?

1. 42% (5 st) Very good
2. 33% (4 st) Good.
3. 8% (1 st) OK.
4. 8% (1 st) Not so good.
5. 0% (0 st) Really bad.

2. Where the goals and content of the course clear from the start?

1. 42% (5 st) Very good
 2. 33% (4 st) Good.
 3. 8% (1 st) OK.
 4. 8% (1 st) Not so good.
 5. 0% (0 st) Really bad.
-

3. What did you think about the lectures?

1. 8% (1 st) Very good
2. 42% (5 st) Good.
3. 8% (1 st) OK.
4. 8% (1 st) Not so good.
5. 0% (0 st) Really bad.

4. How did you like the way problems were solved interactively at the lectures?

1. 33% (4 st) Very good
2. 8% (1 st) Good.
3. 25% (3 st) OK.
4. 0% (0 st) Not so good.
5. 0% (0 st) Really bad.

Comments on the lectures:

Improve the quality of the projector, because the screen is not clear enough when some code is written i.e. in Matlab.

I did not attend. There is no option for "not attended".

Size of the assignments is maybe a little too big.

5. Another idea with the course was to illustrate some techniques in robotics. How well do you think that this worked overall?

1. 42% (5 st) Very well.
2. 42% (5 st) Well.
3. 0% (0 st) Ok.
4. 0% (0 st) Not so well
5. 0% (0 st) Not at all

Comments on the robotics aspect of it:

The most interesting point with the course was its strong relevancy with the SCR program problems.

In that sense, it was a rewarding course for both the student who already had knowledge about programming as well as those who did not have much experience in programming.

A little more hands-on on the real robots..

maybe some testing or trying to actually control the robots.

6. There was no course book in the course. How did you feel about that?

1. 33% (4 st) Perfect solution.
2. 17% (2 st) Good solution
3. 33% (4 st) Ok.
4. 0% (0 st) No so good.
5. 0% (0 st) I would have liked to have bought some books.

Comments:

Websites are better than books on programming, because they are updated periodically. Good links are provided.

I think it is much more reasonable to learn programming based on some specific lecture notes; at least when it is taught for a specific reason.

The lecture notes and lab instructions were helpful.

7. Did you use a textbook on Matlab?

1. 0% (0 st) Yes.
2. 92% (11 st) No.

Which?:

I usually use mathworks notes.

8. Did you use a textbook on C?

1. 0% (0 st) Yes.
2. 92% (11 st) No.

Which?:

9. Did you use a textbook on C++?

1. 17% (2 st) Yes.
2. 75% (9 st) No.

Which?:

C Primer Plus

C tutorial on Web

10. How did you like the lab assignment (not project) in matlab?

1. 50% (6 st) I did not do it.
2. 33% (4 st) Good.
3. 8% (1 st) Bad.

11. How did you like the project in matlab?

1. 50% (6 st) Very good
2. 33% (4 st) Good.
3. 8% (1 st) OK.
4. 0% (0 st) Not so good.
5. 0% (0 st) Really bad.

Comments:

12. How much time did you have to spend on the MATLAB project (to meet

the requirements)?

1. 8% (1 st) 0-4 hours
2. 0% (0 st) 5-8 hours
3. 33% (4 st) 9-12 hours
4. 17% (2 st) 13-19 hours
5. 33% (4 st) 20+ hours

13. Besides programming, did you learn anything from the MATLAB project?

1. 42% (5 st) Yes a lot
2. 42% (5 st) Yes.
3. 8% (1 st) Not so much.
4. 0% (0 st) Nothing.

Comment:

I knew about scan matching before.

Visualizing any steps of the problem was something that was done in the best way during the matlab project. I was really impressed with the idea.

I came to KTH with a pretty shallow knowledge of matrix algebra.

Everyone at KTH seems to have a good grasp of using linear algebra to represent 2D geometry. I wish there were a 5 page primer on that as it would save some time.

14. If you did not finish the MATLAB project yet, why?

1. 75% (9 st) I am finished.
2. 0% (0 st) I did not have time.
3. 0% (0 st) Too difficult.

15. How did you like the lab assignment (not project) in C?

1. 50% (6 st) I did not do it.
2. 33% (4 st) Good.
3. 8% (1 st) Bad.

16. How did you like the project in C?

1. 42% (5 st) Very good
2. 33% (4 st) Good.
3. 8% (1 st) OK.
4. 8% (1 st) Not so good.

5. 0% (0 st) Really bad.

Comments:

Too long and complex

It was very C actually! if I remember it correctly. There were some small points in the implementation so that if somebody did not noticed them, they could make a lot of suffer; such as parsing a sequence of formatted data. A small bug could make everything go wrong. It could cause disgusting impression for the beginners.

Did spend an insane amount of time on position integers overflowing. There is a hint about this in the project description but I probably would have appreciated if it was highlighted or made clearer (or that I was smarter/paying more attention)

17. How much time did you have to spend on the C project (to meet the requirements)?

1. 17% (2 st) 0-4 hours
2. 8% (1 st) 5-8 hours
3. 33% (4 st) 9-12 hours
4. 17% (2 st) 13-19 hours
5. 17% (2 st) 20+ hours

18. Besides programming, did you learn anything form the C project?

1. 25% (3 st) Yes a lot
2. 50% (6 st) Yes.
3. 17% (2 st) Not so much.
4. 0% (0 st) Nothing.

Comment:

I knew about path planning and world modeling before.

I got some ideas about the robotics problems.

The project really helped me understand some of the issues in SLAM, path planning, control, etc. A nice intro. My thoughts and prayers were with people who did not know C already, as their battle with understanding System V sockets, bit-level manipulation and pointer arithmetic evoked imagery of Stalingrad in WW II.

19. If you did not finish the C project yet, why?

1. 67% (8 st) I am finished.
2. 8% (1 st) I did not have time.
3. 0% (0 st) Too difficult.

20. How did you like the lab assignment (not project) in c++?

1. 50% (6 st) I did not do it.
2. 42% (5 st) Good.
3. 0% (0 st) Bad.

21. How did you like the project in C++?

1. 8% (1 st) Very good
2. 75% (9 st) Good.
3. 0% (0 st) OK.
4. 8% (1 st) Not so good.
5. 0% (0 st) Really bad.

Comments:

Not enough lectures on C

It was relatively easy yet it showed some great potentials of object oriented programming.

I saw some of the students who did not really realized the idea with object oriented programming.

Maybe if they should have re-implemented in C the same project which they had already done in C then they could have sensed how much object oriented concept can be helpful. Well, then they would need to implement the project in C in the right way.

It was a nice intro into problems of planning.

22. How much time did you have to spend on the C++ project (to meet the requirements)?

1. 17% (2 st) 0-4 hours
2. 17% (2 st) 5-8 hours
3. 17% (2 st) 9-12 hours
4. 0% (0 st) 13-19 hours
5. 25% (3 st) 20+ hours

23. Besides programming, did you learn anything form the C++ project?

1. 0% (0 st) Yes a lot
2. 33% (4 st) Yes.
3. 33% (4 st) Not so much.
4. 8% (1 st) Nothing.

Comment:

It was a neat intro into probabilistic path planning.

24. If you did not finish the C++ project yet, why?

1. 58% (7 st) I am finished.
2. 17% (2 st) I did not have time.
3. 0% (0 st) Too difficult.

25. Things that were good about the course:

As there are students with many different backgrounds in programming, whatever you do it will be impossible to fulfil everybody's expectation. So I would say it was good, although it was easy for me.

- *Objective of the course*
- *Help of the professor*
- *Practical examples at the lectures*
- *No deadlines to submit the projects*

Introducing programming to people who do not have the background is a great idea.

The projects were defined in a very nice format. It was very much inspiring for me as a student to have a clue about the robotics problem in a very simple format.

The examples in the lectures are very good.

At absolutely no time did I find the in-class explanation convoluted. The projects were motivating and clearly illustrated some of the areas that are going to be tackled in more depth later in the program. The course was an absolutely essential crash course in MATLAB that definitely helped with other courses. I am not sure if it was sufficient for people to learn C/C , but it was definitely good as a refresher.

Interesting projects.

Plenty of useful links in the course homepage that lead to the tutorial documents and sources.

It was good that there was no attendance requirement, so the ones that spent considerable time on programming before, did not have to sit there. Also, it was good that the theme was on robotics, so even if we knew programming there was still a lot to learn.

Interactive lectures, problem-orientated assignments, possibility to ask questions about the assignments and additional information more deeply about the topic

26. Things that should be improved:

- *Almost impossible to learn 3 programming languages within 3 months*
- *Quality of the screen*
- *Give another options than Cygwin, which sometimes doesn't work in some PCs...*

I don't know how much it has been discussed in the lectures that programming is a critical tool for the continuation of the program. I feel some of the students did not take it seriously. They just wanted the course to be passed. I'd like them to be aware of the amount of projects they will have to do with the programming knowledge they could gain from this course, specially matlab programming. I'm sure they would take it much more seriously if they knew how much they would need it in their early future.

There are components of the C Project that could have been exercised through a few homeworks. A lot of the project workload focused on bitwise operations, sockets and understanding a hefty protocol description and then the student still had to apply some robotics ideas. I guarantee you that everyone would benefit from a nice 2D geometry with Linear Algebra primer, even in form of one PDF tutorial one homework. Example topics: point, line and plane distances, circles, ellipses, spheres, intersections, surface areas, normals and projections on lines and planes. Just a useful time-saving thing that could be covered in at most 2 classes (probably in even less time) and covered with a homework or two. The problem is the common formulas and tricks are (at least for me) hard to find in one place and they do show up in more than 2 required courses later on, so it would help students "warm up" to these ideas.

I would not be able to do any of the projects if this was the first programming course I was taking, at least I would not be doing them "the right way."

The projects, especially the first two ones could be made simpler. This would also discourage people from trying to get their projects done by someone else.

I know the reason that there were no hard deadlines on projects was to let people take their time to learn, but as far as I could see, it did not work that way. It seems like this made people postpone doing it rather than spending as much as time they needed to spend. So I would say the course would benefit from harder deadlines.

patric@csc.kth.se

This was generated [ACE](#).