

# eValuering 2.1 - Sammanställning

---

## SAMMANSTÄLLNING

---

**Utvärdering:** Constraint Programming (autumn 2010)  
**Antal svarande:** 21

---

---

---

## WELCOME!

---

Please fill out the survey below to provide the assigned teachers and the department with feedback. Don't forget to click on "Submit Evaluation" when the form is complete. Your answers are anonymous.

---

---

## QUESTIONS

---

The following questions are required by the faculty board. You must answer them.





*What is your general feeling about the course?*

Svarsalternativ	Graf (%)	%	$\Sigma$
1 (= bad)		0	0
2	■	5	1
3	■	19	4
4		0	0
5 (= good)	■	76	16



Medelvärde: 4.48

Standardavvikelse: 0.98



The total amount of work on the course, in relation to the credits (10 högskolepoäng = 10 ECTS credits = 267 hours of work expected), was ...

Svarsalternativ	Graf (%)	%	Σ
1 (<230h)		10	2
2 (230..250h)		0	0
3 (250..280h)		38	8
4 (280..300h)		38	8
5 (>300h)		14	3
Medelvärde:	3.48		
Standardavvikelse:	1.08		

Did you at the start of the course receive information about previous course evaluations and measures taken because of them?

Svarsalternativ	Graf (%)	%	Σ
no		0	0
yes		86	18
I don't know		14	3

Did you get the opportunity during the course at a scheduled time to give anonymous written feedback on the ongoing course (in short: was there a mid-course evaluation)?

Svarsalternativ	Graf (%)	%	Σ
no		0	0
yes		95	20
I don't know		5	1

---

## TELL US WHAT WE SHOULD KNOW!

---

### What in this course has been particularly good?

- a special way of thinking. with the knowledge learned, we can deal with some practical problems. <sup>1</sup>
- Assignments are very well dcheduled and allow a practical application of the theory presented during lectures using a real-life tool. Also the fact that it is over 2 periods allows a sound grasping of the main points. <sup>3</sup>
- Pierre's style of teaching is very communicative, he definitely knows his stuff and is able to transport it. The assignments were tough but interesting - how I like it. The opportunity for making bonus points was motivating. Gecode is awesome and showed me quite a few things about software design. VERY good to work with. In general, I was very happy with the course. <sup>5</sup>
- I liked the project the most where you could use all your skills to make a good model. <sup>6</sup>

- The content, the guest lectures, the instructor (Pierre). One of the best courses I've taken at UU<sup>7</sup>
- example models to help understand how constraint programming is used to solve problems<sup>8</sup>
- Lecturer was very passionate about Constraint Programming so it was interesting to attend the lectures and it also gave more motivation for studying. Instructions for the assignments were prepared very well and included all the details that a student might have needed.<sup>11</sup>
- The subject was very interesting. I've always wanted to know how problems such as the ones given during the course get solved, now I have a good idea about it. Actually it was the most interesting course I've taken so far.<sup>12</sup>
- The course is very interesting, especially in applications of constraint programming. I like the detail slides of this course, it help me a lot to understand lectures.<sup>13</sup>
- - The material taught during this course is great. - The lecturer is very open to any kind of questions. - The material is always available in advance.<sup>14</sup>
- I really liked the assignments, they were fun to do and I learned quite a bit.<sup>15</sup>
- Pierre Flener is an exceptional teacher. The combination of abstract theory in lectures and practical implementation in labs is rewarding and helpful. Assignments are fun and Farshid is helpful with them. Guest lectures were interesting. All in all, this course has been the best I have attended.<sup>16</sup>
- - The lectures, Pierre is awesome! - The subject is very interesting, I didn't even know it existed before and now I am very interested in it.<sup>17</sup>
- The lectures have been good and the subject is very interesting. Pierre lectures are detailed and kept the interest up the whole way. Few lecturers are as convincing and secure as him. Farshid has been very good as well, giving good hints and explaining things in a good way. The assignments were varied and gave a nice hands on experience with constraint programming, as well as a better understanding of the theory behind it. The project was just right in terms of size, contents and difficulty. All in all, the whole course has been particularly good!<sup>18</sup>
- 1) Standard of assignments 2) Lecturer's knowledge on the subject<sup>19</sup>
- really interesting subject. lectures awesome up to midpoint, then rather shapeless<sup>20</sup>
- Labs and assignment, you really learned how to program constraint programs. Exam felt not so good, too much of the standard questions. I would really like to get a correct version of what you wanted for answers on the questions this year so I can learn something from it. But I guess this information can't be handed out since they could be answer on next years exam? I would prefer new questions next year so I can see what the correct answers where.<sup>21</sup>

### **How could the course be improved?**

- some times the assignment is a little bit difficult. we have to read a lot of extra papers to solve them. more guides before the deadline will be better.<sup>1</sup>
- More modelling exercises. Maybe include the steiner triples as an assignment, it's a relatively easy exercise to solve and good preparation for the exam.<sup>4</sup>
- The rules how to hand in where - quite frankly - ridiculous. The need to write huge reports was boring and didn't teach anyone anything. Maybe it would be better to let students fill in a predefined latex-document. At some points, I was sure, that no one will ever actually read the table, I am creating right now --> frustrating The final exam focused on creating huge

tables (the propagation-problem, mostly). It was extremely easy to make small mistakes that could sabotage the outcome of the algorithm severely, even with full understanding. That problem was written for robots, not for students and took me almost half of the time, I worked on the exam. I hope this fact is acknowledged in the grading process.<sup>5</sup>

- Information about how to compile, install cpp and graphics in gist. It took a lot of time that could be spent on modelling.<sup>6</sup>
- Sometimes it felt like Farshid wasn't really sure on how to solve the assignments (e.g. the empty strip dominance) and they were sometimes quite unclear on what exactly should be done (e.g. what to fill into the table when simulating the master propagation algorithm).<sup>7</sup>
- Certain topics were difficult to understand and it would be helpful if more time was spent to elaborate on this. In particular the definitions of propagators and how you can manually use that to perform propagation.<sup>8</sup>
- In my opinion some of the assignments did not require much thinking, but were more about doing quite a lot of manual work. I would have preferred assignments with more programming and problem solving.<sup>11</sup>
- - Giving a better introduction to Gecode. - Finally documenting Gecode! Well it is a great library I have to say, the MPG guide is quite good, but documentation makes a programmer's life much easier. - Giving more modelling examples at the beginning of the course.<sup>12</sup>
- - It's not bad at all to use slides from many sources, but feels a little strange (can this be improved?) - Sometimes the lecturer makes some meta-analysis on the course, i.e. he speaks about teaching the course itself, and I'd like to recommend avoid doing that. - It may be a good idea to divide the lecture about symmetry in two parts, and also to talk about the distinct propagator while we are working on the third assignment. - I don't know if is a good idea to simulate a real-life scenario by forcing students to work in pairs, the subject is complex enough and working alone can be better to go deeper. I'd also really like to see the assignments to be synchronized with the lecture contents (I know you have reasons to do not doing so). - You can organize better the sessions related to the labs. For example, some tips before beginning the labs can save students a lot of time, you could also book simultaneously a Unix lab. Send notifications to our e-mail addresses when something that we must read is published in the forum.<sup>14</sup>
- Have some more introduction before the first assignment on how to set up/compile gecode and some other more code related stuff, like an intro lab.<sup>15</sup>
- MPG MPG MPG. Admittedly, the documentation is not all that bad, but it does seem to be lacking in examples. At every point I needed an example to see possible implementations, the example used was of no help whatsoever. And trying to find new ways to solve problems is very difficult, since the Gecode webpage documentation is very sparse on details. I would suggest a short list of useful/helpful functions for new students. An even better solution (though more time consuming) would be an API akin to what Sun/Java has, where functions are listed with a short explanation, and possibly an example. That resource is very easy to use while clearly explaining what functions do.<sup>16</sup>
- - Better (and earlier) introductory lectures to Gecode. At least two weeks before the deadline of the first assignment. - The final exam was pretty long. I think there might be better ways to prove that a student knows how to propagate without having to fill a \*huge\* table.<sup>17</sup>
- If there's time, an early hands-on lab with Gecode would have been nice, although reading MPG and following the examples given in the first few chapters gave a good start as well. It

could not have been stated more clearly that reading MPG was a good idea => <sup>18</sup>

- set time to pass for bonus credit. 10 minute meeting with asistant where code/evaluated. idtat once and it was great. farshiid is great and could arrange that in a really constructive way. <sup>20</sup>

- A small lab in the begining, that is not anykind of examination, but only there for the student to step by step write a small and easy to understand constraintprogram befor the first lab. This in order to get the most learning out from the first lab and not have to put to much work on understanding how to encode with gecode but instead on how to program a constraint program. This doesnt have to be a lab but could me a hand-out with a step by step instruction on how to write and compile a small constraint program on the unix system. <sup>21</sup>

Please be informative and constructive.