

---

<sup>b</sup>  
**UNIVERSITÄT  
BERN**

Vizerektorat Lehre, iLUB, Hochschulstrasse 6, 3012 Bern

Persönlich/Vertraulich  
Prof. Dr. Oscar Nierstrasz  
Institut für Informatik und angewandte Mathematik  
Neubrückstr. 10  
3012 Bern

Vizerektorat Lehre

**Lehrveranstaltungsevaluation**

Report of evaluation: FS19 Programmierung 2 (2417)

Dear Mr./Mrs. Prof. Dr. Nierstrasz

Please find here the results of the evaluation of your course "Programmierung 2". Following the scanning of the questionnaires, this report was automatically generated and mailed to you.

The questionnaire used was appropriate to the course type Vorlesung. In the report, you first see the mean values of the most important dimensions:

- Conveying the course content
- Course materials to assist Learning
- Commitment of the lecturer
- Complexity and Scope
- Assessment of Individual Lectures

In the second part of the report, you see the answers to all the questions. The number of answers, the mean value and the values differing from it are also given.

Grade 1 on the left hand side equals the lowest grade given by the students, grade 5 or more on the right hand side the highest grade. In 'complexity and scope' grade 3 corresponds to 'exactly right' and is therefore the best grade. In the overall assessment of the course, grade 6 means the best result.

The free comments at the end of the questionnaire are only read by the lecturer him/herself and won't be evaluated statistically. Please don't pay much attention to negative statements of single persons. You are to look closely in case of frequent occurrence of similar comments.

Please briefly discuss the results with your students before the end of the semester. You will find a presentation template on the last pages of the report. By giving serious consideration to the feedback of the students, you can contribute to higher future response rate.

In case you wish to learn more about how to improve your teaching, you might want to discuss the results with the staff of the 'Hochschuldidaktik' (mail address: [hd@zuw.unibe.ch](mailto:hd@zuw.unibe.ch)). Please bring a copy of the report with you, since the staff of Hochschuldidaktik do not have access to evaluation results.

You might find guidelines, regulations, and information about the process under [www.lehrveranstaltungsevaluation.unibe.ch](http://www.lehrveranstaltungsevaluation.unibe.ch) (documents in German).

Should you need more information, you may also contact us by e-mail.

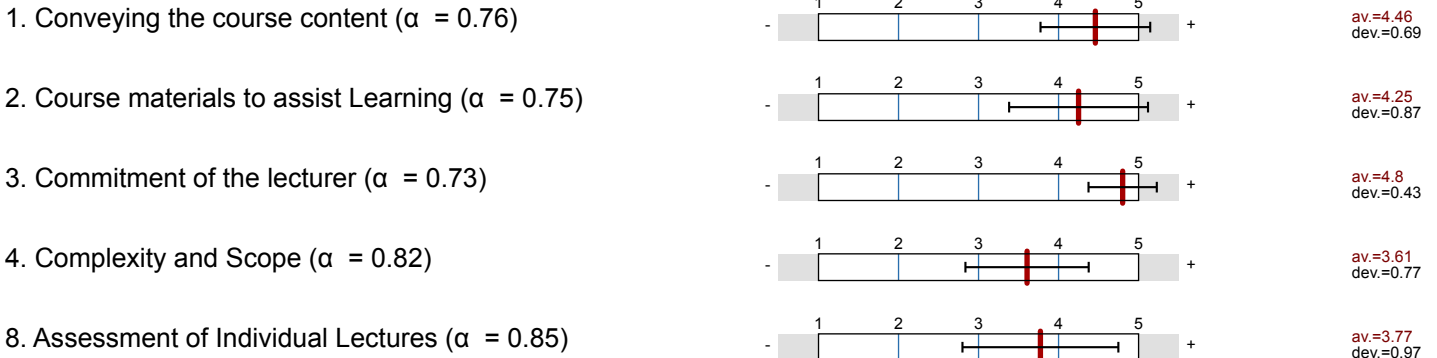
Yours sincerely

D. Wuillemin  
Evaluation office  
Vice-rectorate of teaching

Daniela Wuillemin  
Supportstelle für ICT-gestützte Lehre und  
Forschung (iLUB)  
Hochschulstrasse 6, 3012 Bern

Tel. +41 31 631 51 07  
[lehrevaluation@ilub.unibe.ch](mailto:lehrevaluation@ilub.unibe.ch)  
[www.lehrveranstaltungsevaluation.unibe.ch](http://www.lehrveranstaltungsevaluation.unibe.ch)

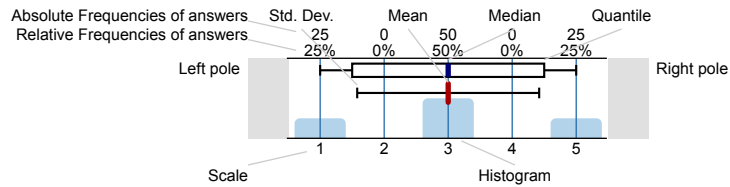
Overall indicators



Survey Results

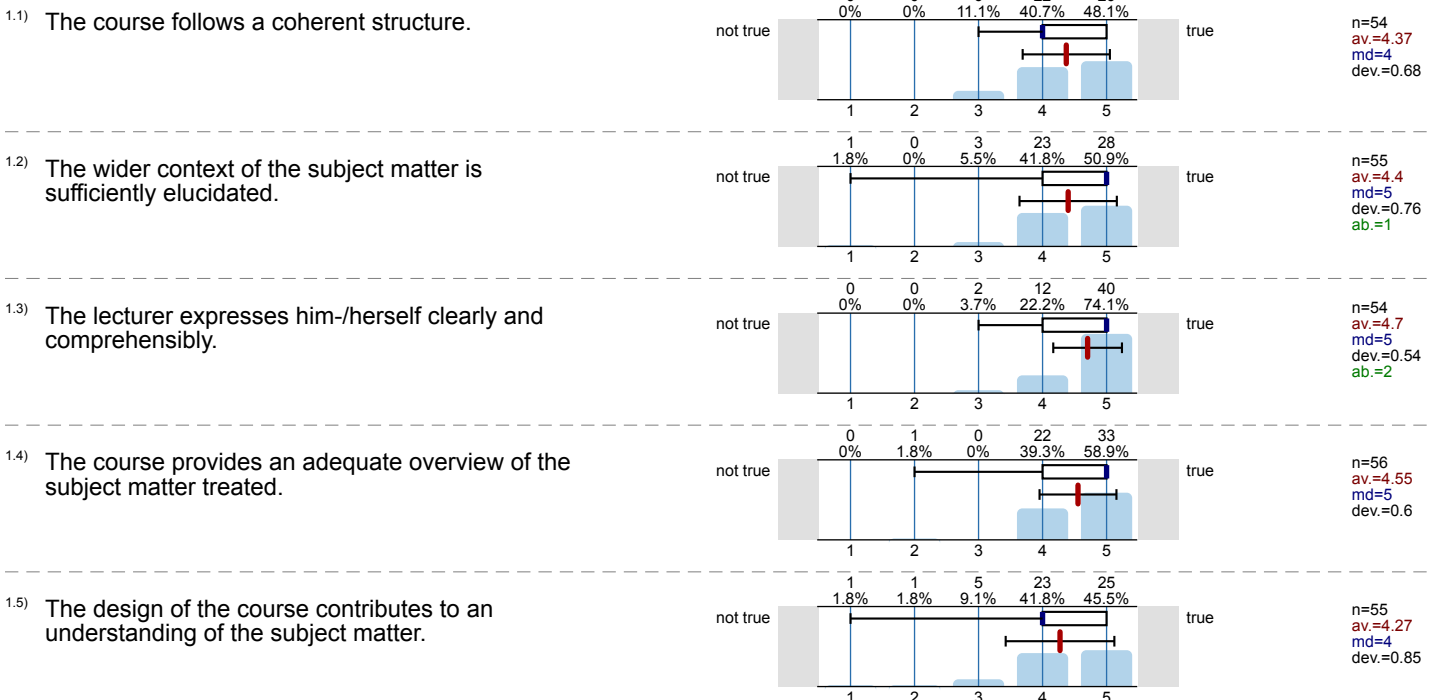
Legend

Question text



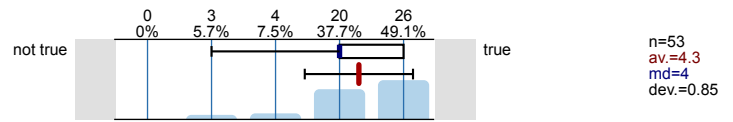
n=No. of responses  
av.=Mean  
md=Median  
dev.=Std. Dev.  
ab.=Abstention

1. Conveying the course content

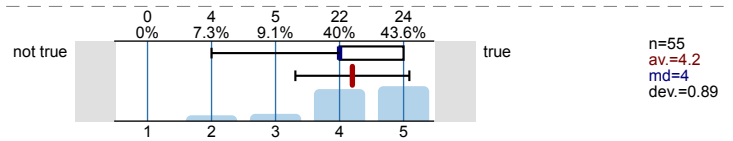


## 2. Course materials to assist Learning

2.1) There is overall enough material provided to assist the learning process (slides, course material, hand-outs, etc.).

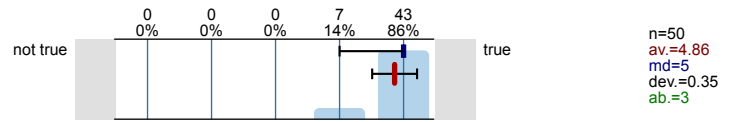


2.2) The course materials (slides, course manuals, hand-outs, etc.) are overall of sufficient quality.

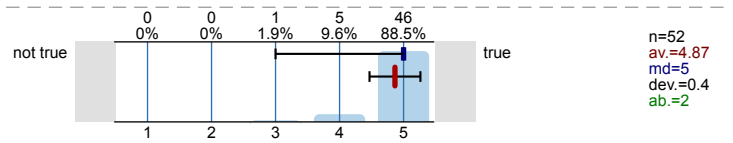


## 3. Commitment of the lecturer

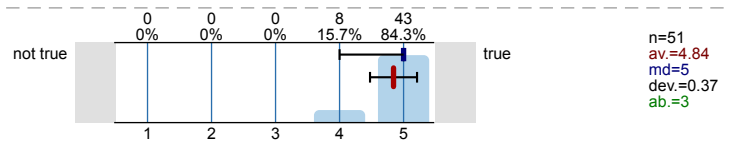
3.1) The lecturer takes students seriously.



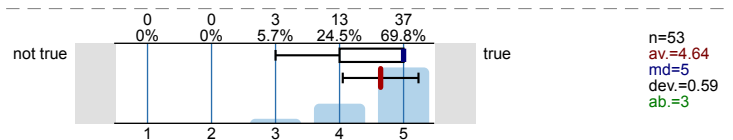
3.2) The lecturer is friendly and respectful towards students.



3.3) The lecturer addresses questions and suggestions from students adequately.

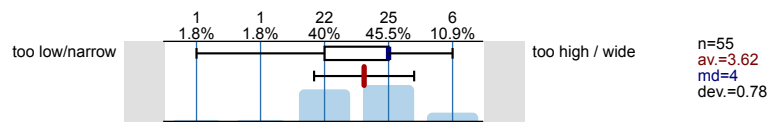


3.4) The lecturer seems to care about his/her students' learning progress.

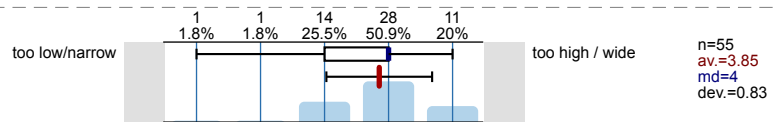


## 4. Complexity and Scope

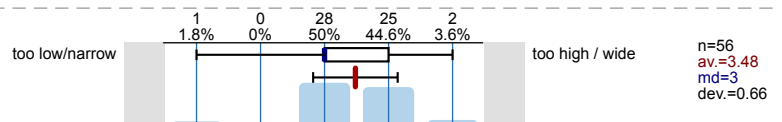
4.1) The degree of difficulty of the course is:



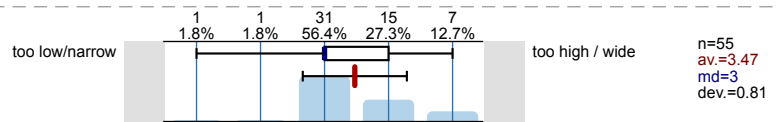
4.2) The amount of content of the course is:



4.3) The pace of the course is:

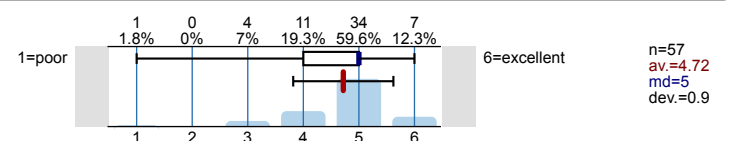


4.4) The amount of knowledge presupposed by the course is:

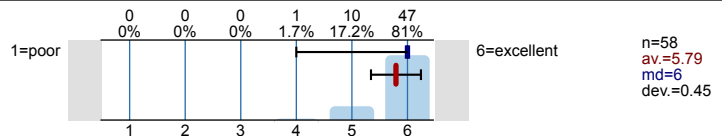


## 5. Overall Assessment

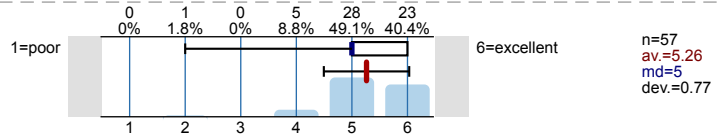
5.1) How would you grade the course as a whole?



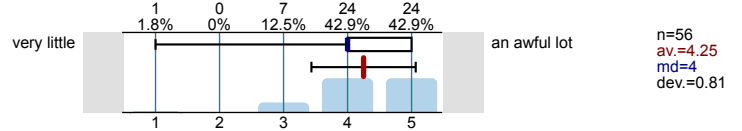
5.2) How would you grade the lecturer with regard to subject expertise?



5.3) How would you grade the lecturer with regard to teaching methods?

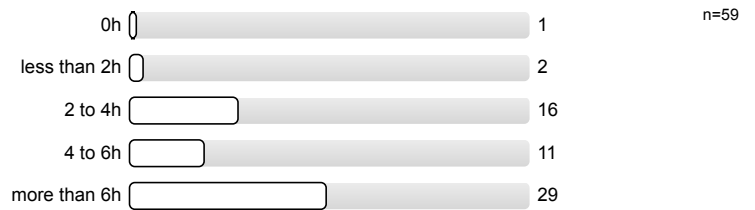


5.4) The course has taught me:



## 6. Socio-demographic Data and Background Variables

6.1) How many hours per week did you invest in preparation and revision for the course (on average)?



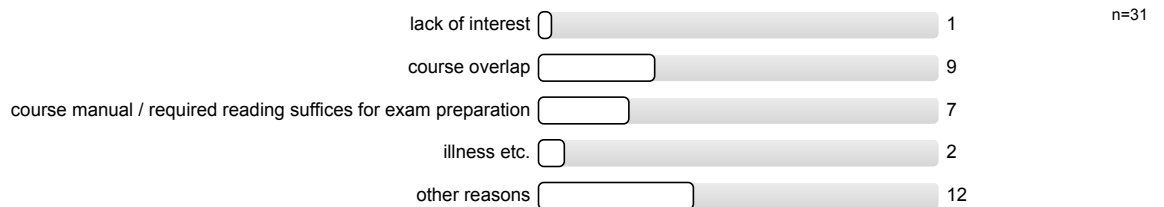
6.2) Was the topic of interest to you?



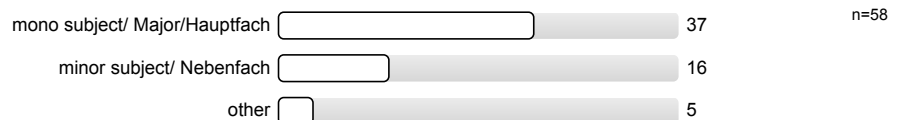
6.3) How many lectures did you miss?



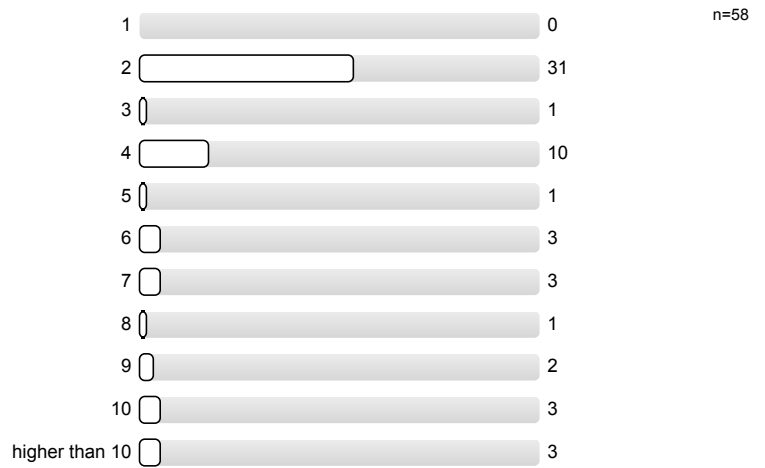
6.4) If you missed more than 2 lectures, please give one reason:



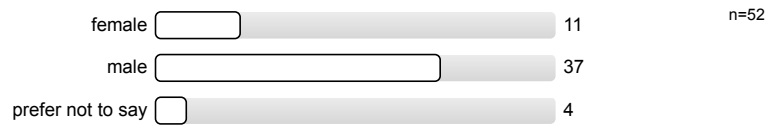
6.5) Allocation of the course in your study programme:



6.6) Your current number of semesters since starting your studies:

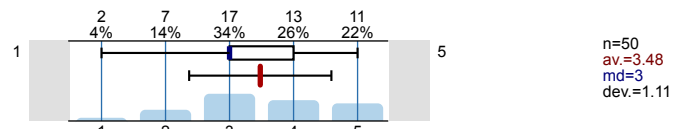


6.7) Sex:

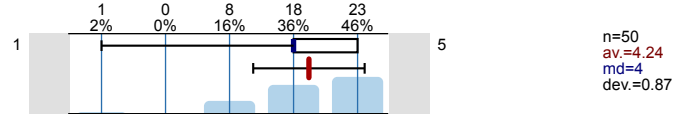


## 8. Assessment of Individual Lectures

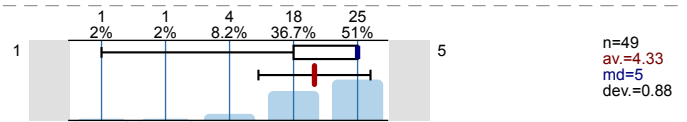
8.1) Introduction



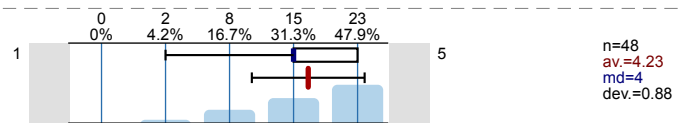
8.2) OO Design Principles



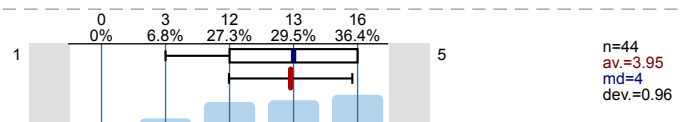
8.3) Design by Contract



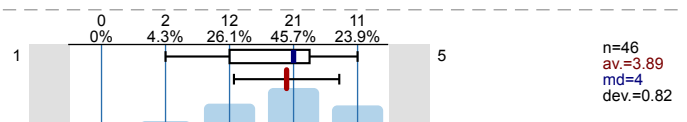
8.4) A Testing Framework



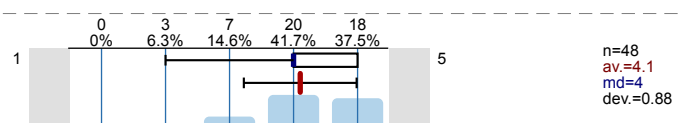
8.5) Debugging and Tools



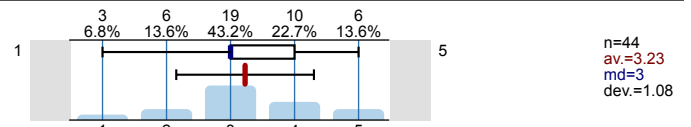
8.6) Iterative Development



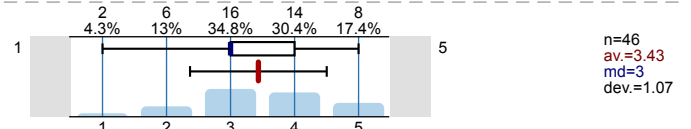
8.7) Inheritance and Refactoring



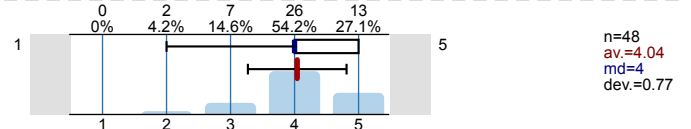
8.8) Advanced Design Lab



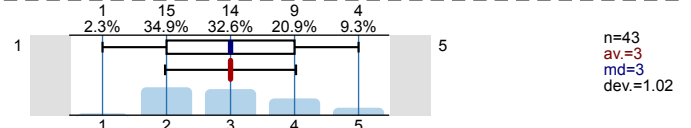
8.9) GUI Construction



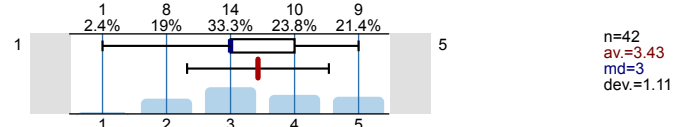
8.10) Guidelines, Idioms and Patterns



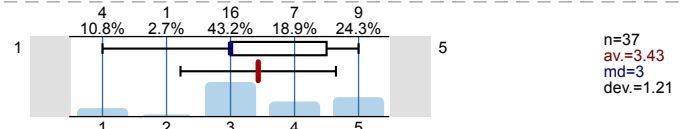
8.11) A bit of C++



8.12) A bit of Smalltalk



8.13) Einblicke in die Praxis



# Profile

Subunit: Phil.-nat. Fakultät  
 Name of the instructor: Prof. Dr. Oscar Nierstrasz  
 Name of the course: Programmierung 2  
 (Name of the survey)

Values used in the profile line: Mean

## 1. Conveying the course content

1.1) The course follows a coherent structure.	not true		true	n=54	av.=4.37	md=4.00	dev.=0.68
1.2) The wider context of the subject matter is sufficiently elucidated.	not true		true	n=55	av.=4.40	md=5.00	dev.=0.76
1.3) The lecturer expresses him-/herself clearly and comprehensibly.	not true		true	n=54	av.=4.70	md=5.00	dev.=0.54
1.4) The course provides an adequate overview of the subject matter treated.	not true		true	n=56	av.=4.55	md=5.00	dev.=0.60
1.5) The design of the course contributes to an understanding of the subject matter.	not true		true	n=55	av.=4.27	md=4.00	dev.=0.85

## 2. Course materials to assist Learning

2.1) There is overall enough material provided to assist the learning process (slides, course material, hand-outs, etc.).	not true		true	n=53	av.=4.30	md=4.00	dev.=0.85
2.2) The course materials (slides, course manuals, hand-outs, etc.) are overall of sufficient quality.	not true		true	n=55	av.=4.20	md=4.00	dev.=0.89

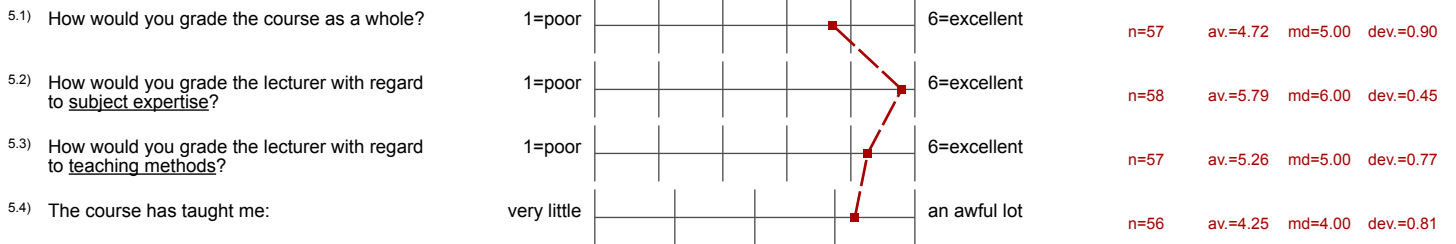
## 3. Commitment of the lecturer

3.1) The lecturer takes students seriously.	not true		true	n=50	av.=4.86	md=5.00	dev.=0.35
3.2) The lecturer is friendly and respectful towards students.	not true		true	n=52	av.=4.87	md=5.00	dev.=0.40
3.3) The lecturer addresses questions and suggestions from students adequately.	not true		true	n=51	av.=4.84	md=5.00	dev.=0.37
3.4) The lecturer seems to care about his/her students' learning progress.	not true		true	n=53	av.=4.64	md=5.00	dev.=0.59

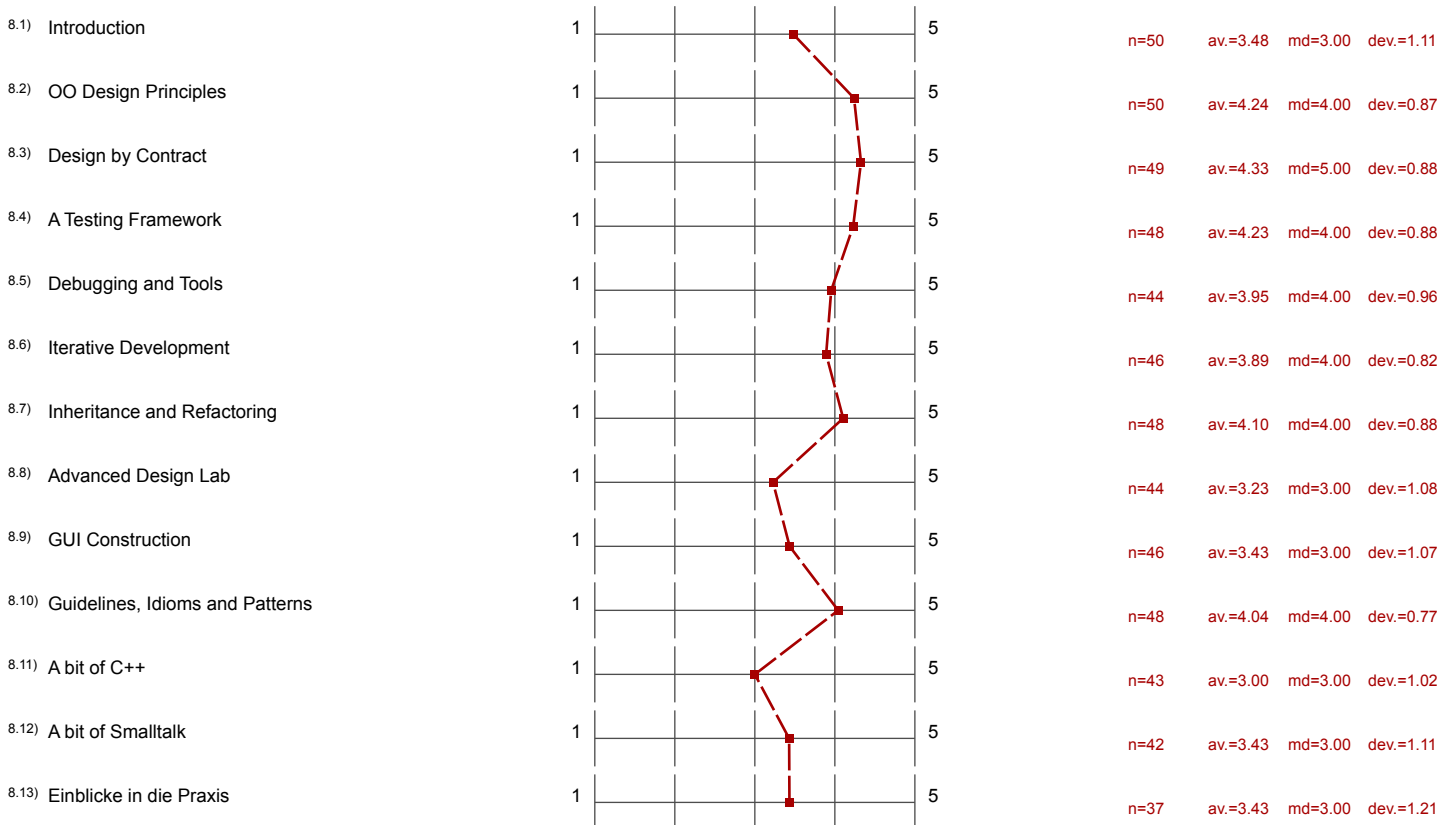
## 4. Complexity and Scope

4.1) The degree of difficulty of the course is:	too low/narrow		too high / wide	n=55	av.=3.62	md=4.00	dev.=0.78
4.2) The amount of content of the course is:	too low/narrow		too high / wide	n=55	av.=3.85	md=4.00	dev.=0.83
4.3) The pace of the course is:	too low/narrow		too high / wide	n=56	av.=3.48	md=3.00	dev.=0.66
4.4) The amount of knowledge presupposed by the course is:	too low/narrow		too high / wide	n=55	av.=3.47	md=3.00	dev.=0.81

## 5. Overall Assessment



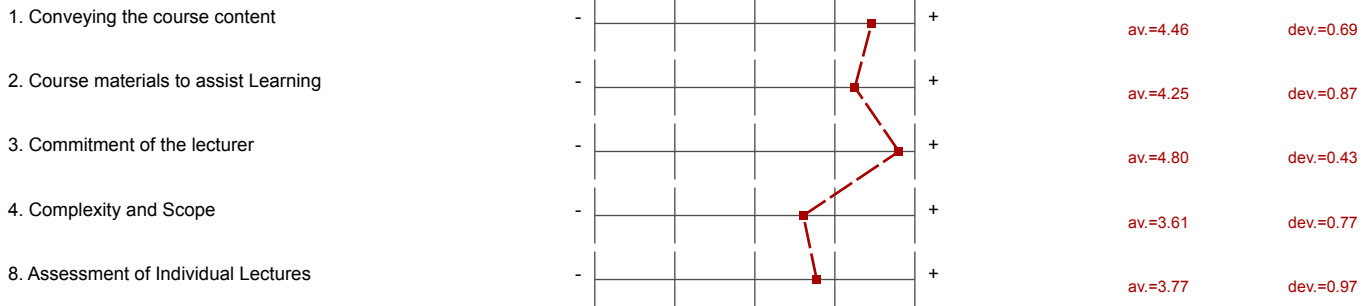
## 8. Assessment of Individual Lectures





# Profile Line for Indicators

Subunit: Phil.-nat. Fakultät  
Name of the instructor: Prof. Dr. Oscar Nierstrasz  
Name of the course: Programmierung 2  
(Name of the survey)



# Comments Report

## 7. Comments

7.1) What did you like about the course?

The lecturer was very motivated and friendly.  
We learned a bit of programming in PA and here we learned to design properly.

exercise feedback, good overview

Prof. expressed himself very clearly and gave lots of examples.

Lab presentation about different topics were quite useful, podcast is great, good explanation of topics, good that we had the pool session (much needed) with competent support, git and piazza are ok

Style of presentation, podcasts

The lecture was quite interesting and well organized

that there were podcasts available

The exercises allow for the students to train and use their knowledge acquired in the course

Good structure and very practical.

Would be good to have more fresh air in the room. Made sleepy...

I got an adequate overview on OO programming

Podcasts are available, slides too.

gute Ergänzung von Theorie mit Übungen  
Podcast waren grosse Hilfe!

Podcasts, Topics.

- Student assistants do special lab presentations
- Andrej and Pavla!
- Prof. DN is a great lecturer
- SW-Erkenntnisse in der Praxis

Very interesting, a lot of practical ex. (Code, demos etc.), the Prof is quite funny. The Assistant team is very nice & helpful.

practical experience ~~and~~ the multistep learning curve

The exercises are interesting and a good inspiration for self-learning.

It's interesting and it teaches some things that are really good to know for being efficient at work later.

good slides, very good & nice lecturer and assistants  
educative exercises

practical view ~~and~~ (look at topics from practical side) lot of examples

The very last lesson was very interesting and the way the professor interacted with students was great.

Learned a lot, interesting exercises

The instructor expressed himself clearly and there was a lot of practical application

It's very intense, but you come out feeling like you've made a lot of progress.

everything ~~else~~



The exercises were really building up skill in programming and the correction, revises etc. were done very well!

That we had podcasts, was very convenient.

- learned so much
- the GUI tutorial was fun and gave time to breathe for the tutorials

programming

Covered a wide range of topics  
Practical part, running examples

- ⊕ great presentations
- ⊕ hidden slides are helpful
- ⊕ feedback of exercises (very careful)

+ podcasts, lots of example codes, cool exercises

I really feel like I'm a better programmer now, I learnt a lot although it was tough.

The lecturer knew a lot about the subject and presented extremely well

Programming

Many exercises to practise coding.

The professor and lab instructors were very helpful and knew the materials that they presented.

The Programming

Therese

The course was very educational and Prof. Nierstrasz has a very good and understandable teaching style. The tutors were helpful. I also liked that the extension of the deadline was granted when needed. I also really appreciated the podcasts (I wasn't able to attend the course in person).

7.2) What did you not like about the course?

C++ and Smalltalk very briefly at the end... not worth it  
-> rather spend more time on implementing a GUI

The lab lecture slides were not always clear.

too big exercises for the last week, generally workload super high  
of exercises

Exercises were ~~too~~ too hectic. Too little time for too much work, ~~and~~ while the corrections came in weeks after submitting them.  
Hierarchical progress as we did not know what was next before

The exercises took way too much time especially towards the exam it would have been nice to lighten the workload a bit.

### Übungsbetrieb

The exercises come back very late  
There is no notification if one must correct exercises, thus there is a chance to miss corrections  
The ~~exercise~~ exam preparation was not well-designed

Exercises were quite messy in the beginning. Had to change partners a lot because of students who quit. As the pace and pressure with testat was high, this was very stressful.

I found the lecture was not following a clear structure. ~~But at some point~~  
Sometimes it felt like "here a bit of that & there a bit of that"

• In comparison to other courses too many hours spent on exercises.  
No advantages or such big exercises. No real feedback on them. (Just: ok, not good)

exercises take too much time

Testatbestätigung und Diskussion der Prüfungsinhalte  
kaum viel zu spät (3 Tage vor Prüfung...)

inconsequent Tutors.  
unclear Tutors. Pre and post conditions always changed.

The feedback on the exercises were not always helpful.

- Exercises take too much ~~time~~ time! Yes we do learn a lot, but it still takes too much effort for 5 ECTS!

The exercises are really a lot of work and very difficult

At the end it would be better to have a shorter exercise. The one with the Patterns needed much time that was needed for other exam preparations. It was generally good but included much work that did not help understanding Patterns

The exercises were quite tedious at times, and debugging could be explained better (the practical application in the IDE)

Group teams since difference in level is too high and ~~that~~ that exercises don't contribute to outcome. ~~that~~

The exercises were a bit too much sometimes

the topics in the first few lectures could be more well structured...

That C++ and Smalltalk felt like they were just thrown in there.

So much workload, especially at the end since feedbacks took too long

The handling of the exercises, corrections were returned very late and the feedback wasn't always helpful

You could maybe be a bit ~~more~~ stricter about the team work ~~of my partner~~, that both really contribute.

Patterns could be explained earlier, without so much pressure in the end  
(one at a time) Too many games.

GUI-lecture (2nd half) somewhat overly complicated. Too difficult exercises;

This course should be a Practicum the exercises are big enough and also everything you need. The things I studied for the exam had only in context to do with what I did throughout the year. And the exam was really pointless in ~~its~~ its content. Thanks for understanding!

Often it was difficult to understand the part, but he talked too fast.

The exercises were way too difficult, you often didn't had a chance to complete them without external help.

- crazy workload
- too much (java, c++, smalltalk).
- GUI code stuff was too much (and not used in tutorial)

~~the exam,~~ ~~&~~ no exam and only exercises!

Too much work, too much details to learn,

- ① → too much work for exercises
- exercise could be more focused

It was a waste of Time.

- management of the exercises/lab hours, would have liked more feedback on the exercises, too much effort needed for exercises

Sometimes, we repeated certain points too often (e.g. "long methods are bad" a.s.o.). Also sometimes the lecturer provides almost too much information (additionally to slides + comments)

Some exercises were tedious and simply took too much time.



Programming Ex doesn't count for the final grade  
The first exercises were hard (for beginners)

That how the exam is built, was only given on Friday  
before the exam!

That the exercise lessons were almost a new course lecture.

~~Test Exam~~ was pretty unnecessary → final project would be better

↗

exercises are hard ~~and a lot of time~~ not enough time  
(espec. if it is a Minor) if you're new to programming } jump from

zu grosse Aufgaben (v.a. wenn Gruppenarbeiten zu wenig beibringt)

7.3) Suggestions for improvements?

- smaller exercises
- slide could be better → had to google a lot

ask someone else to hold the lablecture on design patterns.

I did not understand any of it.

skip gui exercise and let them do the Database exercises earlier or  
postpone gui lecture to later.

~~Give~~ Give an mock Exam so the students knew exactly what to expect from the exam and can better prepare themselves.

~~Give~~ The exercise 5 (i.e. project) ~~should~~ should have taken 2 weeks at most not three weeks  
The GUI design seemed unnecessary

Vary teaching methods in exercise classes more. ~~Just oral~~ Just oral presentation is exhausting and hard to follow.  
Show examples maybe also ~~run~~ in a program live.  
a bit more structure

If you teach us about test driven development you should be able to write and test a small program for the test without bugs.  
↳ game.isValidState() was clearly not tested properly.

8 mind. 1.5 Wochen im November, damit dies entsprecheud in Vorbereitung einfließen kann

Have a straight line. Be clear and consequent.

Better organisation between assistants

- More ECTS or smaller exercises. It is not about the difficulty, it is about the amount!

Define in the beginning of the course the "testat bedingungen" like 8/10 EX must be passed.

The communication for the exam was poor. Tell us earlier and more precisely what is needed. If we have to learn details we need more time.

Exercises should influence the mark (for example 1/3) since exam is really theoretical and a lot of students weren't good in exercises or didn't do anything of high quality and still they will probably pass the exam. This is for me not a good representation of knowledge.

Maybe allow a cheat sheet with fragments of code at an exam. Some things aren't used very often, and using a development environment doesn't help measuring the most important things, are written and it would be a bit closer to real-world conditions.



Patric look into Design Pattern Exams

? maybe more exercises with design patterns since they are usefull but we barely use them.

Feedbacks need to come faster (sometimes we waited weeks)

Don't distract the students during the exam by making corrections (or do it quietly without confusing and taking exam time).

For already answered this above make ~~the~~ programming courses a practical an exam is pointless!

↳ please talk slower.

no exam only exercises

Distinguish clearly what ~~comes on~~ is relevant to the exam.

Do not put it to the mandatory courses for people with 5 years of experience in professional Software engineering.

- try to do more and correct feedback, also maybe a bit faster than 4 weeks, leave exercise 10 since no one has time, allow us to implement our GUI in ex 3. instead of the movieDB, ~~try to extend the exam preparation slides~~

Restructure the course. I knew it's hard as some topics are overlapping. And change the ECTS system in general. In Pi, I worked at least 12h/week (additionally to lectures), in DB maybe 2 - if get

2053






Better distribution of exercises

Avoid above.

This ✓

Have more continuity with the exercise! I would liked to make a GUI for the Quoridor Game

Maybe not quite as many exercises. It was a lot of work, especially considering that we (the students) often have five more lectures with assignments to hand in.

	-					+	Ø	dev.	
1. Conveying the course content	0%	20%	40%	60%	80%	100%	4.46	0.69	
									
	Scale width: 5								
2. Course materials to assist Learning	0%	20%	40%	60%	80%	100%	4.25	0.87	
									
	Scale width: 5								
3. Commitment of the lecturer	0%	20%	40%	60%	80%	100%	4.8	0.43	
									
	Scale width: 5								
4. Complexity and Scope Complexity and Scope: left pole=too low, right pole=too high; grade 3=exactly right	0%	20%	40%	60%	80%	100%	3.61	0.77	
									
	Scale width: 5								
8. Assessment of Individual Lectures	0%	20%	40%	60%	80%	100%	3.77	0.97	
									
	Scale width: 5								

dev.=Std. Dev.