

ID2203-2022

Course Analysis

Paris Carbone <parisc@kth.se>

1. Overview

In 2022 the course was actively re-developed to incorporate the needs that we identified in the 2021 version of the course. While the course objectives remained the same this year, most effort was put into making the course address and assess these objectives more reliably. To that end, the most important changes include 1) a remake of all lectures of the course and presentations with richer context, relating all concepts to real problems and systems, 2) A brand new full chapter on scalable data management which includes the important topic of distributed transactions (e.g., two-phase commit), as well as advanced topics in real-time, atomic time, and the future applications of quantum clocks and their properties. This also includes a closer examination of the Google Spanner framework which puts into perspective all advanced topics in the course such as Distributed Consensus, Atomic Clocks, Linearizable Registers, Snapshotting, and Atomic Commitment for Strict Serializability. 3) A wider range of project topics, programming languages of choice, and the choice of student-proposed projects, 4) New labs with rich content on practical consensus, liveness, and leader election as well as a deeper look into TLA+ and model checking. 5) A richer exam that despite being online provided a more accurate assessment through integrated grading criteria, multiple choices, and questions in little time which encouraged “proof of work” and discouraged students to seek answers elsewhere due to the time restrictions. In its current form, the ID2203 is a first-of-a-kind worldwide that adequately covers all these deep topics in a single course.

2. Course Development Updates

The core changes from previous years that led ID2203 to its current form are as follows:

New Lecture Method and Updated Presentations: Presentations (and slides) were re-made in order to drive the narrative in the form of a dialogue. Teaching by question and answer interactively is rooted in the Socratic method, named after the ancient Greek philosopher Socrates. This educational approach focuses on fostering active engagement from students through questioning and interactive dialogue rather than merely presenting information in a didactic manner. From the student reviews, we could assert that this method is effective in improving retention and understanding as well as active learning through peer interaction. In a course that is heavily rooted in abstractions that compose one another, it was obvious to us that

this method would aid the students in understanding and reasoning about the chained dependencies between implementations of algorithms and the properties they build upon.

New Chapter on Distributed Data Management: A crucial missing piece in the course was a chapter that could somehow put all the knowledge together to make sense to the students in an engineered fashion. To that end, we fully developed a full chapter material out of our experience building popular open-source systems and our interaction in the data management research community. The new chapter addressed in detail how modern scalable cloud databases and transactional query processing systems build upon all the principles learned in the course. This includes clock abstractions supported by the world's most advanced hardware (e.g., GPS system, atomic clocks, quantum-based error correction) as well as the most innovative principles put together (e.g., ACID database properties, strict serializability) to form cohesive protocols such as non-blocking two-phase commit that make data systems work seamlessly. The chapter also inspired many students to pick a systems research academic career and join our research lab in order to push the envelope further in this interesting field.

Project Changes: We decided to keep TLA+ as a project choice but also increased the number of available topics to ~10 while also allowing students to propose their own project. This made students more engaged to pick their favorite problem taught in this course and solve by using all the skills acquired in the course. The project was either done in Scala/Java or Rust by most students. Those who picked Rust had an extra incentive to contribute back code to new promising libraries such as OmniPaxos which made its debut in this course.

New Labs: With the great help and enthusiasm of TAs we developed new labs and guest lectures that triggered the interests of students and deepened their practical understanding of many of the concepts taught in the course such as distributed consensus protocols and their differences as well as pros/cons. PhD student Harald Ng took the lead in several of these labs, developing his teaching skills further while delivering content of exceptional quality and maturity.

Final Exam: The final exam this year kept its covid19 online-restriction. However, we increased the number of questions as well as the difficulty of the questions in order to incorporate more complex thinking and also assess how students could reason about their solutions. Forms of grading criteria were also introduced to encourage students to give equal weight to all course objectives and also to make the examination less complicated using the old point-based system.

Piazza Forum: We kept the Piazza forum to encourage students to post anonymously and help each other like in the previous version of the course. The only difference is that we did not pay the premium fee and used the community edition this time.

3. Course Evaluation Analysis

For reference, we attach the student feedback at the end of this report.

3.1. Analysis of Student Feedback

Students praised the teaching style, unique content, and valuable knowledge acquired in this course. They also identified that their extensive efforts paid off in the end. After the last lecture, a large group of students acknowledged that this was their best experience at KTH and they wished that other KTH courses were on par with ID2203. At the same time, they all agreed that the workload was rather high, whereas, some of them had to abandon other courses in order to keep up with the expectations in ID2203. Some commented that this course, based on the amount of time spent, should correspond to 9 or 12 credits.

3.2. Reflection on Student Feedback

It became evident to us that the changes were all well-perceived by the students this year. To that end, it was clear to us that all changes in the course development were successful. However, most of the criticism focused on the increased workload. Primarily that is due to the combination of heavy theory and new content for most students with many technologies that might be overwhelming such as Scala, Rust, and TLA+. The project was also announced late in the course. That was done deliberately in order to make sure that students have the right background which is the case after three-quarters of the content is presented. However, this left students with very little time (primarily during the examination period) to develop their final project. That was a problem we intended to solve in the upcoming years by either splitting this course into two courses or reducing content and introducing the project earlier in the term.

4. Grade Distribution

This year we did not see any significant skew in the grade distribution which followed a normal bell curve with mostly B and C grades.

5. Plans for next iteration

We concluded that several actions need to be taken to further improve the course:

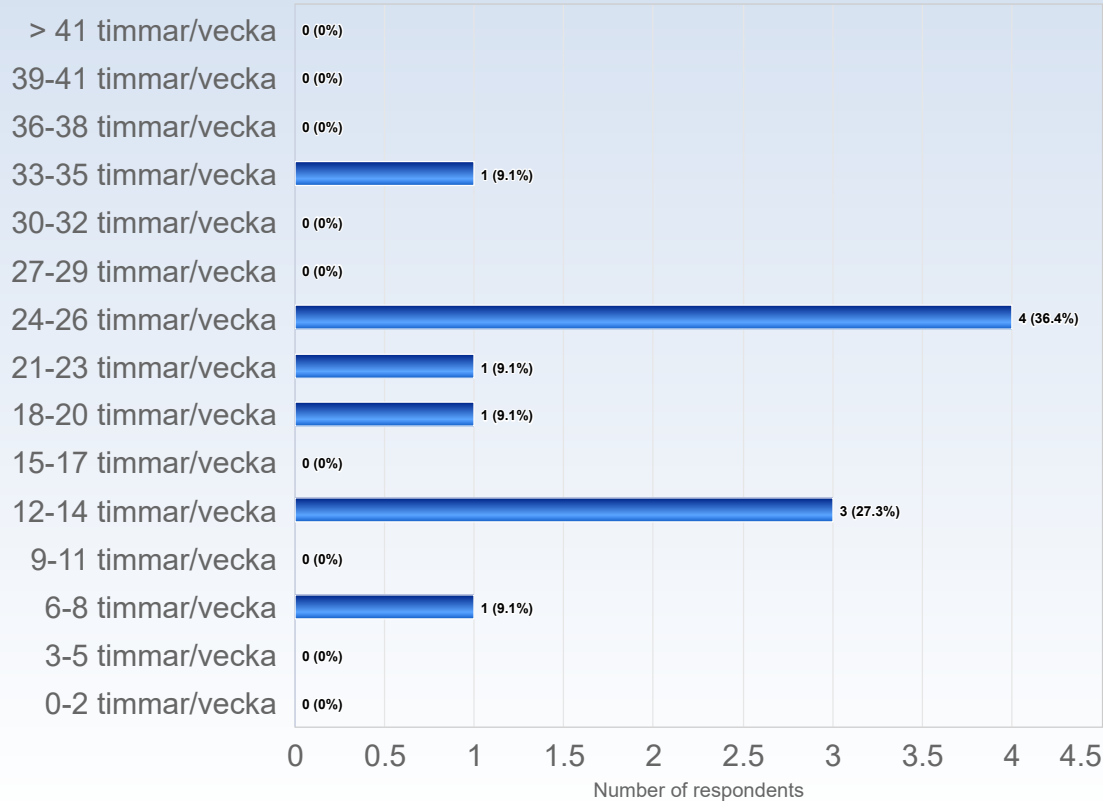
- Reduce the final project workload
- Make weekly programming assignments bonus/optional
- Allow larger groups of students to work together on the project
- Remove some of the theory in basic abstractions that is too trivial and add a chapter on eventual consistency and Conflict Free Replicated Data Types.

ID2203 - 2022-04-18

Antal responder: 39
Antal svar: 11
Svarsfrekvens: 28,21 %

ESTIMATED WORKLOAD

On average, how many hours/week did you work with the course (including scheduled hours)?



Comments

Comments (I worked: 6-8 timmar/vecka)

This course is pretty advance and demanding course in our degree program and it was quite enjoyable subject and it was thought perfectly by the instructors and TA's. The learning from this course has been a lot for me.

Comments (I worked: 18-20 timmar/vecka)

The content of the subject is massive. You learn a ton of theory but there's no time to assimilate it all. There is not much documentation about Kompics (something like StackOverflow), so the labs are sometimes frustrating. I did not have experience with rust, so I struggled a lot doing the last project. The topics are super interesting as the scalability of the systems tends to be now horizontal. The teacher is super good at teaching the topics, it is just that it is overwhelming.

Comments (I worked: 21-23 timmar/vecka)

Good amount of lecture time, could be more lectures and actually give us our full 15 minute breaks to increase attention and happiness

Comments (I worked: 24-26 timmar/vecka)

The workload was very very back-heavy with the project. Otherwise it was the exercises and studying for the quizzes that took most time. With the project I worked maybe 8 hours a day for 2 and a half weeks (maybe more)

Do the graded quizzes and programming assignments in time, start early with the project as it took a good while until progress was made. Collaboration with fellow students is very recommended.

Comments (I worked: 33-35 timmar/vecka)

Course work was overwhelming! I'd say the content corresponds to 9 credits rather than 7.5.

LEARNING EXPERIENCE

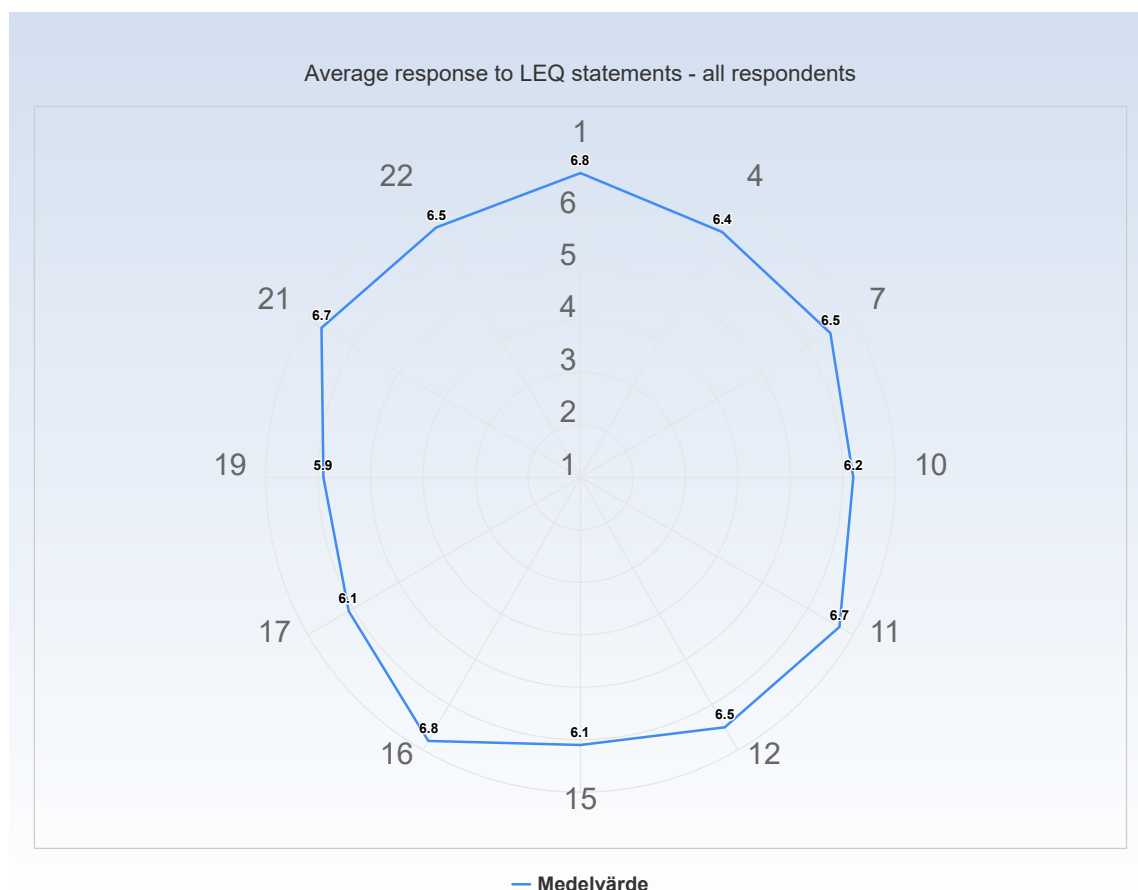
The polar diagrams below show the average response to the LEQ statements for different groups of respondents (only valid responses are included). The scale that is used in the diagrams is defined by:

1 = No, I strongly disagree with the statement

4 = I am neutral to the statement

7 = Yes, I strongly agree with the statement

Note! A group has to include at least 3 respondents in order to appear in a diagram.



KTH Learning Experience Questionnaire v3.1.4

Meaningfulness - emotional level

Stimulating tasks

1. I worked with interesting issues (a)

Exploration and own experience

2. I explored parts of the subject on my own (a)
3. I was able to learn by trying out my own ideas (b)

Challenge

4. The course was challenging in a stimulating way (c)

Belonging

5. I felt togetherness with others on the course (d)
6. The atmosphere on the course was open and inclusive (d)

Comprehensibility - cognitive level

Clear goals and organization

7. The intended learning outcomes helped me to understand what I was expected to achieve (e)
8. The course was organized in a way that supported my learning (e)

Understanding of subject matter

- 9. I understood what the teachers were talking about (f)
- 10. I was able to learn from concrete examples that I could relate to (g)
- 11. Understanding of key concepts had high priority (h)

Constructive alignment

- 12. The course activities helped me to achieve the intended learning outcomes efficiently (i)
- 13. I understood what I was expected to learn in order to obtain a certain grade (i)

Feedback and security

- 14. I received regular feedback that helped me to see my progress (j)
- 15. I could practice and receive feedback without being graded (j)
- 16. The assessment on the course was fair and honest (k)

Manageability - instrumental level

Sufficient background knowledge

- 17. My background knowledge was sufficient to follow the course (f)

Time to reflect

- 18. I regularly spent time to reflect on what I learned (l)

Variation and participation

- 19. The course activities enabled me to learn in different ways (m)
- 20. I had opportunities to influence the course activities (m)

Collaboration

- 21. I was able to learn by collaborating and discussing with others (n)

Support

22. I was able to get support if I needed it (c)

Learning factors from the literature that LEQ intends to examine

We tend to learn most effectively (in ways that make a sustained, substantial, and positive influence on the way we think, reflect, act or feel) when:

- a) We are trying to answer questions, solve problems or acquire skills that we find interesting, exciting or important
- b) We are able to speculate, test ideas (intellectually or practically) and learn from experience, even before we know much about the subject
- c) We are able to do so in a challenging and at the same time supportive environment
- d) We feel that we are part of a community and believe that other people have confidence in our ability to learn
- e) We understand the meaning of the intended learning outcomes, how the environment is organized, and what is expected of us
- f) We have adequate prior knowledge to deal with the current learning situation
- g) We are able to learn inductively by moving from concrete examples and experiences to general principles, rather than the reverse
- h) We are challenged to develop a true understanding of key concepts and gradually create a coherent whole from the content
- i) We believe that the work we are expected to do will help us to achieve the intended learning outcomes
- j) We are able to try, fail, and receive feedback before, and separate from, each summative assessment of our efforts

k) We believe that our work will be considered in an honest and fair way

l) We have sufficient time for learning and devote the time needed to do so

m) We believe that we have control over our own learning, and not that we are being manipulated

n) We are able to collaborate with other learners struggling with the same problems

Literature

Bain, K. (2004). *What the Best College Teachers Do*, Chapter 5, pp. 98-134. Cambridge: Harvard University Press.

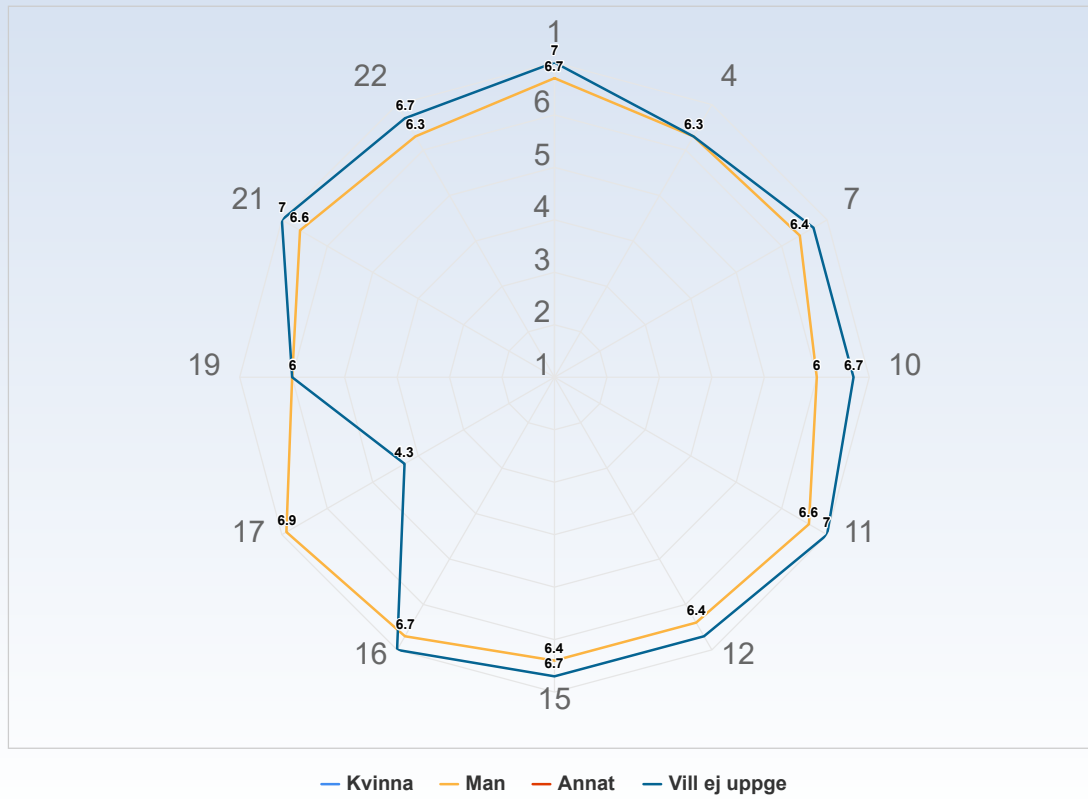
Biggs J. & Tang, C. (2011). *Teaching for Quality Learning at University*, Chapter 6, pp. 95-110. Maidenhead: McGraw Hill.

Elmgren, M. & Henriksson, A-S. (2014). *Academic Teaching*, Chapter 3, pp. 57-72. Lund: Studentlitteratur.

Kember, K. & McNaught, C. (2007). *Enhancing University Teaching: Lessons from Research into Award-Winning Teachers*, Chapter 5, pp. 31-40. Abingdon: Routledge.

Ramsden, P. (2003). *Learning to Teach in Higher Education*, Chapter 6, pp. 84-105. New York: RoutledgeFalmer.

Average response to LEQ statements - per gender

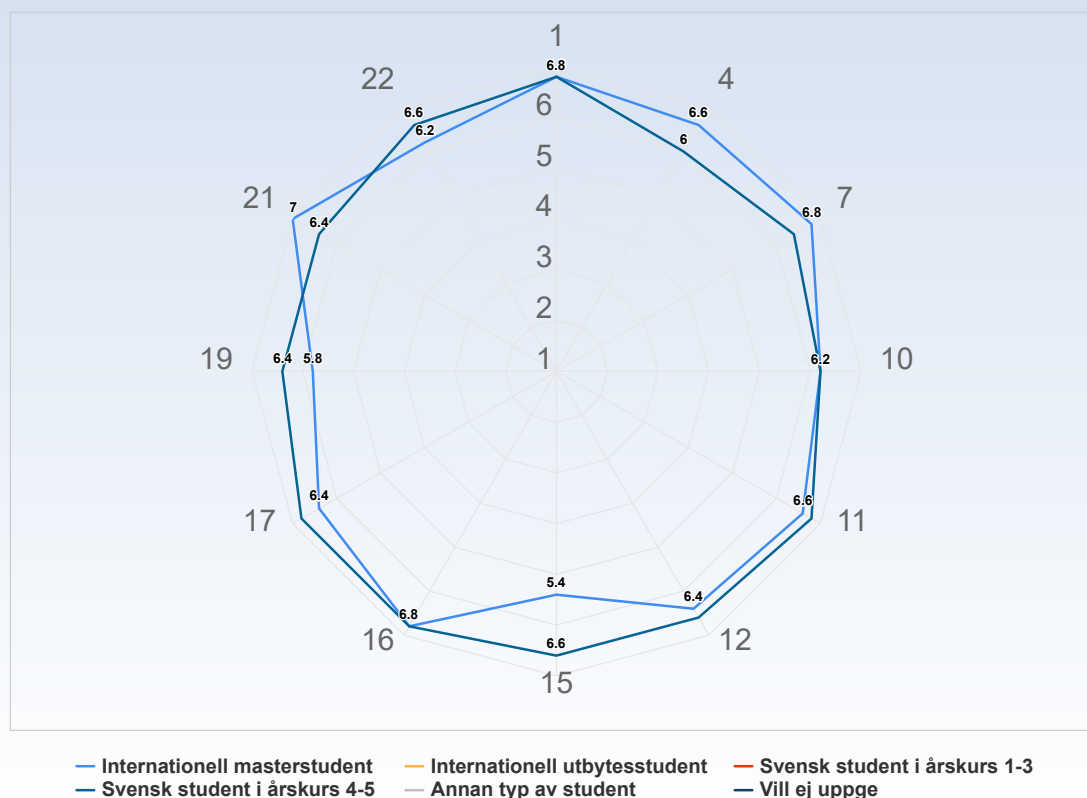


Comments

Comments (I am: Man)

NA

Average response to LEQ statements - per type of student



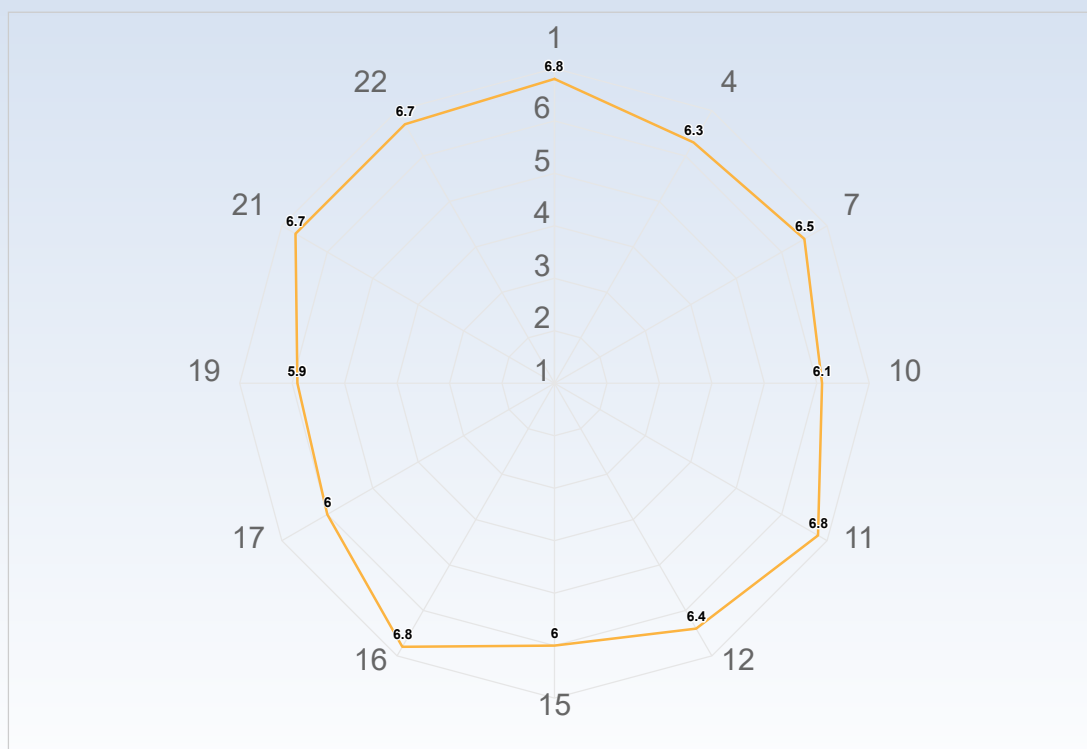
Comments

Comments (I am: Internationell masterstudent)

Master Student from South Asia

It is not acceptable that a project submission date for a P3 course goes into P4, especially when the project is not introduced earlier in the class.

Average response to LEQ statements - per disability



— Ja — Nej — Vill ej uppge

GENERAL QUESTIONS

What was the best aspect of the course?

What was the best aspect of the course? (I worked: 6-8 timmar/vecka)

The project for this course was the best part as we were given different range of interesting topics, like contributing to distributed open source application, creating our own using a library built by KTH Students. I personally enjoyed the challenges of the project and it was fun to work with it.

What was the best aspect of the course? (I worked: 12-14 timmar/vecka)

the content itself is interesting and really important

Well presented lectures. Rarely need to look in the course book.

What was the best aspect of the course? (I worked: 18-20 timmar/vecka)

Paris and Harald. Despite being a lot of topics, you guys teach them super well. But as I mentioned before, there were a ton of topics.

What was the best aspect of the course? (I worked: 21-23 timmar/vecka)

Paris's lectures were truly amazing and should to be honest set the standard for all lectures at KTH. Easy to follow, followed great structure throughout the entire course, had great examples, were explained in a way that was motivating and easy to understand and I could go on. Beats any other course I have taken by a mile. Thank you Paris!

What was the best aspect of the course? (I worked: 24-26 timmar/vecka)

I like that it's a very hands-on course. I really enjoyed the concepts in the course. The labs are actually quite fun and help with understanding the algorithms well. Only the project caused issues.

Paris is an amazing lecturer. I learnt more from him under this course than I ever did from the basic distro course. His engaging lecturing really activated me, but his quizzes very super good cause they forced you to learn stuff during the course.

Quizzes helped with core concepts, programming assignments was fun to do

What was the best aspect of the course? (I worked: 33-35 timmar/vecka)

The design of the course was good. Also the cutting edge focus in the course. Very nice teachers. Thank you.

What would you suggest to improve?

What would you suggest to improve? (I worked: 6-8 timmar/vecka)

This course introduces three different programming languages, like different for programming assignments, different for project and different for testing, it will be nice if this is improved to have only one consistent language like RUST for all use cases in the course.

What would you suggest to improve? (I worked: 12-14 timmar/vecka)

some lectures for example the one about distributed shared memory are impossible to go through in 2 hours. it often felt like a rush through everything.

The quizzes and the exam had "trick questions", e.g. "Select the examples that do NOT fit the description"

Please remove these questions. They do not test you on the course content as much as they test if you can keep two things in mind simultaneously.

What would you suggest to improve? (I worked: 18-20 timmar/vecka)

- Maybe the final exam could be split during the period into 2 or 3.
- This course could be a full semester long due to the amount of material.

What would you suggest to improve? (I worked: 21-23 timmar/vecka)

Some information surrounding the project. Taking on a project in a language that I have never touched before(Rust), implementing an API that I have never touched(OmniPaxos) while replacing an API that I have never touched(Raft) in a sort of limited time was an immense task. To have some good delimitation goals clearly stated would help set expectations for what we should be able to do. This also goes for other project choices like KV-stores etc. To know if we should implement a fully working KV-store that can be deployed on the market or something that demonstrates some level of understanding are both very different levels, and could both be expected in similar courses at KTH.

The points assigned for the project could also use some clarification. I totally get that it is hard to make these points clear while having such an open project with many choices, but maybe do a couple of small examples for point assignments?

Ex:

KV-store implementing reading and writing, basic user testing on 3 nodes - 10 points implementation, 10 points testing.

KV-store implementing reading and writing, crash recovery (and more etc), basic user testing on 3 nodes - 15 points implementation, 10 points testing.

KV-store implementing reading and writing, crash recovery (and more etc), automated testing using std-in N nodes - 15 points implementation, 20 points testing.

And so on. You could also easily state that these are not set in stone and could change depending on what people have done, but that the examples are provided to set expectations and to get students to worry less and implement and work with the topics covered more.

What would you suggest to improve? (I worked: 24-26 timmar/vecka)

1. I wish more time was spent on the more advanced topics rather than almost half the course going over the basics of DS again.
2. The project needs to be introduced much, much earlier. It is simply unacceptable that it is introduced so late and then submission is expected to happen after the start of a new period.
3. The quizzes and assignments keep going till a few days before the exam which left no room for us to start the project before the exam.

The courseload is not much but the reasons given for holding back on introducing the projects simply cause a lot of issues for students later in the period. It is too much at once. Students should be able to plan on their own when they can work on a project and holding back on it till near the end of the period is really bad scheduling in this course.

The project was hard in ways that differed from other hard kth courses.

1. Rust is weird and pretty hard. Has a hard learning curve, weird concepts and as students were under time pressure.

2. The API made sense after awhile, but it was still woefully under-documented with few concrete examples of you didn't specifically do a KV-store. The API assumed knowledge in Rust and the distributed concepts, of which we might have with the concepts, but not Rust. It also assumed that we understood networking which I felt might've been an entire course in itself. Tokyo is paaaaain to work with, and the standard library was even worse!

In summary, I spent maybe 90% of my time with learning asynch Rust syntax, concepts and threading, but rather little about actual distributed concepts :/ I swapped projects many times just to try find something that I could actually start on.

Let student do the assignments and project in the same language, some time had to be spent understanding the two languages used.

What would you suggest to improve? (I worked: 33-35 timmar/vecka)

help sessions for the project and maybe slightly reduce the course work load.

What advice would you like to give to future participants?

What advice would you like to give to future participants? (I worked: 6-8 timmar/vecka)

Don't miss the lecture and ask questions, the instructors of this course are super nice and they know their stuff.

What advice would you like to give to future participants? (I worked: 12-14 timmar/vecka)

attend all the lectures

Make sure that you fully understand the current concept you are studying before you move on to the next one. I.e. no stone left unturned.

This applies to everything.

What advice would you like to give to future participants? (I worked: 18-20 timmar/vecka)

Take the basic course, start doing the final project asap, get some friends, learn rust, get frustrated but overcome quickly.

What advice would you like to give to future participants? (I worked: 21-23 timmar/vecka)

Dont worry too much about the project grading. Just start working and get something that works going!

Communicate a lot using piazza and interact during lectures

What advice would you like to give to future participants? (I worked: 24-26 timmar/vecka)

Make a KV store in Rust, take the easy way out. Go to every single lecture cause Paris is amazing and you might not have another lecturer like him.

Do the quizzes after the lectures, start early with project

What advice would you like to give to future participants? (I worked: 33-35 timmar/vecka)

Study from day 1!

Is there anything else you would like to add?

Is there anything else you would like to add? (I worked: 12-14 timmar/vecka)

Great course, I liked it ☺

Is there anything else you would like to add? (I worked: 18-20 timmar/vecka)

Thank you. I learned a lot.

Is there anything else you would like to add? (I worked: 21-23 timmar/vecka)

Once again thank you all for a enjoyable course. Especially a big thank you to Paris! We didn't see you a lot during the final time of the course as lectures ended and didn't see you a lot in piazza as questions were mostly related to the project, but sincerely thank you! You're lectures are truly truly a pleasure to take part in. I wish you all the best!

Is there anything else you would like to add? (I worked: 24-26 timmar/vecka)

Please release the project topics sooner. Even if material is not yet covered in lectures, students can at least take it upon themselves to learn ahead if they wish to. Preventing someone from doing that simply to hold back on the project is not good from a student perspective as it causes a lot of load later in the period.

The small nongraded quizzes you could do were invaluable when learning the concepts. All of Paris lectures had those, and I felt I understood double or triple as much by just doing those. Maybe add them in the other modules too?

Intresting studies done by the Phd students

Is there anything else you would like to add? (I worked: 33-35 timmar/vecka)

Cheers!

SPECIFIC QUESTIONS

RESPONSE DATA

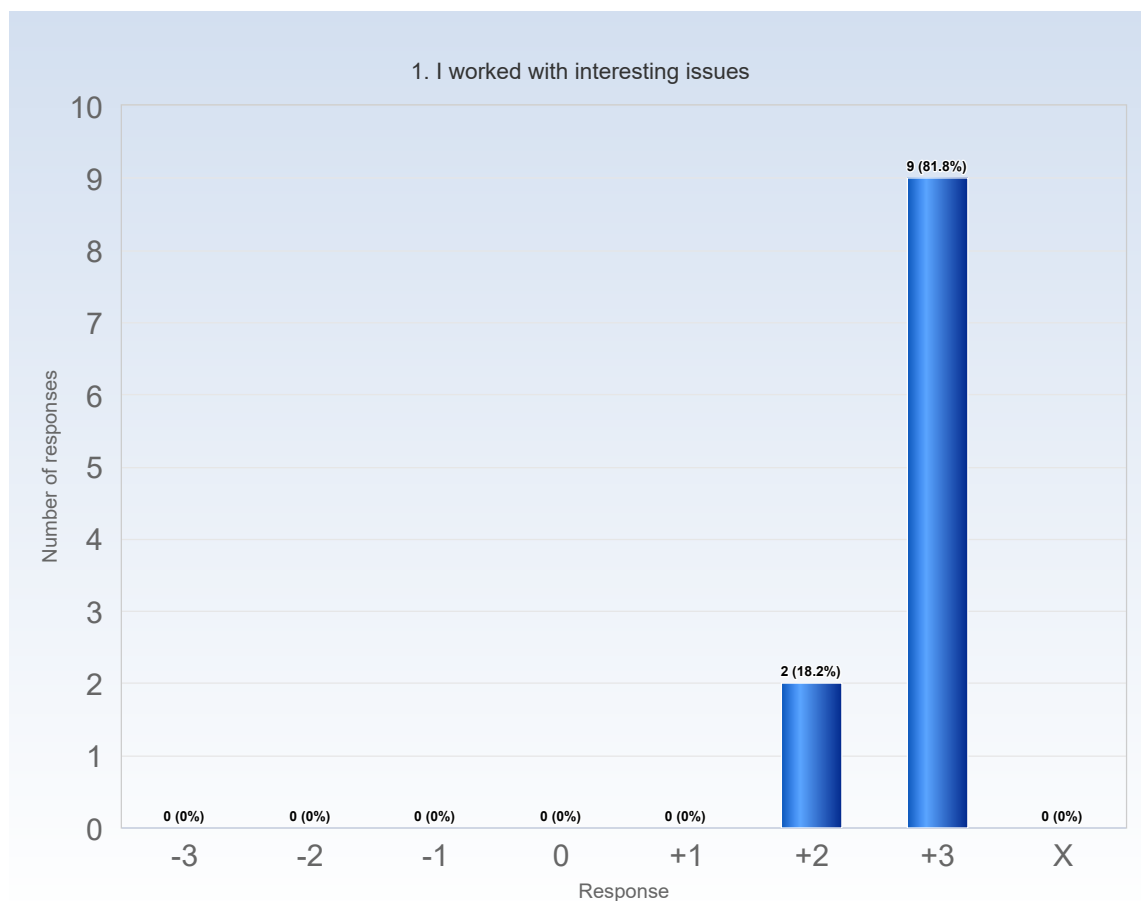
The diagrams below show the detailed response to the LEQ statements.
The response scale is defined by:

-3 = No, I strongly disagree with the statement

0 = I am neutral to the statement

+3 = Yes, I strongly agree with the statement

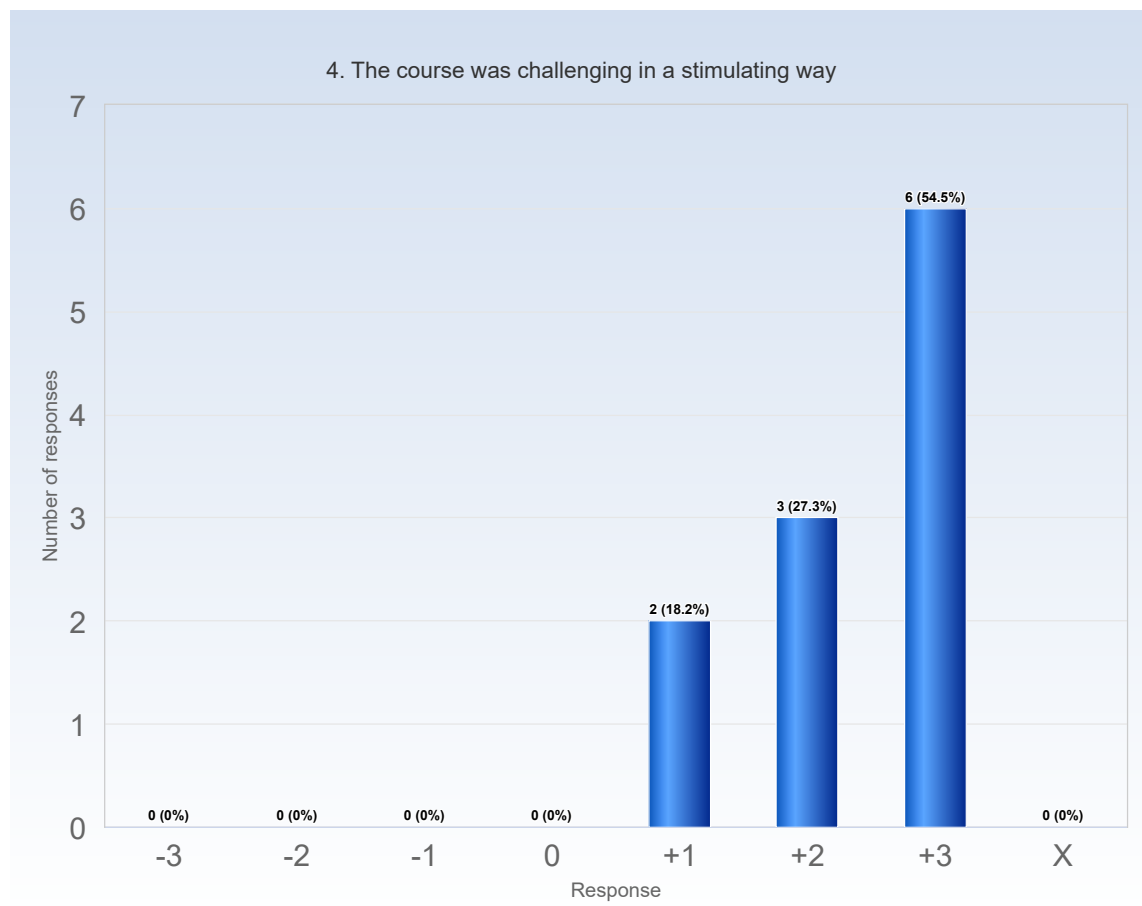
X = I decline to take a position on the statement



Comments

Comments (My response was: +3)

Yes especially the project



Comments

Comments (My response was: +1)

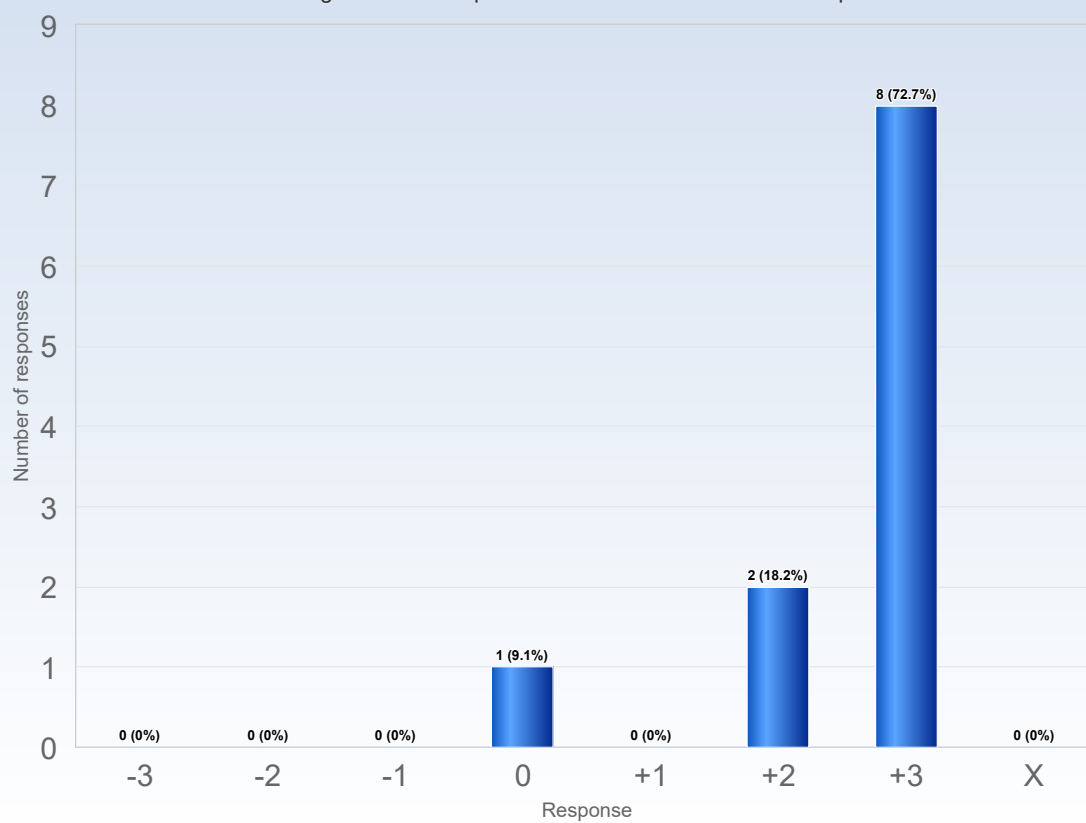
The quizzes, exercises and theoretical lectures were very good in challenging in a stimulating way. But the project challenged me less in my knowledge in distributed systems and way more in how good I was at Rust

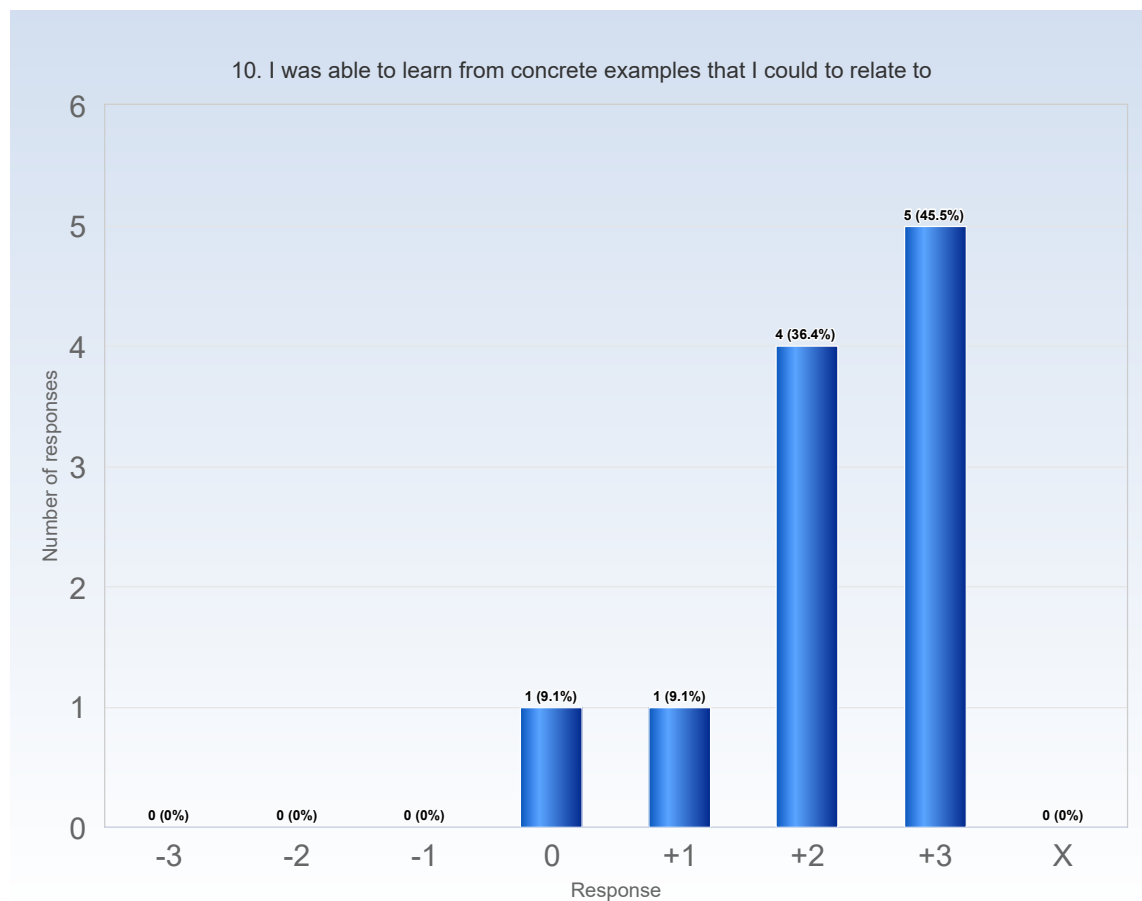
Comments (My response was: +2)

Yes, lectures were interesting as well as the exercises and project. Some concepts were very hard which transformed the stimulation to frustration (hehe). Also, the project was most challenging.

not satisfying challenge was lack of information and structure of project

7. The intended learning outcomes helped me to understand what I was expected to achieve

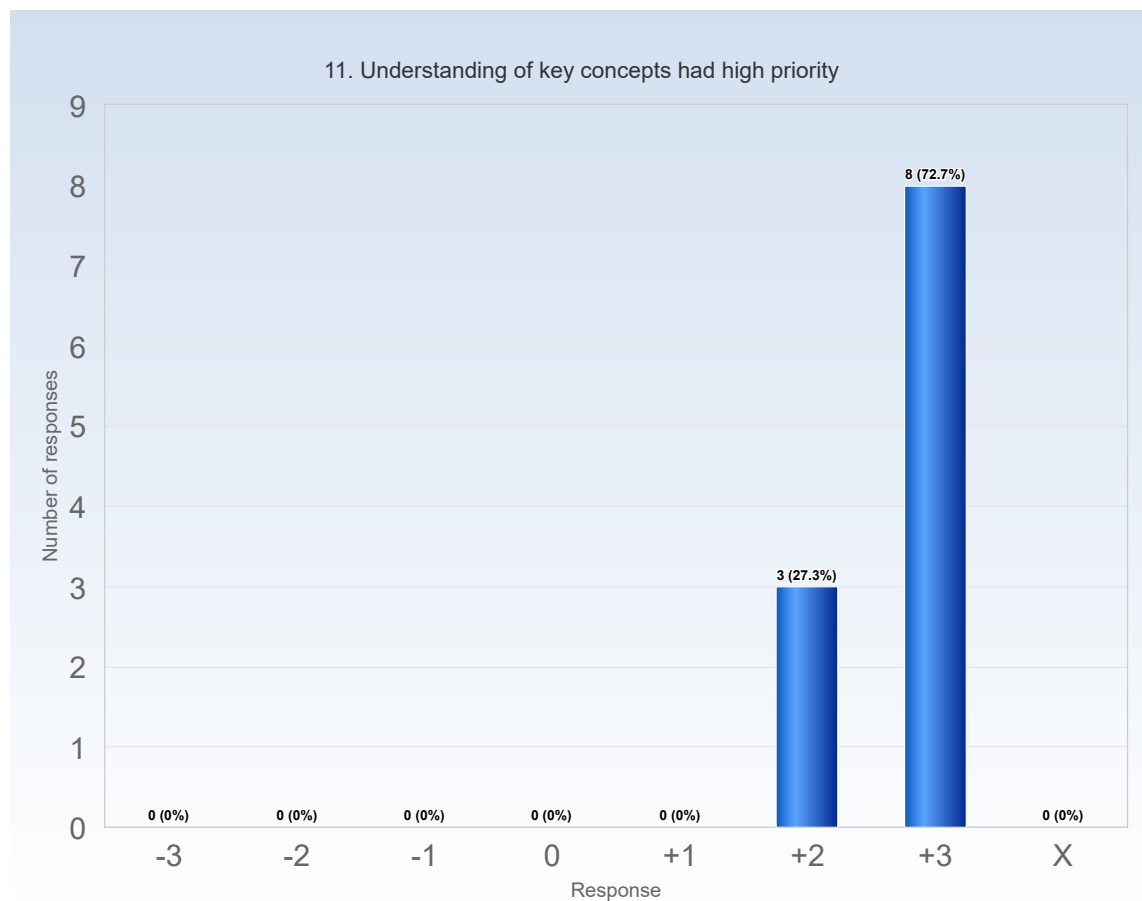




Comments

Comments (My response was: +1)
Yeah providing real worls examples of how the different systems and algos are used. Keep doing this

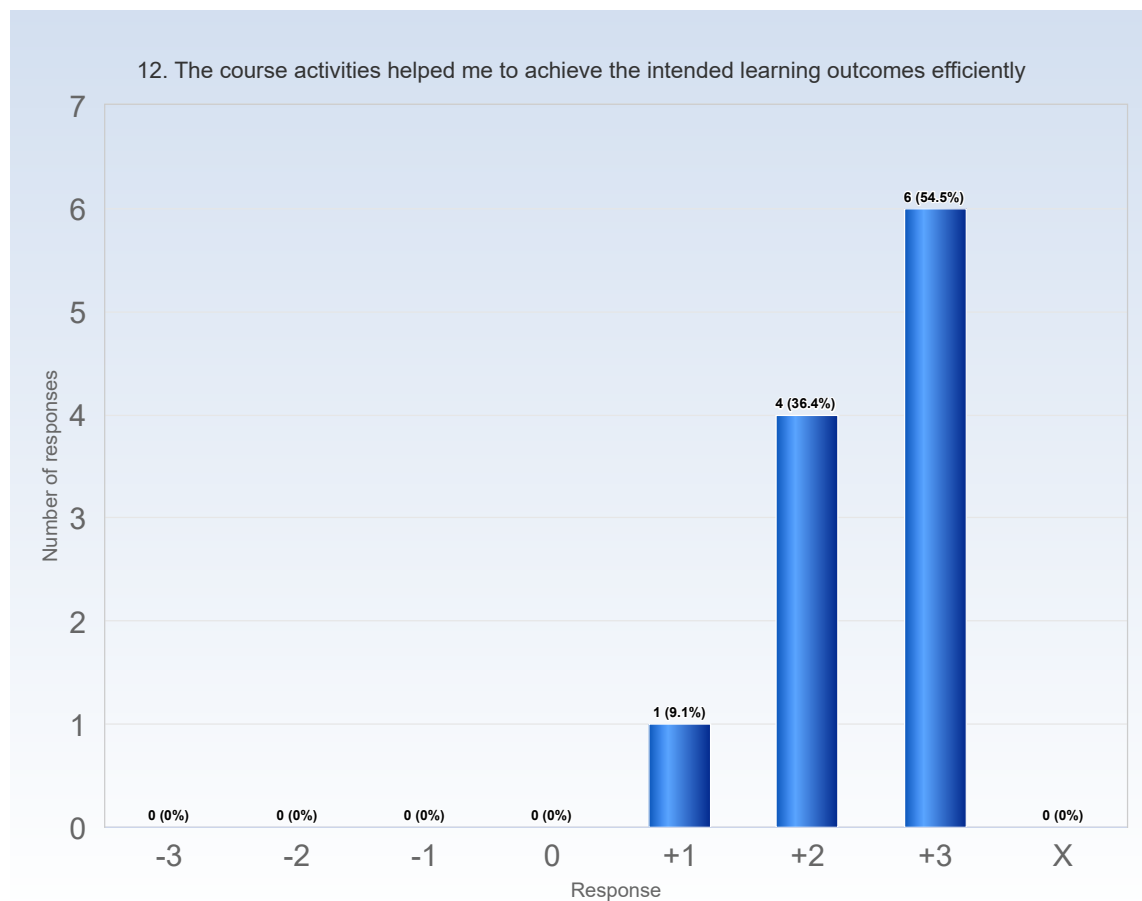
Comments (My response was: +2)
some examples were very abstract but you cant really compare everything to metaphores that are easy to understand



Comments

Comments (My response was: +3)

Understanding key concepts was the key of succeeding in the course

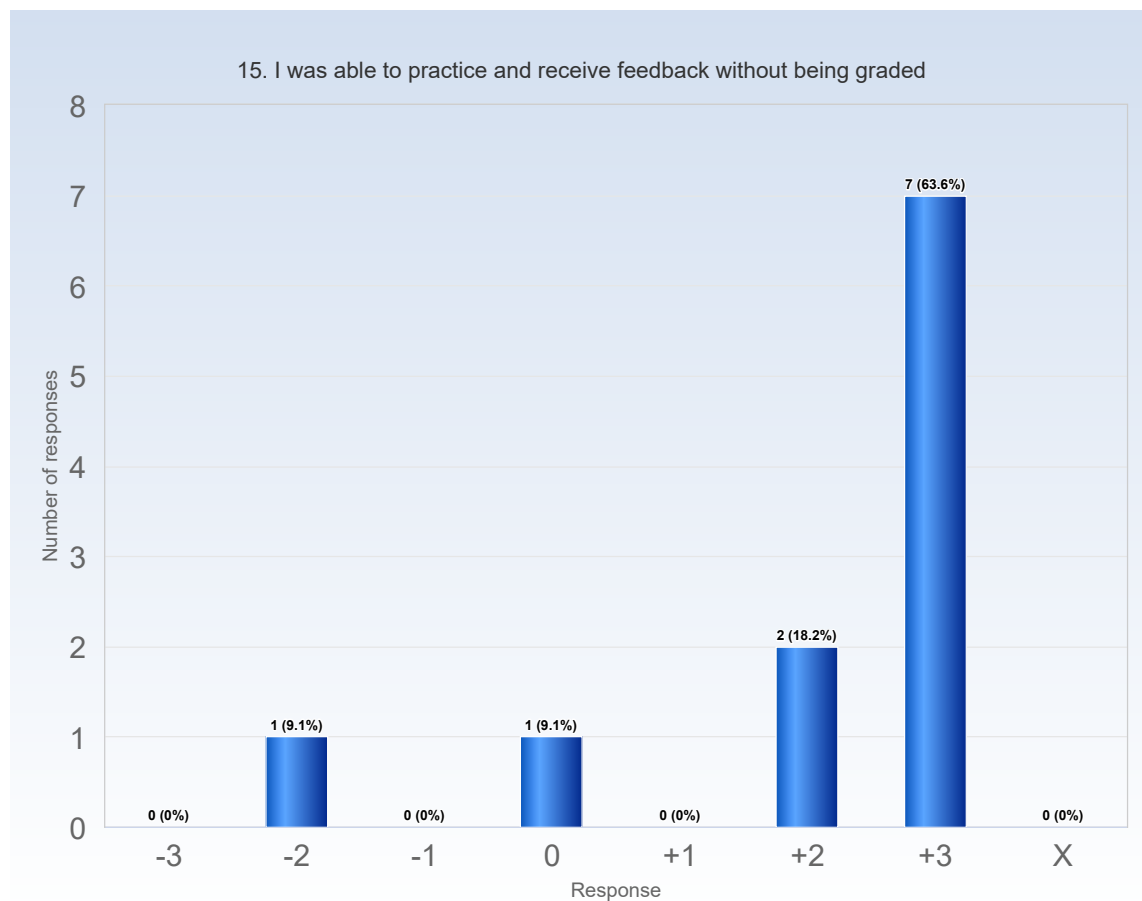


Comments

Comments (My response was: +2)

Quizzes drags this score down by one. IMO they were not very rewarding for my overall knowledge in the course

I think the programming assignments were a bit too much of a black box to understand why the code was working.



Comments

Comments (My response was: -2)

if by practice we understand coding, then all the work has been graded and there were no lab sessions for really practice

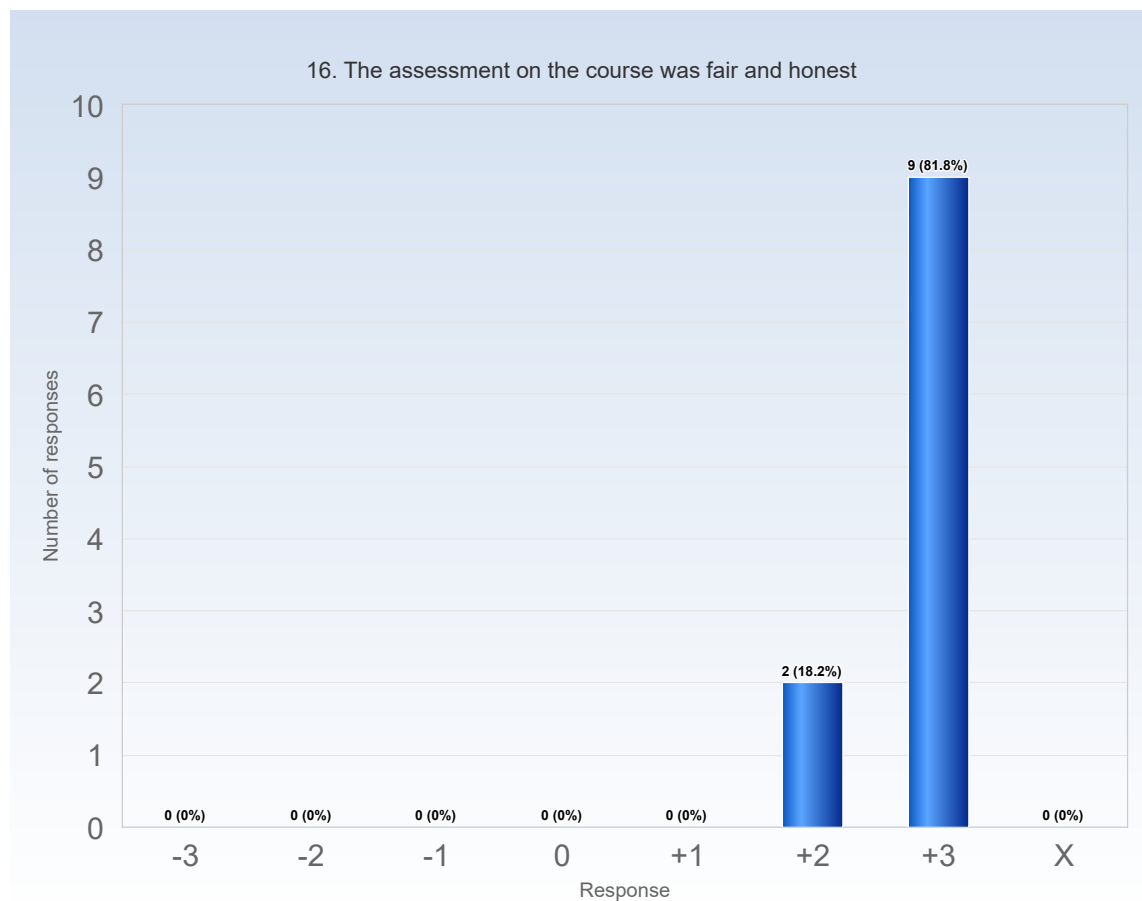
Comments (My response was: +2)

If I didn't get the second chance on the quizzes, it wouldn't have ended well. Some questions you "think" you know what they're talking about, and not until you're wrong and can get challenged on your knowledge do you realize where it's lacking. Very typical example was with the linearization points. I reaaaaally thought I understood that, but then the quiz showed me where I was wrong and I could then properly relearn it.

Comments (My response was: +3)

Yes! Very Active TA's and course lead. Nice!

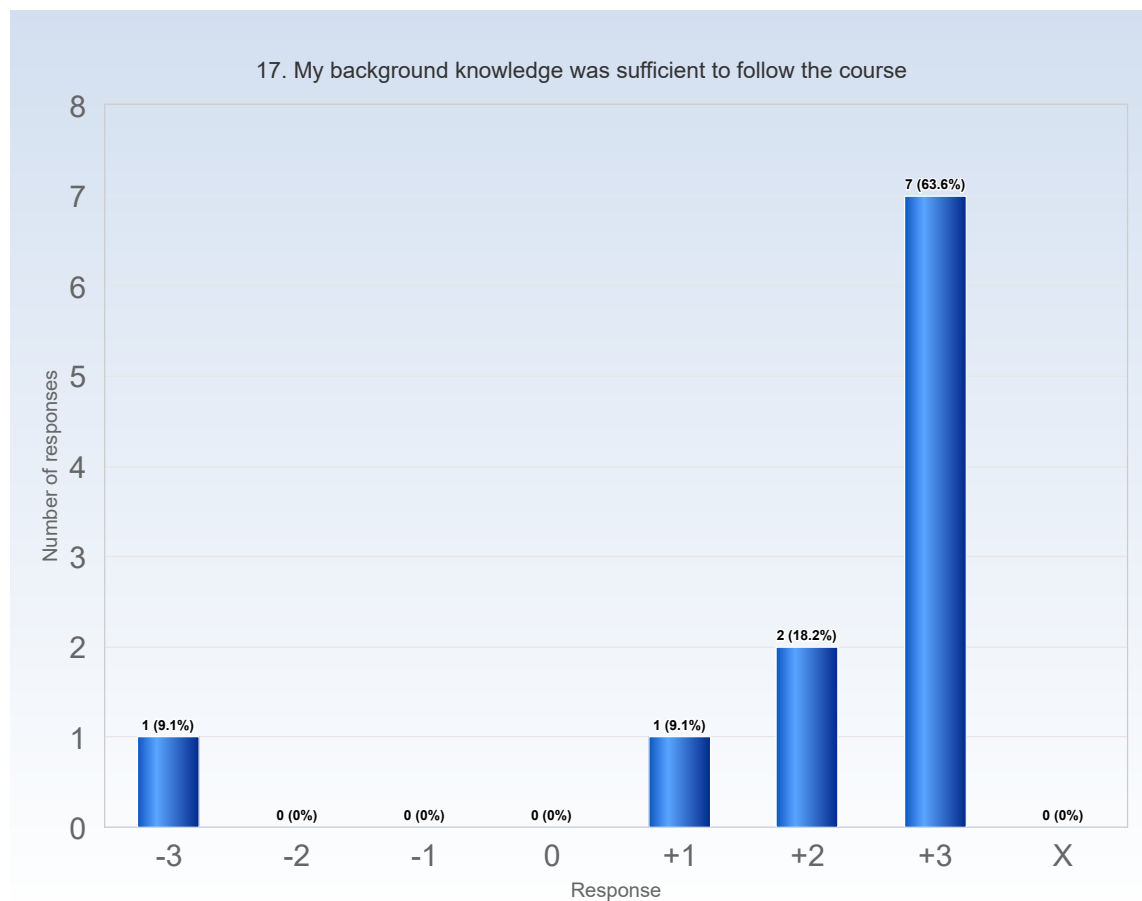
piazza is great for this! <3



Comments

Comments (My response was: +3)

yes, despite much worry related to the project, it was graded way fairer than I had anxiety for



Comments

Comments (My response was: -3)

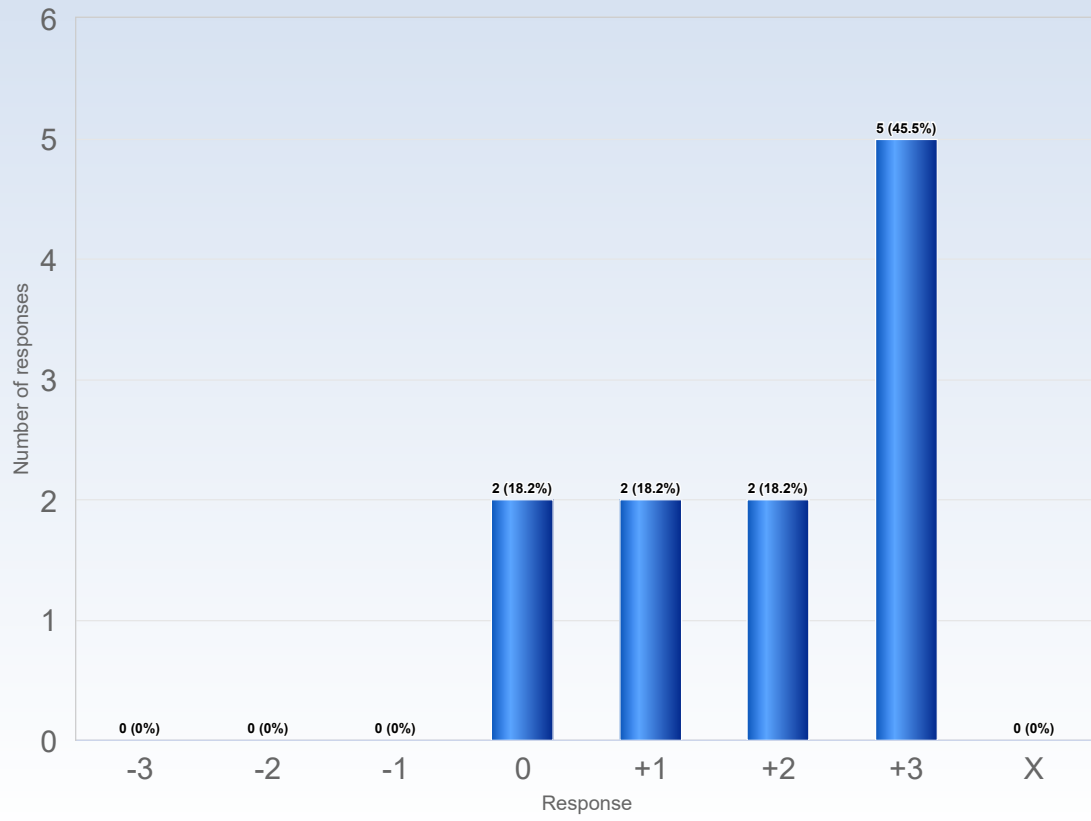
It should be mandatory to have taken the basic course.

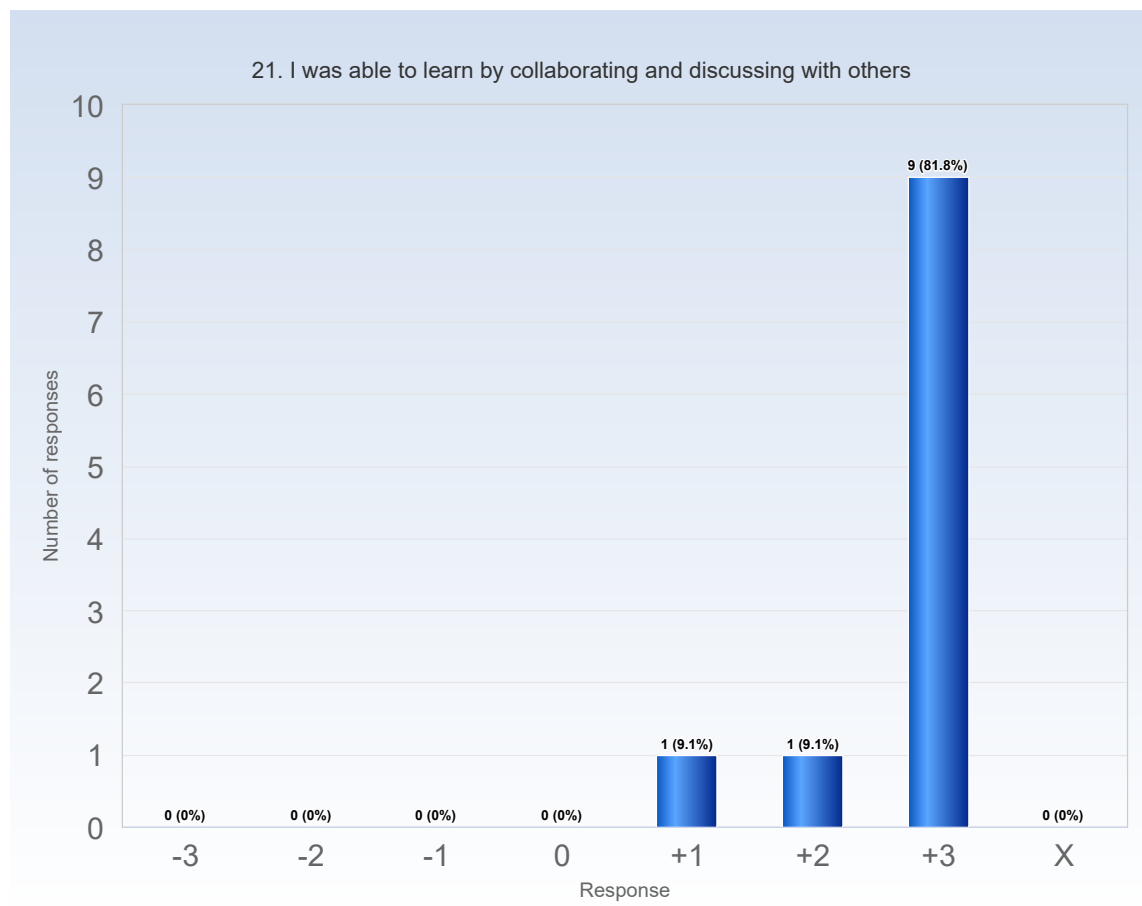
Comments (My response was: +3)

Basic course was preparing for this course very well.

I don't think you need any background in the area to understand the course.

19. The course activities enabled me to learn in different ways





Comments

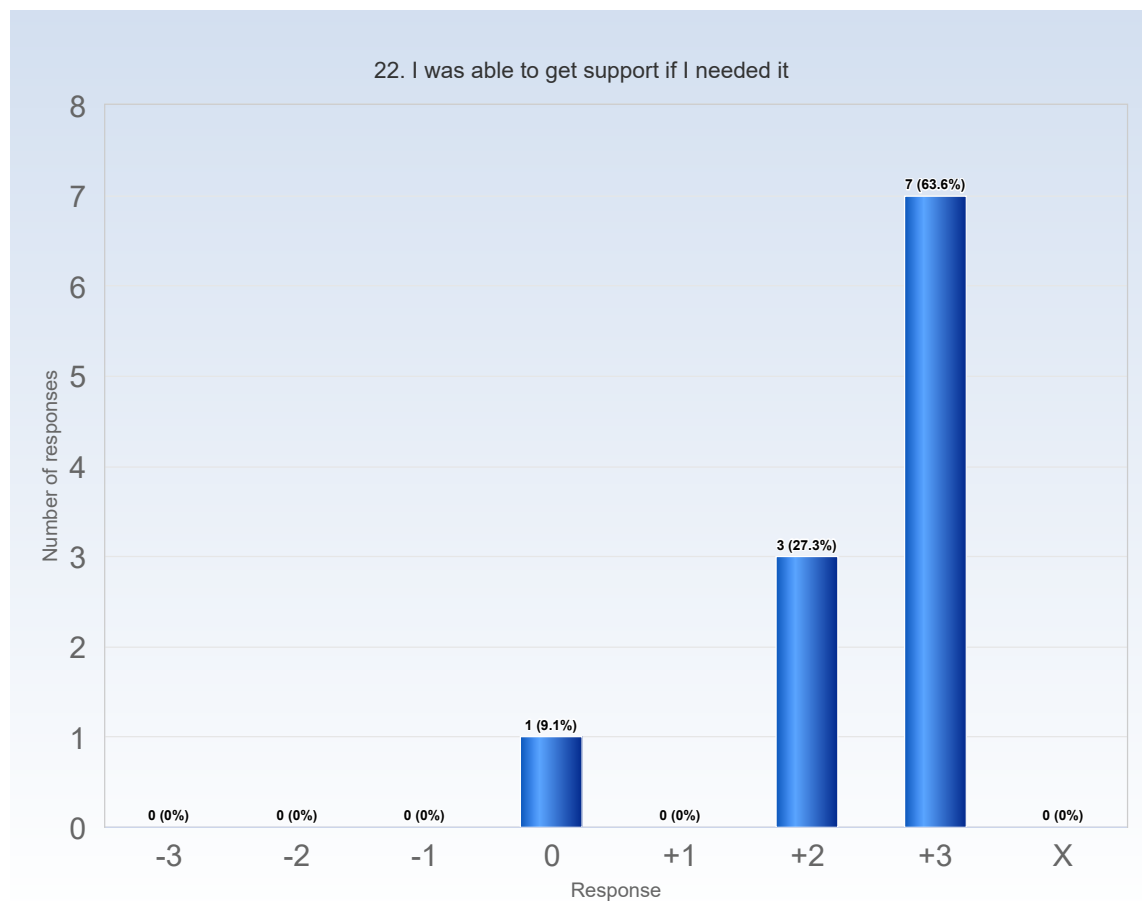
Comments (My response was: +1)

I didn't collaborate much with others.

Comments (My response was: +3)

Yea. Project was not feasible individually. I'd suggest you to offer help sessions for the project.

In the final project, despite being alone, a good collaboration was key.
especially for the project



Comments

Comments (My response was: +2)

Rust is such a different and weird language. It would be cool with maybe one lab or exercise to just let us properly practice with guidance.
Paris was great at answering questions and giving help

Comments (My response was: +3)

piazza once again was managed great
questions on piazza