



myExperience Report

Term 1, 2023

Faculty: Faculty of Engineering

School: School of Computer Sci & Eng

Course: COMP6991 Modern Prog Problems with Rust

Evaluation period: Apr 11 2023 12:00AM - Apr 27 2023 12:00AM

Course Report

Response Data

Raters	Student
Responded	80
Invited	201
Response Ratio	39.8%

Comparison of results for "Overall I was satisfied with the quality of the course"

This course: COMP6991 Modern Prog Problems with Rust

Overall I was satisfied with the quality of the course				
Options	Count	Percentage	Statistics	Value
Strongly disagree	0	0.0%	Mean	5.61
Disagree	0	0.0%	Median	6.00
Moderately disagree	2	2.5%	Standard Deviation	0.68
Moderately agree	3	3.8%	Standard Error (base on SD)	0.08
Agree	19	23.8%	% Agree broad	97.5%
Strongly agree	56	70.0%		

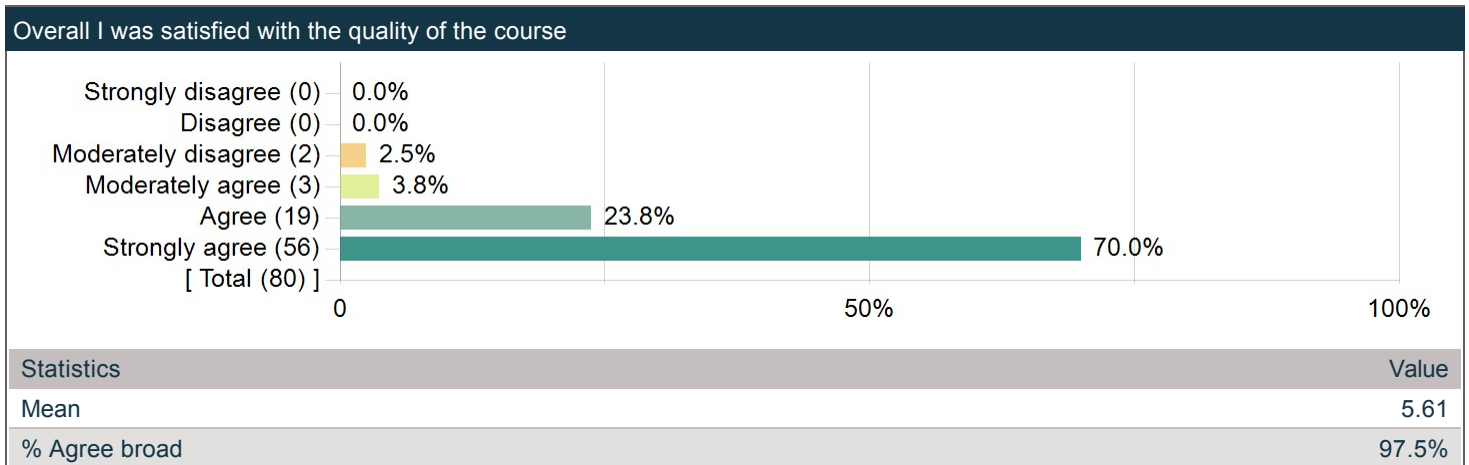
SCHOOL: School of Computer Sci & Eng

Overall I was satisfied with the quality of the course			
Options	Percentage	Statistics	Value
Strongly disagree	2.7%	Mean	4.90
Disagree	3.0%	Median	5.00
Moderately disagree	4.9%	Standard Deviation	1.19
Moderately agree	17.1%	Standard Error (base on SD)	0.01
Agree	35.6%	% Agree broad	89.4%
Strongly agree	36.7%		

FACULTY: Faculty of Engineering

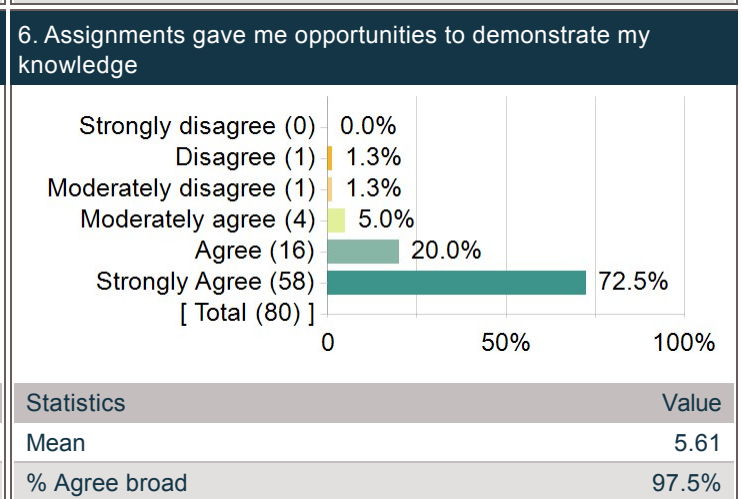
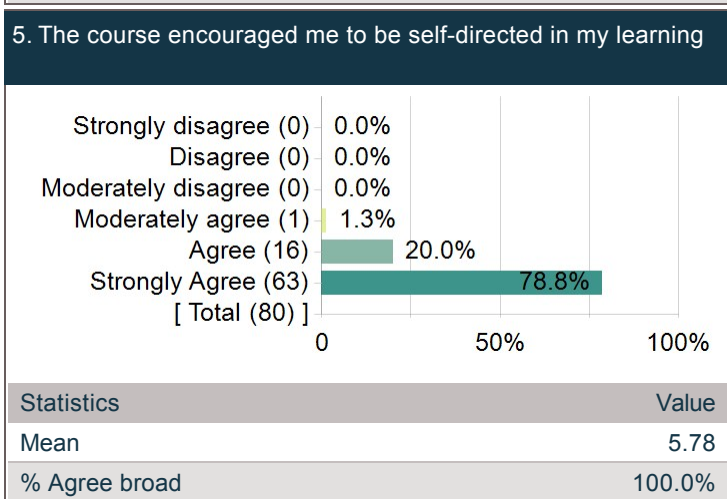
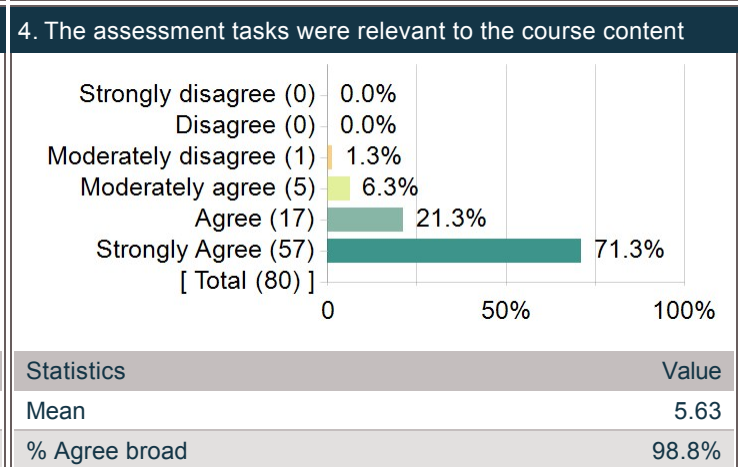
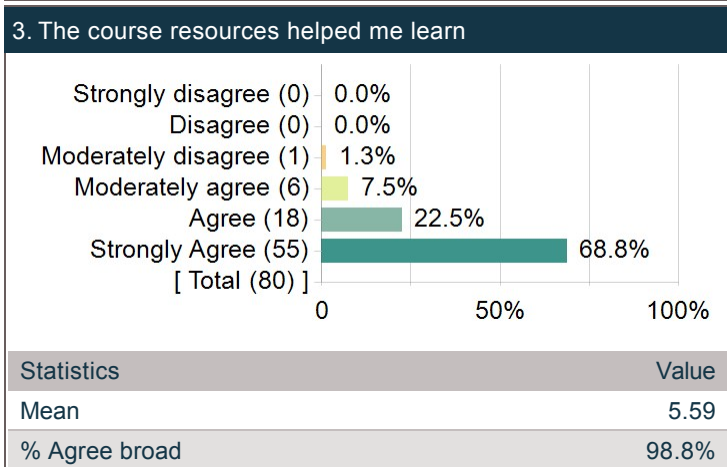
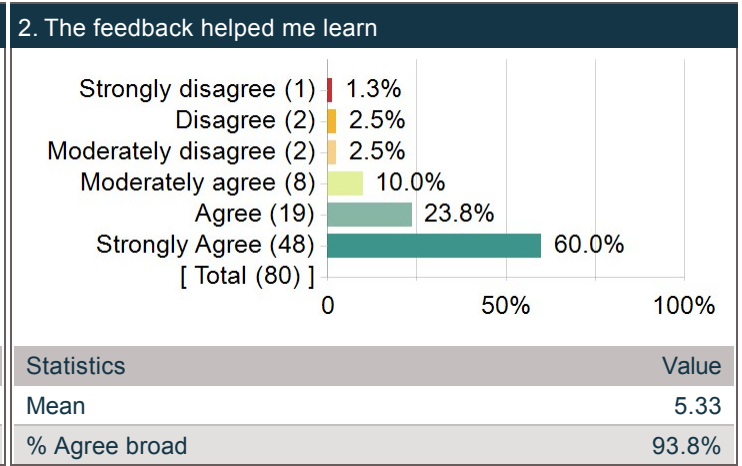
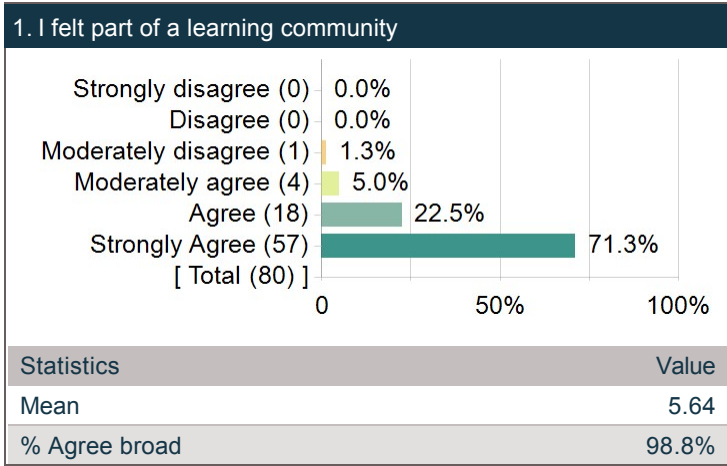
Overall I was satisfied with the quality of the course			
Options	Percentage	Statistics	Value
Strongly disagree	2.6%	Mean	4.90
Disagree	3.2%	Median	5.00
Moderately disagree	4.9%	Standard Deviation	1.20
Moderately agree	16.8%	Standard Error (base on SD)	0.01
Agree	35.6%	% Agree broad	89.2%
Strongly agree	36.8%		

Overall I was satisfied with the quality of the course



The table below shows the percentage of 'Agree' and 'Strongly agree' responses to the question 'Overall I was satisfied with the quality of the course'

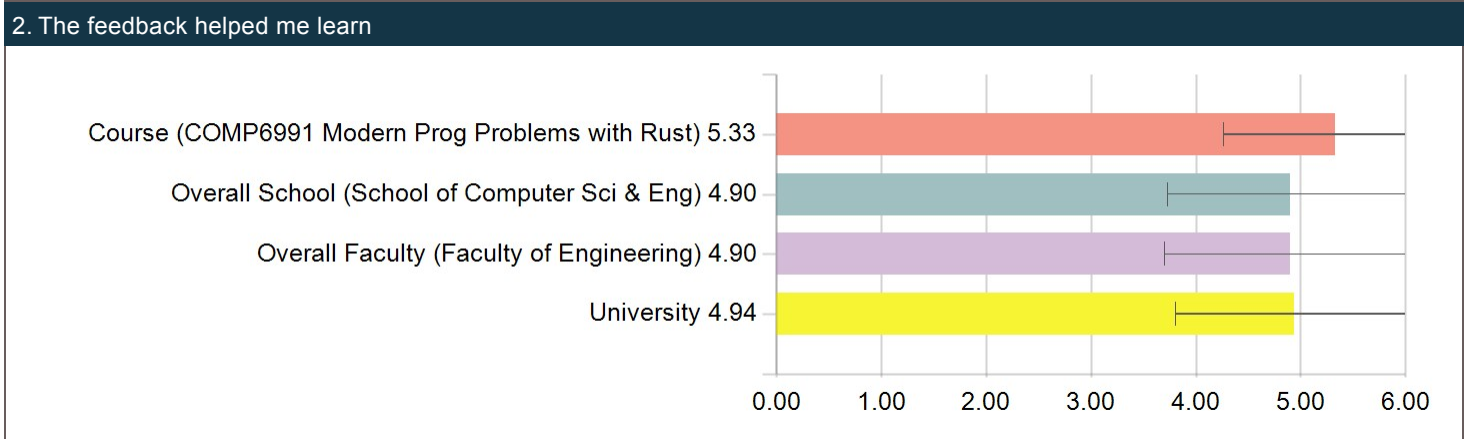
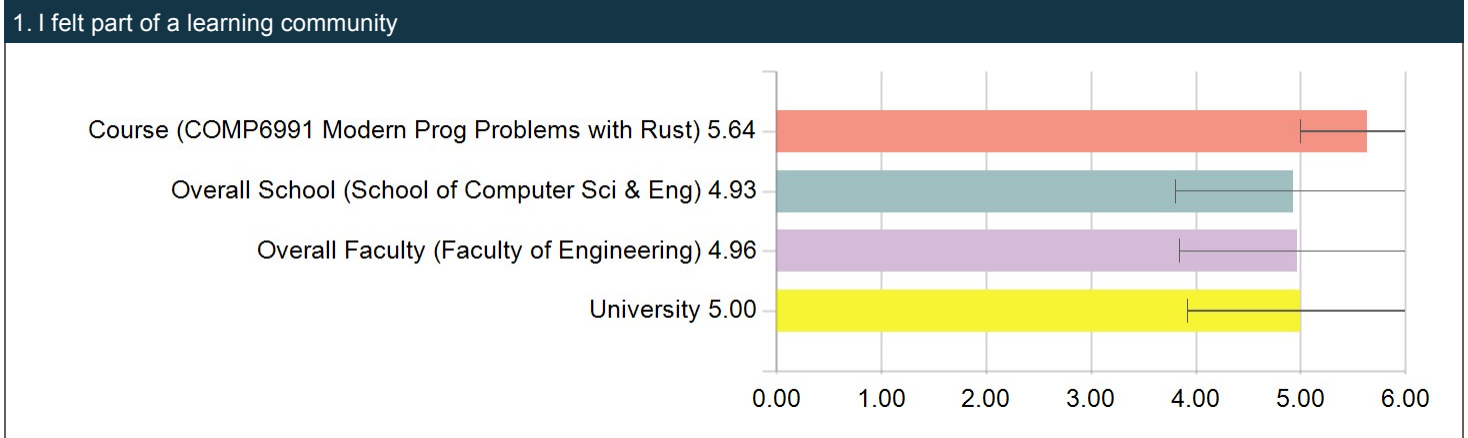
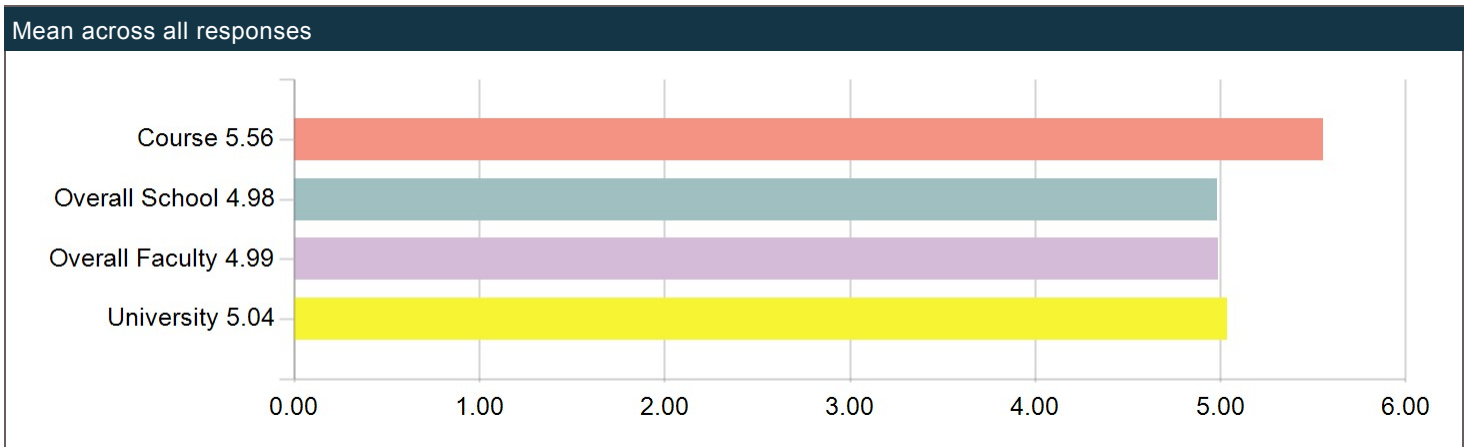
Overall I was satisfied with the quality of the course	
Statistics	Value
% Agree	93.8%



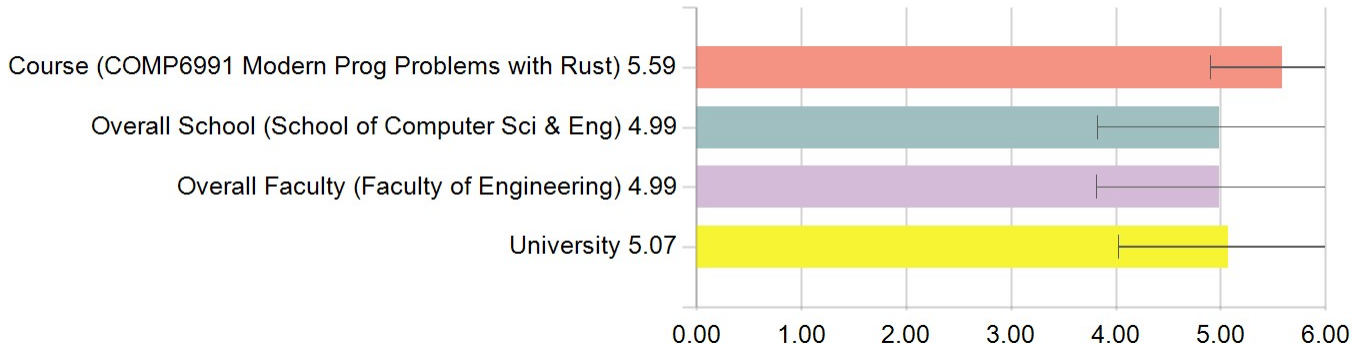
Comparison Statistics

Mean (average student responses between 1 and 6) and StandardDev (Standard deviation of student responses) are used for comparison statistics between Course, School, Faculty and University.

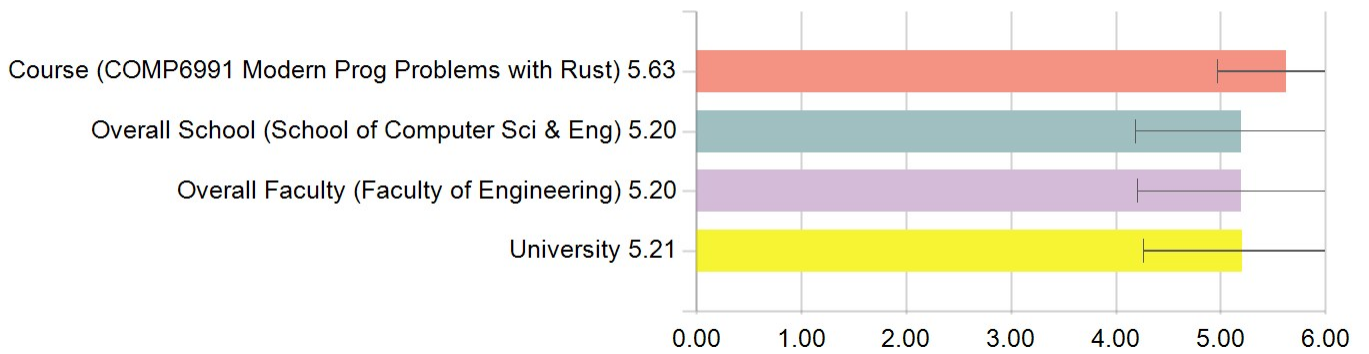
StandardDev



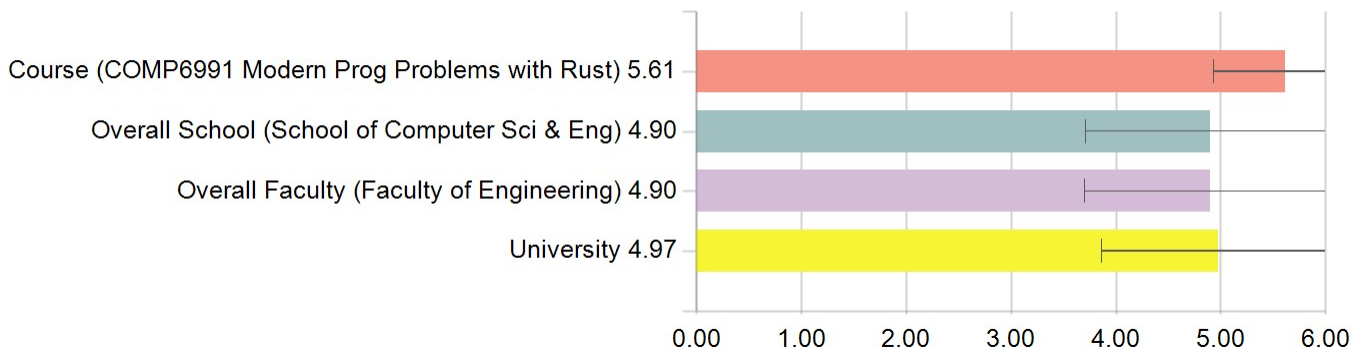
3. The course resources helped me learn



4. The assessment tasks were relevant to the course content



5. Overall I was satisfied with the quality of the course

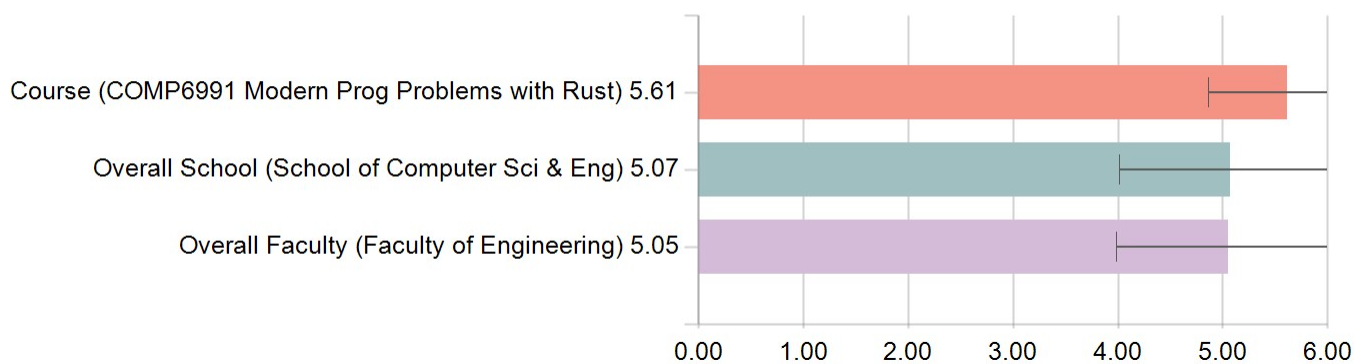


Faculty of Engineering specific questions

1. The course encouraged me to be self-directed in my learning



2. Assignments gave me opportunities to demonstrate my knowledge



Raw Comment Data

What were the best things about this course?

Comments
rust
11/10, such a good course! Lectures were really good and I liked workshops. The workshops got kind of low numbers at the end but I still found them fun.
I appreciate the fact that course staff acknowledge the use of generative AI tools especially considering its' relevance to our industry. I think it's a very forward thinking approach that probably wasn't a decision made lightly, and rightfully so. For the majority of the course however tools like chatgpt weren't a great help in that it would solve the entire problem for me but rather that it was much more like a better version of google that could answer weird questions like whether or not there was some obscure std library function that exactly fit this particular use case and nothing else. Additionally, I appreciate how responsive and helpful tutor feedback was especially on the forum. The responses from [tutor] and [tutor] were often incredibly insightful. Lastly I like how the course assignments have been structured in how open the problem solving is. More often than not especially in lower level courses (1521, 2521, etc...) it felt as if that there was only a single way to solve a problem and that solutions' would vary only by miniscule details such as checking the negation in an if guard rather than the truth value. However in this course, I found it particularly fun and challenging in the way that solutions presented themselves. There wasn't ever a clear solution that was the be all and end all 'right' solution but rather multiple solutions each with their +ves and -ves.
I love Rust, and I thoroughly enjoyed the course. It is taught with passion and I always felt motivated to strive for excellence. The content/assessments were also a breath of fresh air from the vast majority of other COMP courses which are completely outdated.
The passion of the admin team was clear and inspiring. This was a difficult class but I did not mind it.
hard but useful
Lectures were really entertaining. I also thought the angle of the course (thinking about when certain languages/programming paradigms, etc. are applicable and when they are not) was quite unique compared to other courses, and I think that it has helped me to think more critically when programming and reason more effectively about how to approach programming problems.
zac is a cool guy
Lecture are really freaking well. I love him!
learning rust is the best thing!
Best course
The course was very interesting and I enjoyed that it didn't just talk about Rust as language but also, why Rust has a particular design/enforces particular properties and how it compares to other languages.
interesting language, excellent lecturer and tutors
Course content very interesting and well designed. Labs, workshops, and assignments good way for practicing.
I liked the labs.
Workshops were very useful and helped strengthen my Rust knowledge. Weekly exercises weren't overly difficult but still tested my knowledge.
Self directed learning but interesting lectures & the workshops were very helpful
The course was actually really enjoyable, the workshops were a great idea.
BLOG POSTS were pretty great, especially because I didn't do a lot of the labs (and I like writing a lot).
Really really great course! Staff were wonderful and very helpful, really valued our time by not filling the assessments with useless busywork!
very good course, appreciate the effort put into constrcut the course structure, very friendly design in every detail throughout the course, a perfect course, would recommend to anyone, would love to see this structure been used in another course,
The course exercises were really good. I liked the exercise hints because I found usually even with them, that they didn't make the exercise super easy. I also felt that they were relevant and interesting. Similarly, giving some starter code was really helpful for the assignments.
Autotests were helpful and I liked considering the workload of students close to assignment due dates. It would be tough to be a lecture or two behind while multiple assignments are on, and then realise that there are 5 weekly exercises due that week that you haven't been able to start.

Comments
I also felt that Zac did a good job teaching and talking about things we could and could not do and why some ideas might not work, and fixing them, because these are going to be common trains of thought that students will have.
Content was hard but overall I really enjoyed this course.
– It is people like Zac, Tom, and Shrey that really make me proud to be doing CS at UNSW and that makes it the best uni in Aus for CS. Zac in particular is just contagiously passionate and enthusiastic about the content and in each lecture the vibe is so nice and relaxed and I'm honestly in awe of his abilities. Honestly it's my goal to one day be half as good at something as Zac is at Rust. This was my last term at UNSW and this course was a perfect send off. – The examinable structure of the course is basically perfect, your mark will literally reflect the effort you want to put in, case in point the assignments. I wasn't interested in the extended parts of the labs and assignments but I'm glad they are there for people who are super into it.
It was interesting
Learned something I would never be able to by myself
Shoutout to Zac, Shery and Tom, you guys are fantastic!
The tutors and lecturer were really good
The course instructors were very responsible and tutors responded to questions in the course forum extremely actively. The topic of the course is basically the same as the outline.
Very fun course. Staff were very dedicated in having us motivated to learn both the topics in this course and beyond
Great course. Awesome content and great lectures.
The observable passion with which this course is facilitated. Everything from Zac's teaching, Tom and Shrey's lab exercises, the terrific workshops. Prompt forum responses. Everything had such an energy of care and zeal about it. I love the workshop format and forum – how it encouraged such great debate and exchange.
The lectures were at a great depth.
Weekly exercises were quite fun too. Ass2 was particularly cool.
Workshops were great! really fun to just hang out in small groups and discuss/work on different problems. The tutors in the workshops were very friendly and approachable.

What could be improved?

Comments
Incorporate some form of individualised feedback in the interim, increasing the staff on this course if need be.
Although understandably a lot of work has been put into making workshops educational, I found that often times they would just miss the mark on engagement often because of upcoming assignments/labs due (including from other courses). This term I only attended workshops during the early weeks (1–3) and would only attend them if I required extra lab assistance. I'm not sure what the solution to this is either, I definitely empathize with the fact that having a lab session where students would work on lab exercises would run into the exact same problems as well. Perhaps more motivating factors for workshops such as mandatory attendance or even attendance marks could remedy this. One suggestion I have is maybe having something along the lines of blog posts but instead with workshop attendance i.e. attending all 9 workshops is the equivalent of 1 weeks worth of lab marks. This way students would have another way of remedying marks lost from labs.
Side note: not sure if it's right to nitpick here but thought I'd mention it anyway, the workshop times for timetabling were completely horrible. I found that the workshops always ran too late which was also another factor (albeit small) as to my low attendance.
Feedback on assignment 1 and weekly exercises sooner.
Often in the course it felt like I was having to write code that produced a lower quality result for the end user in the interest of the "best practice" or "idiomatic" way of writing code, including avoiding unsafe code which was trivially easy to prove was safe. I find often the term "idiomatic" is used to justify a way of writing code in a way that is not relevant to the actual output, and I found it deeply frustrating having to write code to unrealistic and contrived rules which forced the output to be worse. This pertains to the assignments only.
Honestly, I found I got nothing out of the workshops. I did not like having to work with a partner; it just felt like I was explaining the content the whole lab. The labs felt too large to address and didn't help me learn anything new. Smaller, more focused questions like the weekly exercises were more useful to me.
Though, in the labs, I really appreciated the tutors' knowledge on the weeks subjects.

Comments
at least some help for the assignment
Best course I ever learned
use edstem! use edstem! use edstem! use edstem! use edstem! use edstem! use edstem! use edstem! use edstem! fix cargo and 6991 cargo problem
Maybe some single option questions added for labs to understand come concepts
Unfortunatly my experience of the workshops was mixed. I felt that some of the tutors did not have a comparable understanding of the content and this made some of the workshops fairly boring. I ended up going to some other workshops outside of my tutorial time as I found other workshops to be more engaging (particularly Tom Kunc!) and were able to answer more of my questions.
Delete plugins from assignment 2 pls :(
I really enjoyed it and to be honest my only problem with the course was that my brain fell out of my head around week 3 and didn't come back which somewhat hindered my experience, but the course itself was pretty great.
Only thing I would change is make the lecture slides more informative (rather than delivering all the info verbally) to help with revision :)
course forum don't have a separate tag for assignment1 and assignment2, emm, some suspicious blog post seems like made by chatGPT, like this ML in Rust? contain literally no information at all
I found getting started for assignment 1 quite difficult. I got a bit lost in the termgame library looking at ChunkMaps and CharChunkMaps. I thought that I might need to figure out a way to load the map by using swap_chunkmap. Eventually I figured it out but it was quite intimidating at the start.
For assignment 1, I found a suggestion by [tutor] to be helpful. He suggested to start with parsing from a file before implementing player movement. This made more sense to me because when you understand how to load the map, you will understand how to move the player around as well by setting and unsetting the player character at a given (x, y). It also means that you can see the player moving so you know it is working.
Whereas when I was trying to do player nav before using any kind of map, I found it a bit confusing because I could place the character on the screen, but didn't have any representation of the map to work with.
For workshops I found that they can be quite hard. Maybe it is due to the learning curve of rust at the beginning, but I really did not understand how anything worked at the start, so I didn't feel like I got that much accomplished during workshops. I think at the start the workshops felt a bit tough to do. There would be a long list of questions that didn't seem very trivial. I think it would be good to have some simple questions to start off with, because at that point most students are really trying to get over the steep learning curve of Rust and are hitting the usual compiler errors. It would also help because the Rust docs are well written but long and can be confusing to go through and try and find a function you need when you're starting out.
For lectures, I think a few slides that could be used as a syntax cheat sheet/summary would be very helpful as well since it's hard to remember every example in a lecture. You need to go back and skim through it. Whereas with at least some summary notes, you know what you need to Google if you find something promising.
For example, &T is a shared borrow of T. Allows reading but not mutating. &mut T allows mutating.. Or: traits are essentially templates of function signatures that a type needs to implement.
1. I think there needs to be a better guidelines for new students on what level of programming is needed prior to taking this subject. I really struggled throughout the course because I may not have enough programming knowledge (post grad student MIT) 2. We did not receive any assignment 1 feedback prior to assignment 2. So i struggled to understand if I was truly understanding the core concepts in this subject throughout the trimester. 3. Assignment 2 was quite chunky
Recorded tutorials, or even solutions from these tutorials uploaded. Assignments feel wayyyy too difficult without having attended these tutorials, it feels like if you don't show or can't show up to tutorials you are shooting yourself in the foot. Help sessions would also be a nice addition or help for this course, and maybe a lecture or something on style or design. Since most of the assignments focuses on design, it feels like most of this should be known before this course, also assignments could be more specific on what is classified as "bad design" (I was very unsure on what was allowed and considered fine design and what wasn't).

Comments
Not knowing borrowing for so long made me write such bad code it made me physically disgusted with myself.
<ol style="list-style-type: none"> 1. I really hated how the lecture is live-coded and there are not enough notes. When the scripts are posted on git there are no comments and I can't understand the track history :(Really hard to follow through even when I listen to the recording multiple times per lecture :(2. I find the course very intense and not friendly for newbies like me. It goes on steroid after borrowing and there is not enough 'breathing space' for me to digest the items when there are weekly hw + lab + assignment. Should release more pre material at week 0 3. Assignment is too hard... I really struggle as a Rust Newbie (and not gonna lie, I get why Rust is safe etc., but it's very pedantic to use!)
<p>Assignment 2 NEEDS to tell the student about the types file, MAYBE have documentation for it too.</p> <p>There are so many "pre implemented" types and methods and such the spec does not tell you about.</p> <p>For example : if the spec asks you to implement X command that may give A, B C error, it doesn't mention that the given type parser (which the spec does not mention) already handles errors A and B.</p>
Assignment and labs felt really hard. high workload compounded with high difficulty
In the future, perhaps it could take advantage of some modern tools, such as GitLab, which can combine with assignment and lab to improve the ability to use VCS.
Marking could be improved. From the start the long assignment marking times were foreshadowed so that was OK, but lab marking also took a long time and could have some improved quality control. Might not be super popular, but pulling the assignment 1 due date earlier into the term to try to have marks back for assignment 2 would be good. I felt pretty blind going into assignment 2 with practically zero feedback.
<p>I think try as it might the course still did feel a little like "learning Rust". And while I was thrilled to have that opportunity (its the only chance I've had to really dive into the why behind decisions made in a language – I've learned SO MUCH, as opposed to just rushing to solve some problem in language x), and the course DOES make an effort of pointing out the flaws in some of the language design decisions (in painstaking detail – I will never look at christmas lights the same again, thank you Tom), some of those efforts feel a little surface level when a lot of that discussion doesn't translate into the work.</p> <p>If I were to recommend the course to a friend, ignoring the brilliant course delivery, I would say something like "if you're interested in learning Rust". Where as I'd love to say something more like "yes I learned a lot about using the right language for the task and why Rust is good and bad at that"</p> <p>I think if you could play into the idea you're going for more with the weekly exercises that would be good. I mentioned the christmas lights one because that is a perfect example of a brilliant exercise. It gets right to the heart of the cost involved in not having NULL values. Something closer to how the industry might approach Rust as a language. Maybe some code migration or translation from something like C to Rust? Or writing Rust tools that have to plug in/integrate into an existing C++/C codebase (like the Quest part of Ass1 could be its own assignment where you write a crate for a quest system that could be called by C++ game). Or model an assignment on a real world example of Rust being used to solve a programming problem (like Discord having crazy CPU spikes every two minutes, for example)– get an existing codebase that has a problem and have to solve that problem with Rust.</p> <p>Also I wonder if there were a way you could showcase Rust incredible performance when written well. I think making assignments have some performance benchmarking would be too much but is there a way you could maybe have some sort of informal competition across the trimester with the workshops? Where workshop teams could compete somehow for bonus marks. Would encourage comradery and for students to do their own research into increasing performance outcomes – again sorta modelling how the real world looks at Rust.</p> <p>Also please, please, please include WASM in the course xx.</p> <p>I wouldn't bother leaving such a detailed "improvement" comment if I didn't thoroughly love the subject and was so invigorated by the admin team and their passion for the course (honest to god probably saved me from uni burn out). These are all more "nice to have" and maybe some food for thought if you're looking for it.</p> <p>The bonus questions in the labs should be worth bonus marks in my opinion. This could act as another way to make up for lost marks, like blog posts, while still encouraging students to engage with the course content beyond what is strictly required.</p> <p>A few other minor issue. Some of the lab questions had unclear descriptions, or unhelpful autotests (I think the channels lab question let you pass the autotests even if you missed part of the functionality.)</p> <p>The separation of stages in assignment 2 was a bit counterintuitive. It's tricky to split it up nicely but having stage 1 being single client and stage 2 being multi-client means that you'd have to rewrite a lot of your code if you actually tried to implement stage 1 before stage 2. Most people I spoke to just tried to solve both stages at once (using threads from the beginning).</p>