

Project Title: **Course Evaluations Fall 2024**

Courses Audience: **59**
Responses Received: **55**
Response Ratio: **93.2%**

Report Comments

Guide to the Interpretation of Course Evaluations at UT Austin

The goal of course evaluation process at UT Austin is to drive teaching excellence and to support continuous improvement in teaching and learning experiences. The two sets of scales used for core evaluation questions and the associated weights are:

Strongly Agree (5)
Agree (4)
Neutral (3)
Disagree (2)
Strongly Disagree (1)

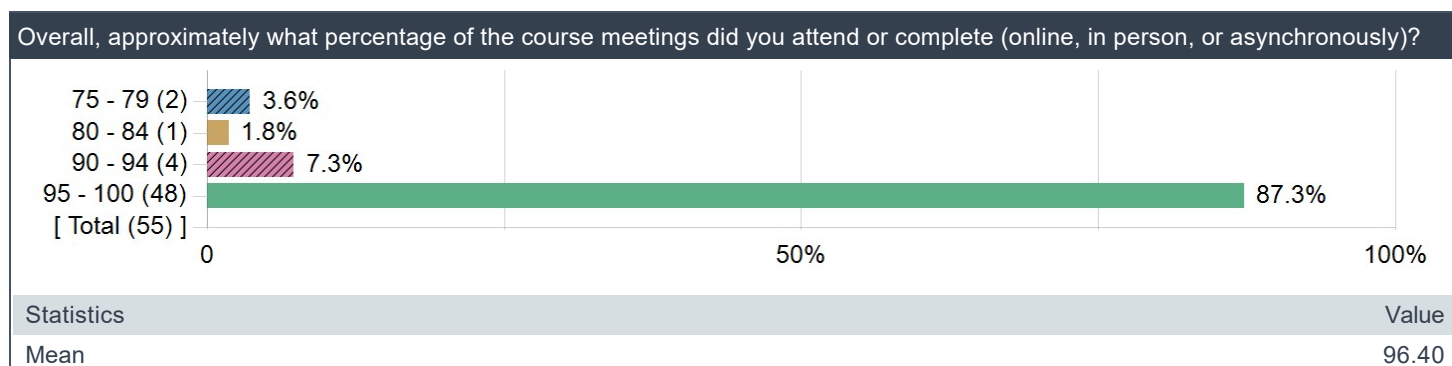
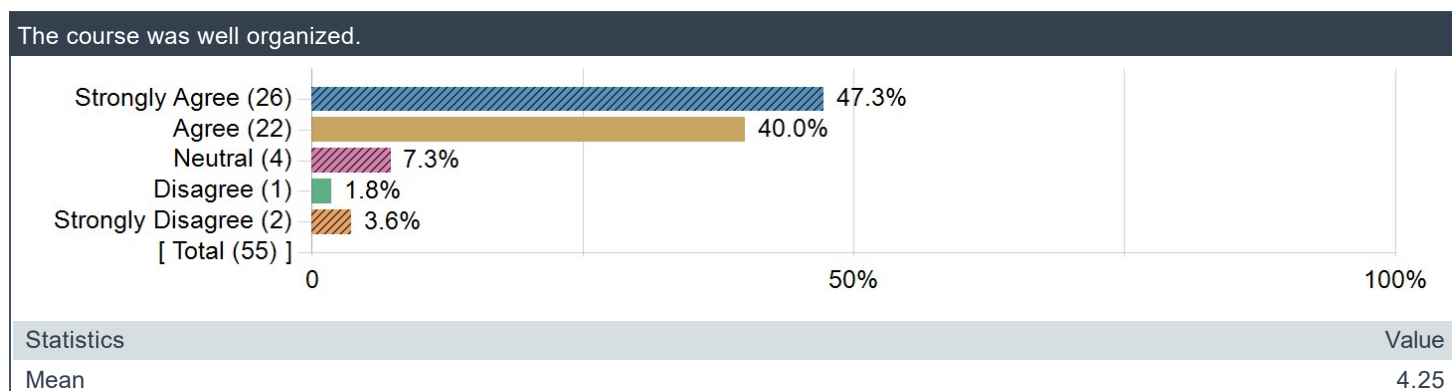
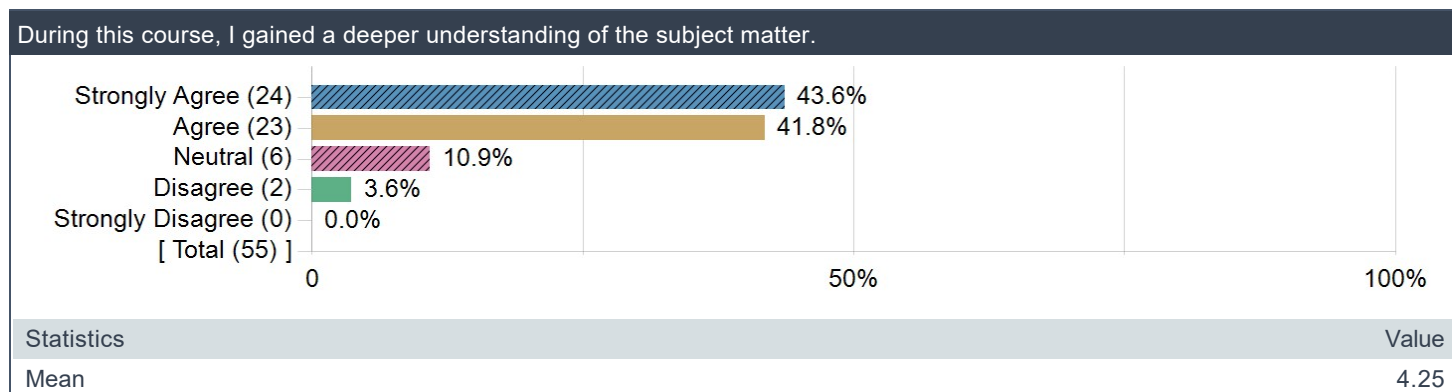
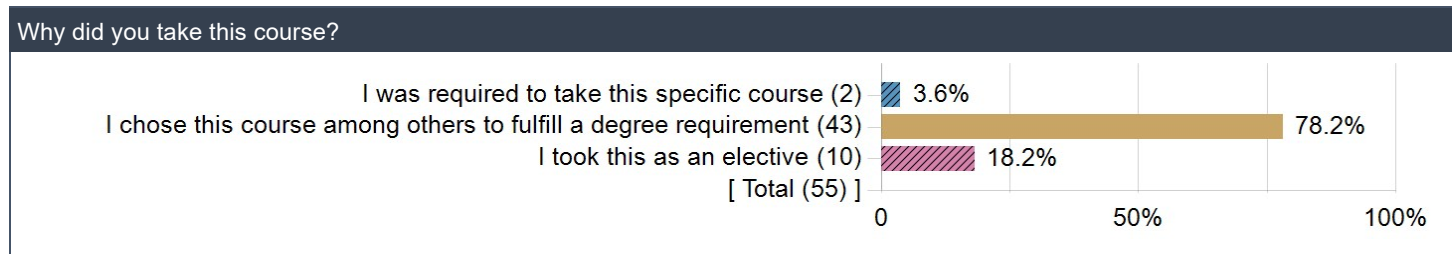
Excellent (5)
Very Good (4)
Satisfactory (3)
Unsatisfactory (2)
Very Unsatisfactory (1)

The Mean is calculated by adding all of the weights for a single question and dividing by the number of respondents. The course workload question is not averaged.

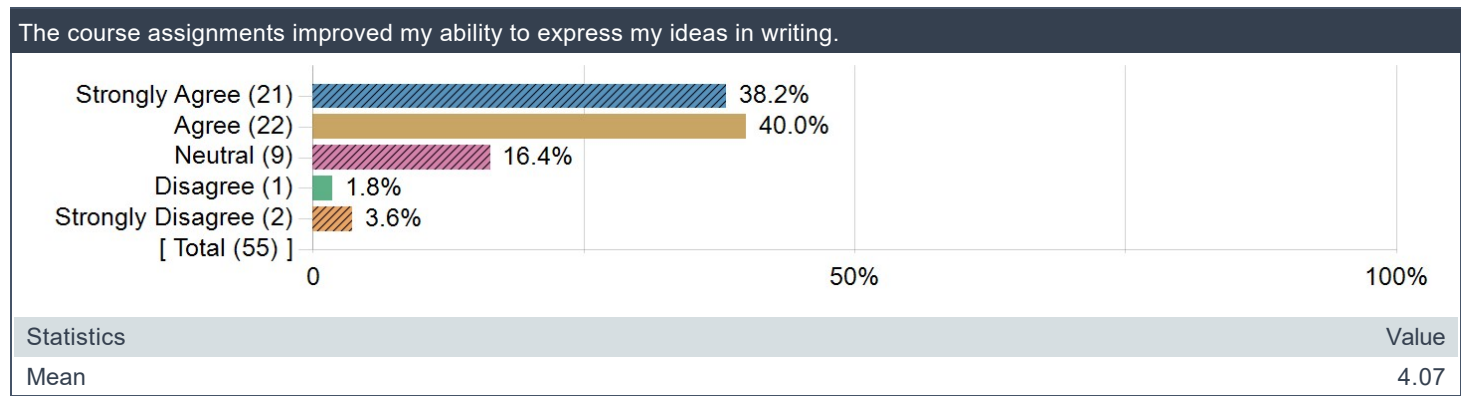
The number of students (e.g. respondents) marking each option is reported for each of the items. These frequency distributions provide information about the level of student ratings and the spread and shape of the class distribution of responses. The distributions thus provide a picture of student perception of a course.

Course evaluations provide snapshots of student perspectives on their course-level learning experiences. Most experts on teaching evaluation advise that no individual method gives the complete picture of an instructor's teaching effectiveness; multiple and diverse measures, on multiple occasions, are advised to give a full picture of the teaching effectiveness of a particular instructor. Moreover, other factors, such as size of class, level of the class, and content of the course, can cause small variations in the ratings. Therefore, student perspectives for a particular instructor or course should be interpreted as a snapshot, and not as providing complete information on the teaching effectiveness of that instructor.

Course Questions



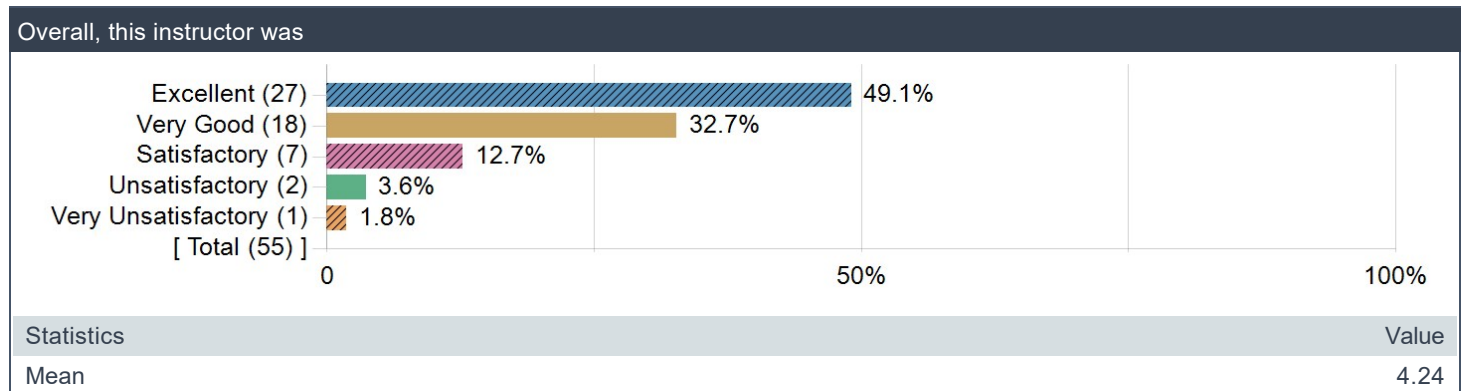
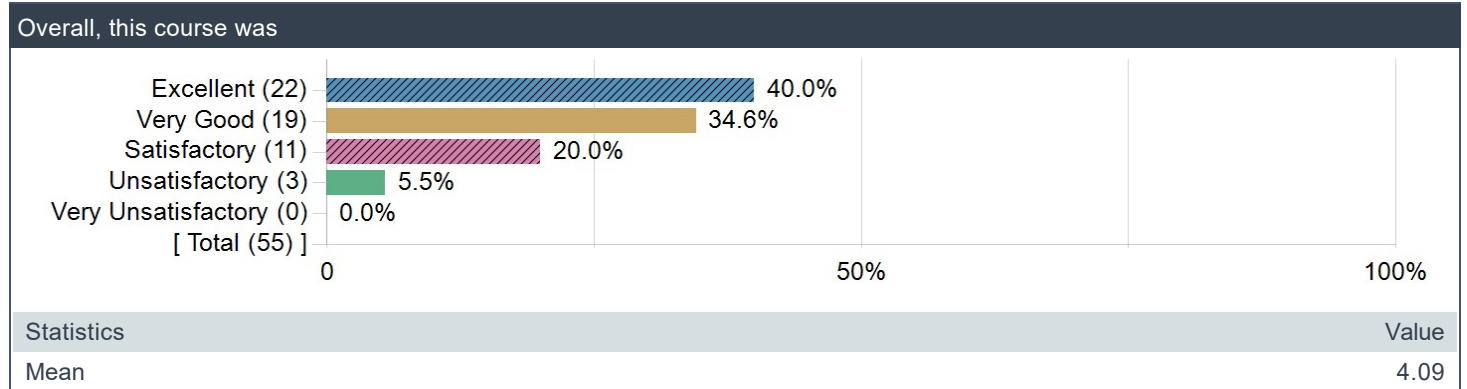
The course assignments improved my ability to express my ideas in writing. (Flag Question)



Instructor Questions

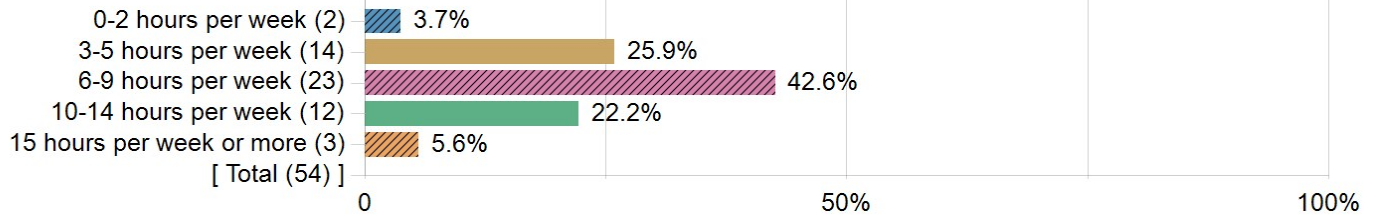
	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree	Responded	Mean
The instructor clearly explained the course objectives and expectations.	56.4%	30.9%	12.7%	0.0%	0.0%	55	4.44
The instructor fostered an inclusive learning environment.	49.1%	36.4%	14.5%	0.0%	0.0%	55	4.35
The instructor effectively explained the concepts and subject matter in this course.	49.1%	38.2%	9.1%	3.6%	0.0%	55	4.33
The instructional techniques kept me engaged in learning.	45.5%	30.9%	14.5%	5.5%	3.6%	55	4.09
The instructor checked for student understanding of the concepts presented in the course.	52.7%	34.5%	12.7%	0.0%	0.0%	55	4.40

Overall Questions



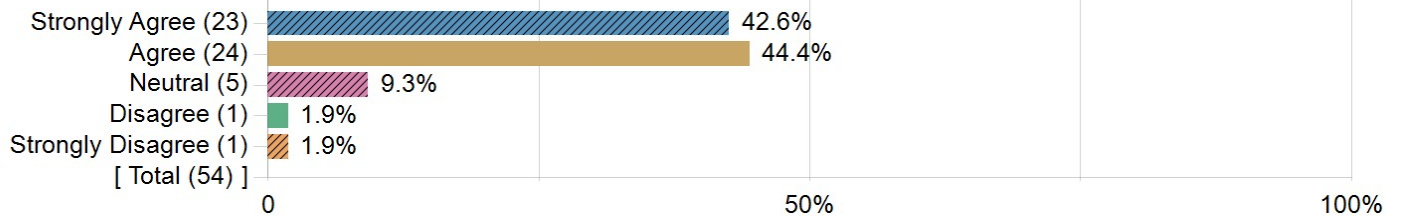
College, School, or Unit Questions

On average, approximately how many hours per week did you spend working outside of the course? Include time on homework, reading, reviewing, papers, projects, etc.



Statistics	Value
Mean	3.00

The course format (online, hybrid, face-to-face) helped me to learn.



Statistics	Value
Mean	4.24

Comment Questions

Identify aspects of the course that were the most effective in helping your learning.

Comments
I thought the projects forced us to take initiative in our own learning and thus I ended up learning more than I expected to.
i think the project split into phases really allowed me to pace and take in the workload of creating a full stack web app.
Having exercises that related to the lecture material was very helpful in actually understanding implementation beyond theoretical.
In class exercises
Exercises and the canvas quizzes.
Professor Downing was very organized and has planned out the course effectively.
Projects, lectures
The lectures were very well organized and made sense.
The papers were interesting to read.
The in-class content was insightful.
Great teacher
The daily quizzes helped keep me accountable for my learning
The group projects / projects in general were the most effective in my learning.
The lecture content was greatly informative and I learned a bunch of useful knowledge.
The lecture style was very engaging, and the exercises helped my hone my understanding of lecture material.
The in-class exercises
I think that the lectures presented a lot of interesting information that isn't really covered in other classes. Also, working on the project throughout the course helped me learn a lot more about full stack development.
The in class exercises were helpful for practice.
The lectures were informative and easily to follow. I think the cold calling helped the class stay engaged. The papers were helpful in understanding design patterns.
The most effective aspect of the course was definitely the project. The project allowed me to truly learn and practice real-world software engineering as we were developing a website. This was more helpful than anything else learned in the course and is the reason I took it.
Quizzes and Exercises
I liked the collaborative exercises since they let us work together to learn
Enthusiasm about subject
Quiz
I thought the project was most effective in actively learning.
The lectures
Coding exercises in class
The project and having to learn how to use tools / do web development without help
The lecture notes were useful for reviewing and solidifying information.
I believe the weekly readings were a great way to introduce topics in lecture and how they are applied in real world scenarios.
I thought the cold calling was helpful even if I was intimidated initially, it really helped me maintain my focus in class and even when I messed up it wasn't embarrassing.
Cold calling and making class a conversation between two people helped me stay engaged with the lectures and what was being taught.
The project were the most effective in helping me learn.
The projects were the only part of the class where we learned typical "software engineering"
The group projects
Very engaging professor. Though the cold calling made me nervous at first, it helped reinforce my learning.
I think calling on students during lecture is a very effective way of ensuring that students pays attention and attends class.
The instructor had many rhetorical techniques to give hints and suggestions when someone doesn't know the answer.
Very nice and very entertaining.

Identify the aspect of the course that you found most challenging, why you found it was challenging, and

suggest one thing that could be done to help future students meet that challenge more effectively.

Comments
I think some more in class instruction would be helpful. The first phase of the project seems virtually impossible if no one has ever done any full stack app before. I think a general guide on where to look to start would help immensely. I also think the cold calling kind of puts people in survival brain mode and it makes it really awkward for other students when the professor just keeps grilling them.
I found working on the projects consistently after phases was a bit tougher because it is instinctual to take some break after completing a phase. However, Downing gave substantial amount of time for a little down time between phases.
Certain parts of the projects were very difficult due to a lack of information on how to fix problems. There isn't much of a way to fix this since there is a lot of freedom given in implementation.
Project time management
Self-learning because I wasn't used to the structure. However, I think it was an important skill to have.
Sometimes the details of assignments were difficult to interpret.
Last project, instructions were kind of vague, but it wasn't too bad
Getting direction on the projects. Have clearer instructions / make it clear to the TA what is expected at each phase.
The projects were a bit challenging at times.
Great
A lot of the technologies we used for our project were unfamiliar to me. The lecture content was very loosely related to the project as well. A lecture or two going over how to use some of these technologies could be helpful.
Assignments like the papers, blogs, etc. seemed like tedious, busy type of work. Not necessarily challenging on its own but it can add up. The cold calling in class was probably the most difficult aspect if the code was confusing for you and what not.
The most challenging part was the projects, as the lectures didn't really cover the project content and was rather separated from one another. We had to figure out how to do things by ourselves, so teamwork definitely mattered a lot. Active communication is always key!
The projects were most challenging due to their scale. If possible, there could be more TAs in the future so that they could spend more time on each group
The most challenging parts were reading through some of the longer papers with tons of detail
I think the aspect of the course I found the most challenging was that we had to complete a project throughout the course that was not related at all to what we were learning in class. It made the project a lot more difficult since we had to do a lot of independent research. I think it would have been helpful to go through some of the aspects of the project in class to provide more context and guidance on how to approach it effectively.
I think that having to self learn everything that we use is annoying, I took the class so that I could learn software engineering from a professional (you), not myself.
I found the projects to be the most challenging. I would have preferred if we spent less time going over Python in class and had some lessons on react / backend technologies like flask so that we didn't have to figure out the entire thing on our own.
The most challenging aspect of the course was the in-class quizzes every class. I found it very challenging to have to come to lecture every day to take a quiz about the last lecture's content. Sometimes, the busy life of college students makes this difficult. I would suggest keeping the quizzes but changing the structure to accommodate the students better.
Labs
I thought the projects were pretty challenging
N/A
It's hard to implement website for beginner while the class material didn't cover that. And also it was hard to find adequate for group meetings.
I think making the lessons more geared towards the project and software development technologies would be helpful
The projects. Sometimes I found it difficult to find all the relevant information for a project since it was scattered across the phase description, the rubric, and the issues document.
Quizzes, could use more notes/examples
I don't like how most lectures aren't relevant to the project. I wish we had more lectures covering how to use more tools we were required to use on the project.
I found the exercises to be the most challenging because of the limited time you had to complete them. One thing future students could do to effectively complete the exercises is to work with the people around them.
I found the quizzes to be the most challenging as they were quite the time crunch. I'd say one thing to improve this would be allowing more time.

Comments
I think that having a bullet point rubric that we had to make into gitlab issues was a bit annoying and arbitrary. I think just having a csv along with the rubric after P1 would be nice
The exercises and the projects were the toughest for me, mainly because I am pretty new to python, and learning how to host a website was also new to me.
The projects were challenging but I would not change them because they taught me so much.
The projects were somewhat challenging.
The intro solo project, i couldnt figure out gitlab and officehours are far few and between so i took the loss. Pipeline
N/A
I think the daily quizzes is probably the most challenging part, not because of the difficulty but instead trying to stay on top of it.
I mean the homework stuff was basically self-taught, so it felt like there was a large disconnect between what we're learning and what we're doing.
Learning Web Dev from scratch was harder than I expected.