Individual Instructor Report Fall 2025 Version A for C S 373 - SOFTWARE ENGINEERING (55090) (Glenn Downing)

Project Title: **Course Evaluations Fall 2025**

Courses Audience: **58**
Responses Received: **56**
Response Ratio: **96.6 %**

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. 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. For additional details, including the scales and how the Mean scores are calculated, please review the Report Guide at the end of this document or, UT Austin's Viewing Course Evaluation Results webpage.

Creation Date: **Friday, December 19, 2025**

Core Questions

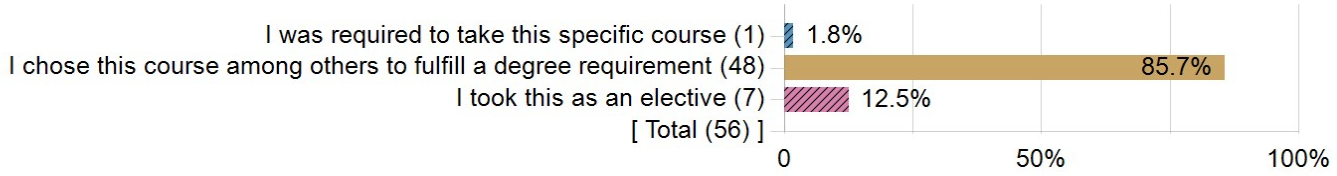
	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree	Responded	Mean
During this course, I gained a deeper understanding of the subject matter.	41.1%	44.6%	8.9%	3.6%	1.8%	56	4.20
The course was well organized.	42.9%	41.1%	8.9%	7.1%	0.0%	56	4.20
The instructor clearly explained the course objectives and expectations.	48.2%	44.6%	5.4%	1.8%	0.0%	56	4.39
The instructor fostered an inclusive learning environment.	58.9%	32.1%	5.4%	3.6%	0.0%	56	4.46
The instructor effectively explained the concepts and subject matter in this course.	48.2%	35.7%	12.5%	1.8%	1.8%	56	4.27
The instructional techniques kept me engaged in learning.	51.8%	28.6%	14.3%	5.4%	0.0%	56	4.27
The instructor checked for student understanding of the concepts presented in the course.	57.1%	35.7%	7.1%	0.0%	0.0%	56	4.50

Overall Questions

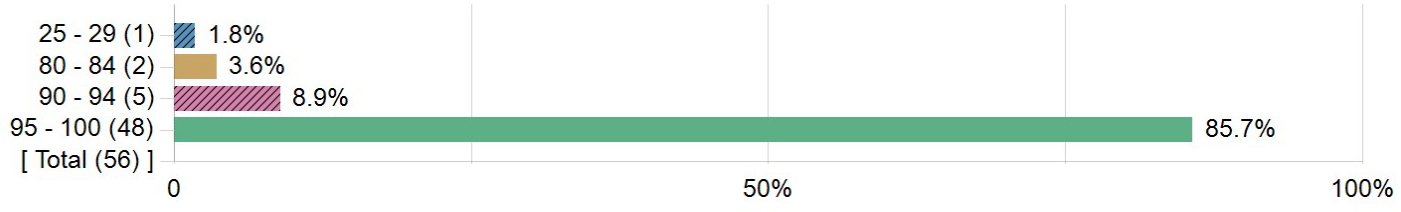
	Excellent	Very Good	Satisfactory	Unsatisfactory	Very Unsatisfactory	Responded	Mean
Overall, this instructor was	46.4%	33.9%	16.1%	3.6%	0.0%	56	4.23
Overall, this course was	32.1%	33.9%	28.6%	3.6%	1.8%	56	3.91

Other Course Questions

Why did you take this course?



Overall, approximately what percentage of the course meetings did you attend or complete (online, in person, or asynchronously)?



Statistics	Value
Mean	95.52

Comment Questions

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

Comment
The lectures taught intermediate python topics that were useful to know but I don't understand how they are relevant to the projects. The TAs in office hours were also quite helpful in answering any questions.
The collaborative quizzes, exercises, and projects.
projects
the projects were really good for learning
The rubrics were very clear on the projects.
The lectures were very effective for learning the concepts.
Daily quizzes and exercises help me practice concepts learned in class.
Cold calling while in class was really effective in helping me understand the material
The professor calling on people makes you pay attention in class.
Professor Downing was very knowledgeable on class topics and had very good examples to show, and that really helped me get a deep understanding.
He cold–called on people meaning you had to pay attention
The projects were helpful in really learning about the content
In class exercises and example problems performed to explain complex topics
Lectures, exercises, and the projects helped me learn the most.
for Software Engineering specifically, it was helpful to learn a bit of SQL in class. But would have appreciated course content involving more software engineering related concepts
The projects and my teammates
Exercises
The problems were effective. They allowed me to practice with Python and gain confidence using it. Also, I liked the team project.
Lecture topics towards the end of the semester were helpful and seem relevant to software engineering design
I enjoyed the lectures being interactive I feel like it helped me stay engaged and pick up on more of the content.
Clear grading rubric for like the overall class the like 39 quizzes for an A thing, clear grade bounds, Professor Downing was very responsive on Ed and proactive with Canvas messages,
The aspects that were most effective were the deep explanations of the attributes of Python and some simple functions in SQL, all of which made me more proficient in the two.
I enjoyed the collaborative nature of the class on quizzes, projects, exercises, because it helped with my understanding.
Cold calling and daily quizzes were helpful! It helped me be engaged.
The cold–calling on students in lecture helped reinforce my understanding of the content for quizzes.
Cold calling during class really helped with learning that day's material. It forced us to think about the problem and come to a solution.
I liked how we had multiple activities and methods of learning including in–class quizzes, projects, exercises, problems, and papers.
Constant and variety of assignments keep students engaged both inside and outside of class.
I enjoyed reading over the papers and learning about SOLID principles n other coding practices
Collaborative quizzes, exercises
Office Hours were helpful, organization of work was helpful.
The projects were very helpful in learning how to manage real life systems with AWS and other tools.
Having a hands on project was very helpful in learning how to actually develop a website. Additionally, the lectures were helpful too.
The instructor was very open to questions. Ed discussion was a good resource for course questions.
Professor Downing is a pretty down to Earth guy who makes the class environment seem very chill and relaxed. While at the same time, he was very passionate about teaching and making sure the students understood the material, which I appreciate a lot.
The lectures were effective.
The projects and the lectures. The projects were good at testing practical and techinical skills. The lectures helped with our

Comment
fundamentals.
I like Professor Downing's way of teaching, and I liked the organization of the class materials.
The cold calling is good way to stay engaged. When it's not you called, you still want to pay attention in case you are next, and I think its fun to try and respond correctly in my head to see if I am properly following/understanding.
Going over things like SQL and unit testing were good and helped me get an insight into those tools.
I appreciated a lot of the specific, guided examples in class, and the exercises were helpful for gaining experience with the ideas, even though sometimes I felt they were a bit strict on time. I also appreciated the project work, even though it was not reinforced in class necessarily. I also very much appreciated the design principles in the papers. They were practical and insightful.
The way in which Downing teaches makes it easy to learn and interact in class. In addition, the constant papers & problems makes it so we are learning other things that might not be taught. The lectures and notes being all online make it easy to review back as well to remember what we learned. The daily quizzes helps keeps us on track as well, as whenever I started slacking, I would still look at the past lectures to catch up if i ever missed class.
Having quizzes every lecture to help us deepen the understanding of what we have learned
The projects were actually helpful in learning software concepts.
I appreciated the in class quizzes and the way that we get called in class because that makes us focus in class and understand the material to be able to answer.

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.

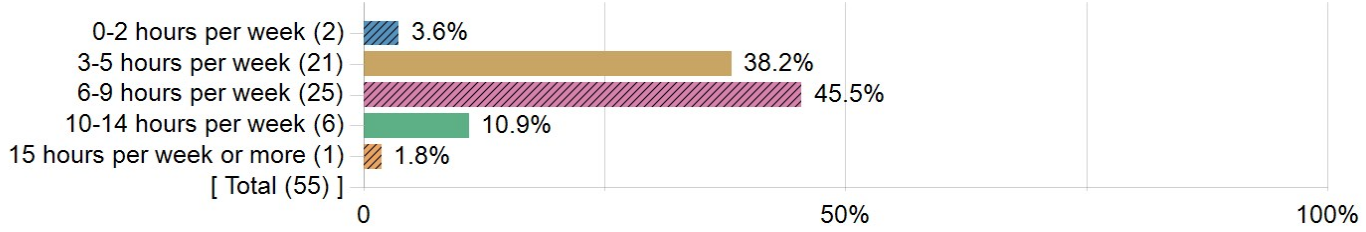
Comment
The "group" aspect of the projects was most challenging to me. Our group had one less person than the usual amount and our TA had been unresponsive at times and didn't schedule any check–ins throughout the semester. Moreover, one of our group members wasn't participating a lot and did work quite late (often 1–2 days before the deadline) which affected the rest of the team as some group members' work was dependent on it being completed. This also meant that errors or bugs would come up last minute and we wouldn't have the time to resolve them. While the work assigned wasn't particularly challenging, the group experience was quite stressful in trying to pass phases. I think to reduce the stress of being let down by groups or unexpected circumstances, students should be graded individually in a group by formally assigning themselves parts of the rubric and then graded on how well they met those specific requirements. I think it makes sense that not all group members need to suffer a fail in a phase as a result of another team member's fault. In addition to this, I feel like the content covered in class should be a bit more relevant to the project. I do think that the intermediate python knowledge was useful to learn and I did learn quite a lot, but it would have been a bit more useful if we were taught how to use some of the tools required for our project or if things were clarified a bit more.
Exercises were sometimes very difficult and stressful on a time crunch, especially when we didn't know how to implement what we had just learned. I would suggest letting students know when to pay extra attention because we will have an exercise coming up, or have us complete it from home without the time limit. Projects were also difficult because we weren't taught how to use all the new tools and could've used more guidance.
projects
i think finishing the project and making sure we met all the requirements
Doing the projects when the projects didn't really relate to anything we learn in class.
The projects were by far the most challenging. More resources to help learn to do what is needed would be helpful, or perhaps a guide.
The most challenging part of the course was the disconnect between the projects and the content we learned in class. In class we learned about python and its ins and outs but on the project we were dumped a whole lot of tools we didn't know how to use.
I felt that the lectures had nothing to do with the project.
Some of the timed exercises were more diffucult than others
The projects, especially setting up the hosting.
A lot of what you do in the projects may require you to self teach yourself since its not covered in class so over I think giving students a heads up and maybe some resources they can go to may be helpful.
The project instructions were a little vague to understand.
The projects were the most challenging part due to the amount of requirements for each phase and working in a group of 5
The projects were really hard for me. Our group was unable to meet frequently (our entire group has only met up once so far) and coordination became extremely challenging as a result. The CATME role assignment was supposed to align our groups based on

Comment
availability, but our entire group almost never had large period where we were all available, only an hour or two during the day and a few hours late into the night.
Projects can be challenging if you have no previous experience in web development. If the lectures would incorporate concepts and tools needed for the projects it would be great. Additionally, SWE is not entirely web development so having a wider range of projects would be better.
The Projects were challenging. It was a lot of self learning for everything – I think in the future it would be good to go over basic software engineering concepts – such as REST APIs, selecting data sources, storing them, displaying them – how to communicate with frontend and backend – and so on
The grading makes me not want to try after I miss a requirement for an A. I found that I had to learn literally everything to do with software engineering on my own. I thought the lectures would help, but they didn't. The only thing we get related to Swe is the project.
Projects: subject material not covered in class
Lectures were challenging. They were hard to keep up with sometimes and I had trouble connecting what we did in class to the projects. I would have liked to learn more about SWE technologies such as React, in class.
Working with a team on the main class project was frustrating at times, if you end up with a team that can't communicate it can be hard to efficiently succeed in the class
In the initial projects I found it hard to understand what my goals were and what exactly I had to complete. The documentation for the projects is a bit spread out with some things being on ed and some on canvas and some on the project page. I also do not like that the system of makeups is the same whether or not you have a verified emergency.
Could have given more makeups to students with accommodations. Getting treated like everyone else was challenging since I have documented disabilities this would help future student feel heard and perhaps do better in the class. Additionally setting clearer guidelines on how to interact with the TA in terms of the IDB project as it was confusing at first to think of them as a grader or a manager or a stakeholder or how often to meet. Additionally the class spent a lot of time in python, but less on stuff I felt helped the project
I didn't like the calling on people.
I found the papers to be a little tedious, especially since there was a requirement of 4 comments, which made it more about just commenting for the sake of commenting.
I think maybe more time on the exercises would be helpful since I sometimes need a little bit more time on the harder exercises.
Maybe cracking down on people who don’t do work in projects based on the CATME survey results.
I found learning the required tools for the projects to be the most challenging. I think if some of the tools were just introduced in lecture, it might be easier for students to get a grasp on them during the projects.
It was sometimes challenging to figure out the next steps to take for the fullstack project. I feel like material in class, although useful, was not really relevant to the project we had to do.
I think it would be a lot more helpful if we were able to choose the members of our group; I felt that it was challenging to work with people I didnt know because I felt more uncomfortable informing them that they were not being a good team member
At times, I found the project instructions to be very confusing, vague, and disorganized across Ed Discussion, TA announcements, and rubrics, and this made it pretty difficult to stay organized and know all the requirements clearly. It would be nice if they were all consolidated into one place, and if the instructions were clearer and more detailed, and new requirements for each phase were highlighted. Also, it would be nice to learn at least some things related to the projects in class instead of it being purely self–study.
Very little room for "errors": floor grading system keeps everyone stressed and students can barely miss classes at all to earn a good grade. Also, no makeups are allowed for the last 3 weeks for some reason which frustrated me who saved up for a class absence during one of the days in the last 3 weeks of class.
The project phases were difficult since I started learning frontend from scratch
The project was challenging because it used a lot of different frameworks that weren't always covered in class. Maybe having tutorials for parts of the project would be helpful for future students.
Teach some of the software tools rather than going in depth of cool Python trick a programmer can do.
Deadlines were hard to follow, especially in the beginning, and they should be made clearer. The content outlined in the syllabus was far from what we learned in class though. I was expecting to be taught concepts related to full–stack engineering – however I was mainly taught python and SQL syntax throughout the year, and general coding practices and functions to use when working with those languages, which is also useful information – just not what I signed up for.
Time management with the group projects was difficult. It would be nice to have more structured guidelines on how to complete the project, or at least set up the AWS environment.
Time limits on quizzes and exercises were very frustrating and didn't allow me to fully think through questions. I struggle with shorter time limits.

Comment
The class when it came to assignments was very confusing. The Kattis problems that we did felt like interview prep/LeetCode questions, the group projects were about actual software engineering on an actual live project, and the material taught in class felt like an Intro to Python class. Don't get me wrong, I learned a lot, but it really feels like a "jack of all trades, master at none" kind of thing.
The quizzes were challenging.
The most challenging part of the course were the Exercises. I ran out of time on some of them. Also the projects.
N/A
On the flip side of cold calling, sometimes it feels like a drag when someone is called who seems to not have been paying attention. We will have covered something, switch to a new person and it feels like they have no idea what is going on. (But I guess in that vain, it is still good because material gets repeated/reinforced)
The project itself just felt disconnected from the class at times and class felt more like a python course at times.
I found balancing the work on the project to be difficult and inconsistent, and managing expectations for the website was difficult. We ended up choosing an idea that was substantially harder than it perhaps needed to be, and I feel that we were not confronted with the scraping realities early enough. I feel like TAs should be working more directly with groups in developing that initial idea, and provide advice on what the scope may entail before the project gets too deep. I also think that accordingly, changes to scope could be encouraged. My biggest issue with the class was the defaulting to AI, which genuinely introduced challenges. I feel my learning was impacted negatively by a norm to consult AI for everything, particularly in the project environment, where I feel more explicit learning resources could be provided in a way that is not necessarily holding the hand of the student either. Unfortunately I feel like this was reflected in the attitude of the instructor toward certain questions as well. Professor Downing had wonderful insight in general, and I found him pleasant to interact with, but I do not think an instructor's first response to any question should be to ask an AI. I think this deprives students of potentially positive conversations and learning moments, especially when AI is far from 100% reliable for information and understanding.
Overall the semester–long project was the hardest as it had no real connection to the material we were taught. Of course I understand it is near impossible to teach what we were required to do in a short time since it was essentially making a fully–functioning website (which is software engineering); however it did feel like there was no real connection between the two things taught. We learned a lot about the fundamentals of languages like Python, Java & SQL but it rarely came up during the project. I am not exactly sure what can change here but it is a bit jarring to see the project and what we are learning daily.
First is that we can’t use electronic devices during lectures, so I can’t take notes with my iPad, which causes that I don’t have a guide during revision. The project is also very challenging cause the project has nothing to do with what we have learned on lectures and we need to learn everything by ourselves. I think it’s better to learn something related to the projects.
I thought the material in class was not very helpful in terms of knowing software concepts. Most of the work for learning on the projects has to be done on your own.
I found the project to be challenging because a lot of the skills and tools we needed were not covered in class and thus we did not have a basic foundation of understanding to be able to complete the project in a more timely manner. I feel like if there was a section that described how to use the tools and an overview on how to go about the web development of the project, students would have an easier time understanding and also appreciate this class more since I originally thought that this class would be more about CI/CD web development project type class where the class itself focused more on the tools and skills we needed.

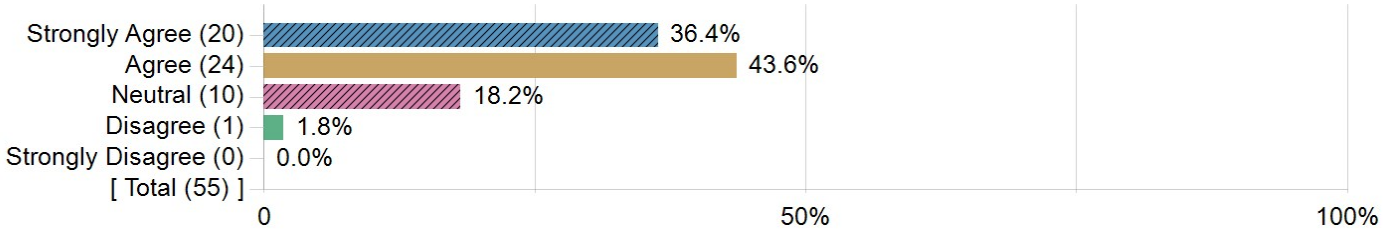
College, School, or Unit Questions

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	2.69

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



Statistics	Value
Mean	4.15

Report Guide

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.