

Individual Instructor Report Spring 2025 Version A for C S 373 - SOFTWARE ENGINEERING (51170) (Glenn Downing)

Project Title: Course Evaluations Spring 2025

Courses Audience: **55** Responses Received: **55** Response Ratio: **100%**

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

Why did you take this course?			
ا ا was required to take this specific course (0) ا chose this course among others to fulfill a degree requirement (46)	0.0%	83	3.6%
I took this as an elective (9)	///////////////////////////////////////	6.4%	
[Total (55)] –	0	50%	100%
	U	50%	100%







C S 373 - SOFTWARE ENGINEERING (51170) Glenn Downing

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

The course assignments improved my ability to express my ideas in writing.			
Strongly Agree (17) – Agree (21) – Neutral (10) – Disagree (2) – 3.6% Strongly Disagree (5) – [Total (55)]	30.9% 38.2% 18.2%	100%	
0	50%	100%	
Statistics		Value	
Mean		3.78	

Instructor Questions

	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree	Responded	Mean
The instructor clearly explained the course objectives and expectations.	54.5%	36.4%	5.5%	1.8%	1.8%	55	4.40
The instructor fostered an inclusive learning environment.	54.5%	29.1%	7.3%	7.3%	1.8%	55	4.27
The instructor effectively explained the concepts and subject matter in this course.	54.5%	25.5%	9.1%	7.3%	3.6%	55	4.20
The instructional techniques kept me engaged in learning.	56.4%	27.3%	3.6%	9.1%	3.6%	55	4.24
The instructor checked for student understanding of the concepts presented in the course.	67.3%	27.3%	1.8%	0.0%	3.6%	55	4.55

Overall Questions

Overall, this course was		
Excellent (18) – Very Good (20) – Satisfactory (11) – Unsatisfactory (3) – 5.5% Very Unsatisfactory (3) – [Total (55)]	32.7% 36.4% 0%	100%
Ŭ	50 /0	100 /0
Statistics		Value
Mean		3.85

Overall, this instructor was				
Excellent (27) Very Good (16) Satisfactory (8) Unsatisfactory (1) Very Unsatisfactory (3) [Total (55)]	.8% 5.5%	29.1%	49.1%	
0		50)%	100%
Statistics				Value
Mean				4.15

College, School, or Unit Questions



The course format (online, hybrid, face-to-face) helped me to learn.			
Strongly Agree (24) Agree (22) Neutral (3) Disagree (3) Strongly Disagree (2) [Total (54)]	5.6% 5.6% 3.7%	44.4% 40.7% 50%	100%
Statistics			Value
Mean			4.17

Comment Questions

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

Comments

I think the lectures and exercises actually helped me understand the concepts taught much better.

Professor Downing explains the concepts to us during class helped with learning.

Professor Downing is extremely knowledgeable about software engineering and is an excellent communicator. His method of cold calling helps ensure students pay attention and I appreciate the no devices policy as it prevents distractions in the class.

I liked the in class exercises and how he cold calls on people to make sure we are paying attention. I also liked the no technology rule, since I tend to get distracted even by other people's devices.

Collaboration on quizzes and exercises were the most effective because they allowed me to work with my classmates to learn better.

Nothing - the course honestly was structured terribly. Nothing project related was taught in class in a timely manner.

The lessons were super helpful in developing a better understanding of a variety of important topics, such as fundamentals of Python, Java, and system design.

I thought it was nice to have an interactive class where people were called.

Professor Downing ensures that the students gain a solid understanding of the subject by keeping students engaged. He emphasizes the importance of the correctness of term usage to avoid confusion. I really like the class

Professor Downing is one of the best CS professors at UT. He cares for his students and has the most fair grading policy out of any class I have ever taken. He does a no technology policy which allows you to fully pay attention to the lectures. He facilitates conversations with students to help us better understand the material.

- The short, time-constrained quizzes tested your understanding of basic chunks of material pretty well

- The projects had just the right amount of constraints on them to make them interesting
- Instructor had clear explanations of concepts
- The papers effectively exposed me to a lot of interesting and crucial CS concepts

In class exercises

Professor's lectures helped me gain a much better understanding of design principles and python/java language features.

The exercises

The notes being posted were always helpful

Emphasized self-learning, but also helpful lectures for topics related to Software Engineering. Collaborative quizzes and exercises in class were very nice to reduce stress on the material.

the exercises and quizzes while many help a lot with learning and deepening understanding.

Our in class meetings and the problems with the participation we had to have made it so that you had to pay attention and be present in class. That almost made it difficult not to learn anything.

I think that the projects were good for collaboration and getting some real experience programming something. I also think that the quizzes/exercises were good for collaboration as well.

I really liked the quizzes and exercises and how working with your class

The notes published after class was effective in my learning.

Quizzes

I thought that the topics that we needed to learn in order to do the projects were good.

The lectures were well-organized and Dr. Downing is quite knowledgeable about the material.

I liked learning about design patterns and how to make better programs. The exercises and papers were very good at teaching us the material and allowing us to think more as software engineers

The exercises were helpful for learning the concepts we discussed in class.

The engaging lectures and the daily quizzes helped to sharpen my understanding of the topics.

The cold calling

The cold–calling, although a bit nerve–wracking at times, is a useful tool that helped in learning. It ensured that people understood the material being covered. The same goes for the exercises.

Projects and quizzes and exercises

The project was admittedly an interesting one, it helped me learn about APIs, various testing frameworks, and CI/CD pipelines, among other things.

Comments

The Quizes and the free canvas project strucutres give us the properly foundaiton in order to become software engineers — which part of it involves finding information online — data, APIs, documentation — etc.

I liked how Professor Downing walked through the code and explained where everything was coming from and why we choose certain implementations over others. The quizzes and exercises being collaborative was also helpful.

The open-endedness of the project.

Well structured

I think that the exercises we did in lectures were very important for my learning because they allowed me to apply what we had learned to real examples.

the daily quizzes made sure I paid attention in class.

I enjoyed the projects a lot – even though they were very self-directed and self-paced, they were a lot of fun and I enjoyed doing them with my group.

I liked the exercises and the lecture policy that no phones or laptops are allowed. I also liked the cold-calling method.

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 found the IDB projects to be challenging because we were kind of on our own in completing them. I think having many resources to get started as well as mentioning things in class that can help us with the projects can help.

The exercises were the most difficult

Our group had only 3 members so the web development process was challenging to complete on time. Balancing the teams slightly better where possible would be helpful.

The group project was quite challenging since we only had 3 people in our group.

I found the project the most challenging due to the nature of having to rely on others in order to complete the task. I don't really have a suggestion to fix this challenge though.

The lack of instruction related to projects.

Dealing with group members for the project...

I wish that there was more emphasis on learning full stack and stuff that we had used in our project.

The project was definitely the most challenging aspect of the class, yet very rewarding. It was a full stack project so challenges are expected.

Nothing is very challenging, but there are quizzes every single class. They are not difficult though.

- The short quizzes were occasionally sloppily made/hard to understand

- Instructor's way of lecturing was sometimes awkward, sometimes honed in on the same point repeatedly without giving clear understanding

- I wish the projects were moreso the focus of the class... we learn so much about Python, SOLID principles, etc. but none of these were enforced on the project

- I wish we more o learned how to properly use Git. This is crucial for good group projects, my partners committed things with terrible commit messages, didn't use PRs, and didn't use issue boards despite these being a soft "requirement" of the project

Learning backend and full stack development on our own

The projects are hard because not everyone does the same amount of work

Group projects because I had to learn new technologies.

By far the exercises, but that was made up by the extra exercises later on in the semester

Very little guidance on the projects, but this helped us figure out how to use the tools / put together the components on our own. Overall class felt fair.

For the most challenging was projects but only due to how much needs to get done. Perhaps more weekly deadlines that meet timeline goals perhaps this could imitate like a actual industry job where you have to meet deadlines.

Having to get a good score on the quiz every day does get a little tiring but it's fine, overall the quizzes are doable and the cooperation helps.

I think that it was hard to focus/see relevance in the lectures, because the lectures felt more like a Python/SQL/Java class rather than a software engineering class. I would've much preferred if he just taught what was in the papers, as those seemed more relevant than what he actually taught.

I really hate the project structure of this class, because we're not taught any of the materials in class. When you ask they professor

Comments

or TA's for guidance the most common answer is to "look online". I am here to learn these technologies if I should just look them up what is the point of this class, so I think the class structure would be way better if the lectures covered java script (react), AWS, and SQL (not just queries but actually connecting the database and using it in a real world scenario). I think learning random python syntax is quite useless, and we just forget them the next week.

TLDR: Please switch the lectures to actually important topics (java script (react), AWS, and SQL (not just queries but actually connecting the database and using it in a real world scenario))

The projects were the most challenging, because some phases were quite some work.

Projects

None of the stuff we needed to know in order to complete the projects were covered in class, or if it was it was only covered far after the project was due.

There is a massive ramp–up to get Project 2 setup from an AWS standpoint. Having longer timeframes for Project 2 and Project 3 would greatly ease development.

I originally entered the course to learn frontend (html/css/js/React), and I aimed to learn it in class. However, it was something that we pretty much had to learn ourselves. Since my project group already had too many people working on the frontend and too few on the backend, I ended up learning basically nothing new about frontend development. I believe the class would benefit from taking away some time from Python and diving into some basics (or even deep into) React or other frameworks, just how it does with SQL.

None of the tools we used were discussed in class. I would prefer that we learn about our tooling rather than random python trivia.

I think the projects were more challenging as we are with random teammates so planning to do the assignment with them wasn't easy. I think being able to choose people on your team or at least some would make it easier so that way it is easier to communicate.

The projects were the most challenging, especially since we used many tools not taught to us in the class.

waking up early

The banned usage of laptops in class feels rather unhelpful. I understand the banning of phones, but laptops are used for notes as well (where people want to write their own) and are also used to search material while it is being covered.

Furthermore, this class seemed to be a simple introduction to Python, Java, SQL rather than software engineering. We did not cover tech stacks that much or JavaScript, which was heavily used in the projects. It felt like students were simply thrown headfirst into expecting to know how to create the website, so covering this material in greater depth would help future students.

The projects were most challenging because there was definitely a learning curve for the different topics that were covered in frontend and backend. This made it challenging. Additionally, the randomized groups were difficult because some members did not contribute as much, and I felt like I had to do more to cover their parts as well.

It feels like there's very little leniency in quizzes and assignments. You need to be 100% on your game and even a little error can ruin your grade without as much chances for improvement as OS.

The msot challenging aspect of this course was definetly the projects since users are to implment a complete tech stack — serverless ckoud environemnt, with CI/CD pipilines, API calls — so stay organized and having delegated responsabilities make the work environment much easier (something that I definetly could had done differently).

I think the most challenging part of the course is the semester long IDB project. I definitely learned a lot of skills from it, but I was unfamiliar with the tech stack, so it took me longer to understand it.

The exercises, due to them being somewhat random, maybe some type of announcement before exercises.

Cold Calling

I don't think there was any part of the course that was too challenging.

the daily quizzes were also the most challenging part of the class, as I had to be sure to be on time.

I think the most challenging thing was probably just keeping track of all the different requirements and making sure that I was fulfilling them for the A. At the end of the day, I think that it was helpful to have these different components because it kept us on our feet.

One aspect that was challenging was the quick 3 minute quizzes that start RIGHT at 10:02, and the grading system where the lowest subcategory is the final grade. What could help is maybe the quiz could be at the end of class? Sometimes my elevators don't work and I have to rush to make sure I don't miss the quiz.