



Individual Instructor Report Spring 2025 Version A for C S 371P - OBJECT-ORIENTED PROGRAMMING (51165) (Glenn Downing)

Project Title: **Course Evaluations Spring 2025**

Courses Audience: **68**

Responses Received: **66**

Response Ratio: **97.1%**

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.

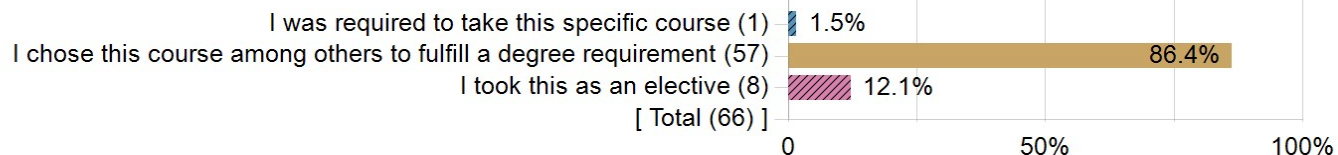
The number of students (e.g. respondents) marking each option is reported for each of the items. These frequency distributions provide information about the level of student ratings and the spread and shape of the class distribution of responses. The distributions thus provide a picture of student perception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

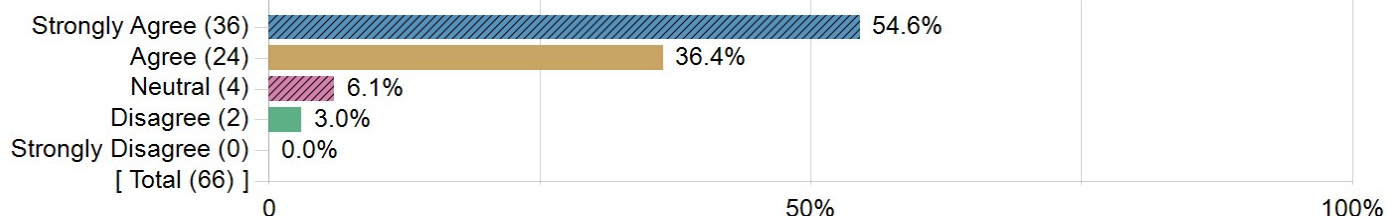
Creation Date: **Friday, May 9, 2025**

Course Questions

Why did you take this course?



During this course, I gained a deeper understanding of the subject matter.



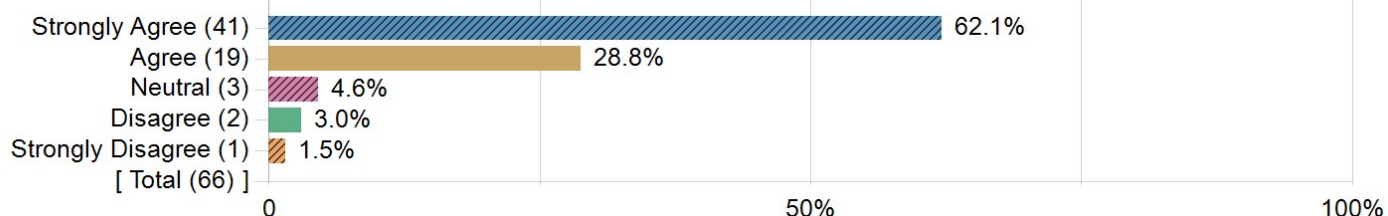
Statistics

Value

Mean

4.42

The course was well organized.



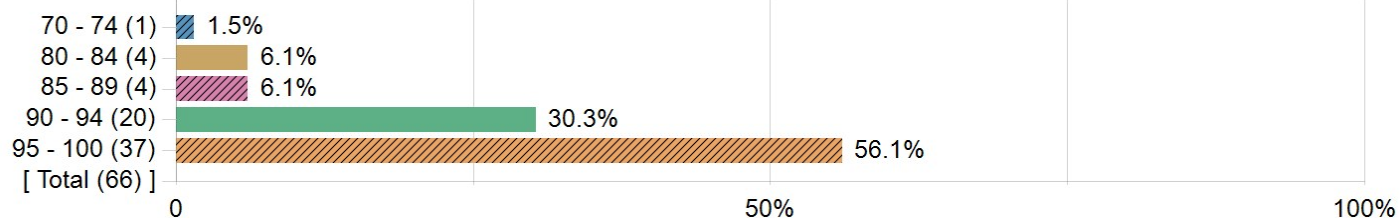
Statistics

Value

Mean

4.47

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



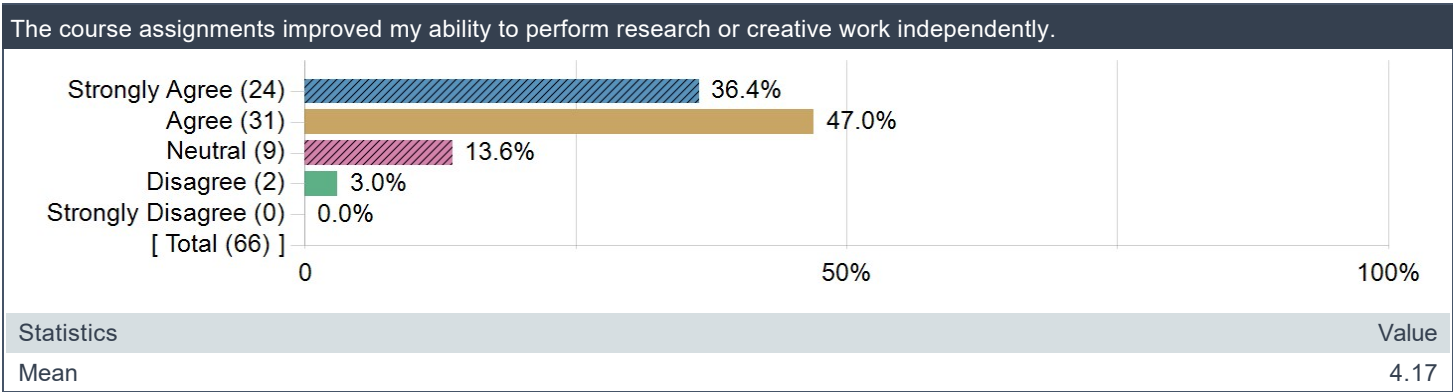
Statistics

Value

Mean

93.26

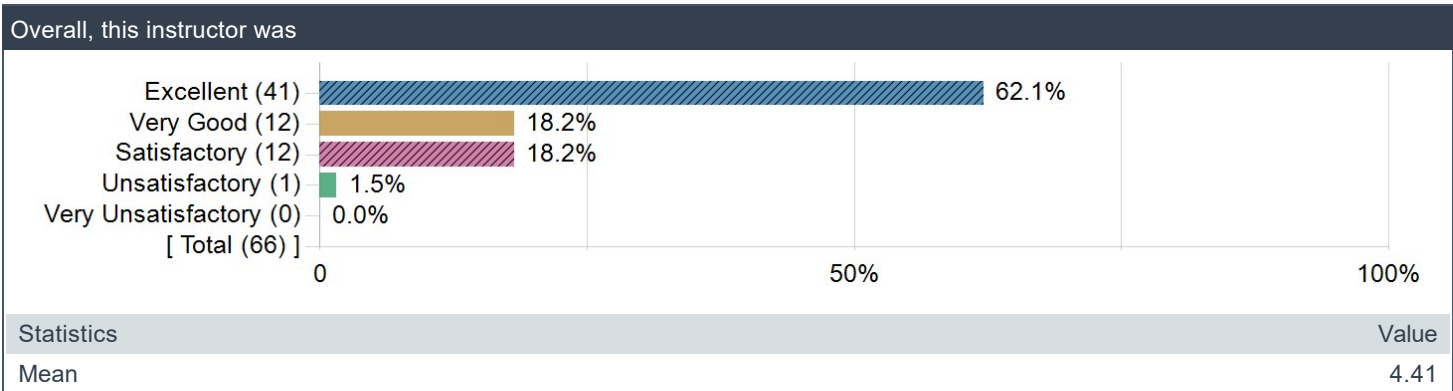
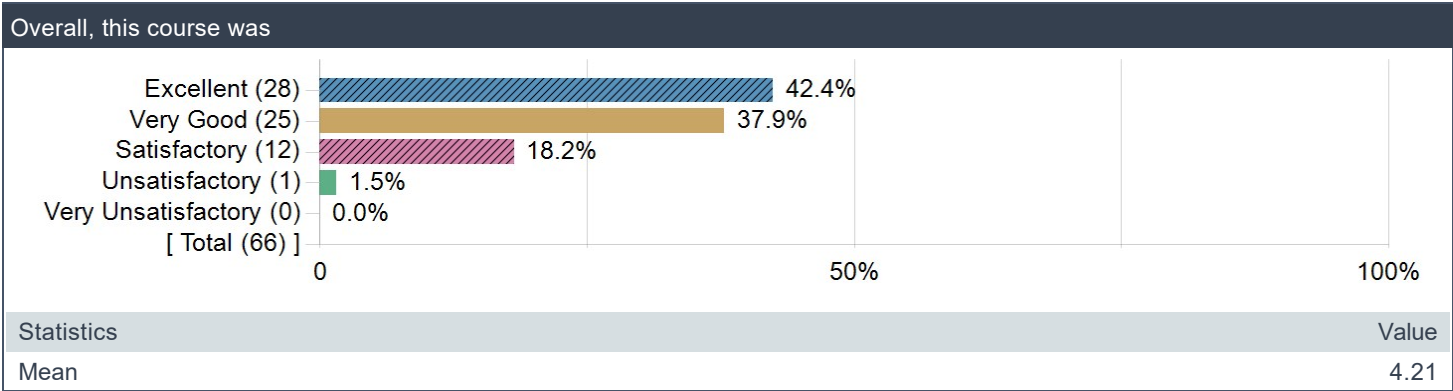
The course assignments improved my ability to perform research or creative work independently. (Flag Question)



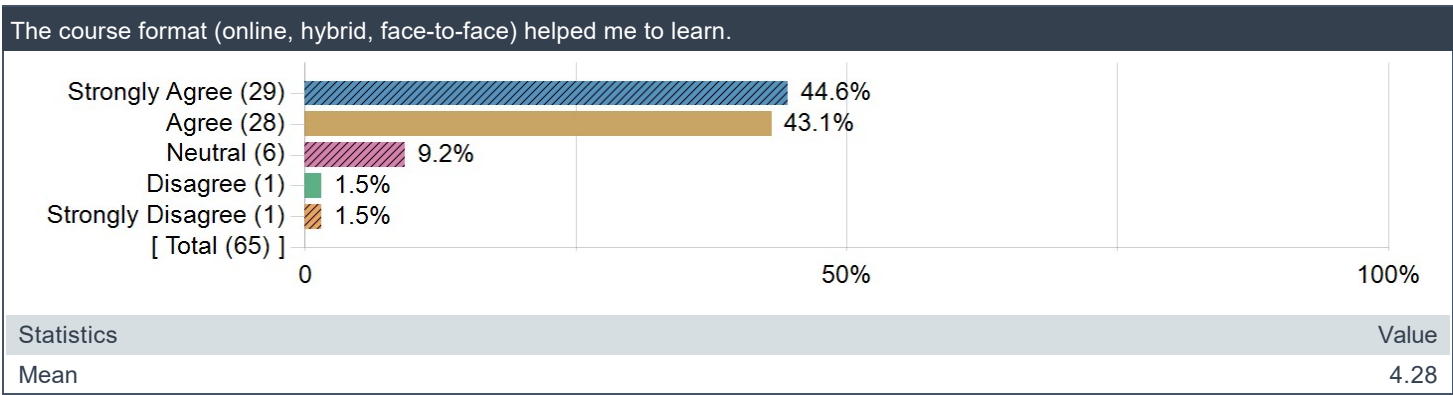
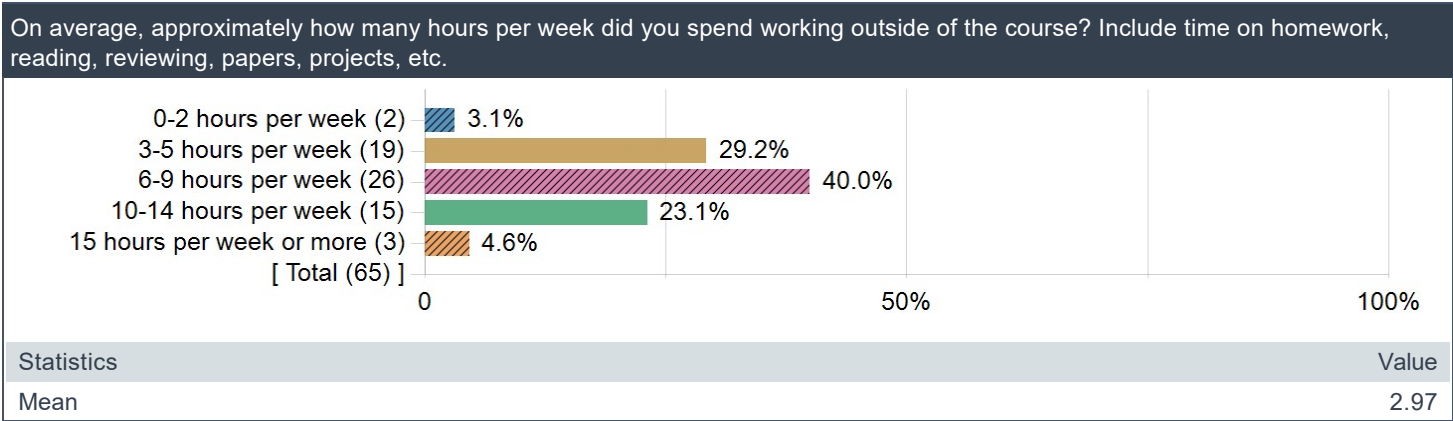
Instructor Questions

	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree	Responded	Mean
The instructor clearly explained the course objectives and expectations.	68.2%	24.2%	4.5%	3.0%	0.0%	66	4.58
The instructor fostered an inclusive learning environment.	54.5%	28.8%	12.1%	4.5%	0.0%	66	4.33
The instructor effectively explained the concepts and subject matter in this course.	51.5%	37.9%	6.1%	4.5%	0.0%	66	4.36
The instructional techniques kept me engaged in learning.	62.1%	27.3%	4.5%	3.0%	3.0%	66	4.42
The instructor checked for student understanding of the concepts presented in the course.	60.6%	28.8%	9.1%	1.5%	0.0%	66	4.48

Overall Questions



College, School, or Unit Questions



Comment Questions

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

Comments
Detailed syllabus and class website. Helpful TAs and professor on Ed.
The strong emphasis on collaboration, in quizzes, exercises, and online discussions.
Teacher was fantastic at breaking down subjects
Professor Downing's lectures are magnificent.
The async coursework
His teaching style of explaining code segments line by line was most effective.
Quizzes
Doing the quizzes
The cold-calling really helped me learn and pay attention in this course.
The collaboration encouraged for all aspects of the class was great at helping me build my understanding through my peers. Additionally, the challenging nature of the content helped to push me more to learn.
Good exercises, cold calls
The projects were very good and the exercises provided some benefit, but have some problems that I will describe later.
This is a programming based class, so Professor Downing mainly walked through code during class. The way he explained the concepts as we went through the code helped me. I got a stronger understanding of C++ than I had with C even if I worked with C before. I also liked how the quizzes and exercises were collaborative.
I think the requirements for the assignments help prepare for the professional world in which you need to do a lot of testing and developing.
Prof Downing was extremely helpful in making sure that we understood the material, and would cold call us. Although I didn't initially like the cold calling, it made me pay attention in class, and as a result, I learned more material.
Quizzes/Exercises
Quizzes during every lecture forced me to be up to date, while some were tough, I overall felt it positively contributed to my learning even though i really do not like :(it the daily quizzes and cold calling. also the projects
I liked the exercise and the in class quizzes as that forced us to attend class in person
I think having to show up to class, as well as do makeups in case you did not attend(which 'forced' you to watch the lecture online) were very helpful in keeping me engaged, alongside the no devices policy. Although I did take notes on my Ipad
His lectures were really good, they explain the topics well and really helped me learn. The papers were also all pretty interesting, and I believe I'm a better programmer thanks to them.
I think the lectures and the projects were really effective in helping my learning
Cold calling as it would make me more engaged into the lecture.
I thought that the class lectures were mostly engaging and aided in my learning.
I liked when you walked through the projects in class.
Although it was tedious managing and creating the repo for each project is a useful skill to learn
The cold calling and no electronics definitely made me more attentive in class.
Repeated quizzes, projects that were done
I liked his energy, and the cold calling helped me stay focused
Having access to online lectures, as well as the detailed cases for various syntactical details in C++ that we learned.
collaborative quizzes, exercises
Staying on my toes for cold calling
The micro-quizzes were helpful. The projects were helpful, but got tedious at times.
Class notes, concise lectures
Cold-calling, and just the lectures in general.
Very good lecturer, I enjoyed learning about the nuances of OOP.
The exercises really allowed me to get practical experience with what we were learning.
The course website was solid

Comments
In class participation, pair programming projects
The quizzes reinforced what I learned during the previous class. I liked getting to collaborate with the people around me.
I think the projects and the exercises were helpful in learning the concepts we discussed in class.
Ability to maintain engagement with the class
Cold calling made me stay more focused in class as well as not being allowed to have devices open during lecture. In class exercises helped me learn and use my knowledge to problem solve, and the projects outside of class gave me great opportunity to practice all that I have learned as well as apply my learning in different ways.
I think cold calling is a very effective way in ensuring that people pay attention in class. I know some people aren't a fan of it, but for me it made me put more effort in paying attention during lecture because I didn't want to look stupid if I got called on and I didn't know the answer. I also think the daily quizzes were a good way to force me to review the previous class' material when usually I tend to not review lectures until studying for an exam.
The daily quizzes and assignments
Posting the notes after class.
I think the grading rubric, although a little annoying, was helpful in ensuring I learned the material of the course. Having to meet a threshold in all categories in order to achieve a certain letter grade ensured that I tried my best on every assignment, which in turn led to me learning the material better. The pacing of the class is also good, doesn't feel too rushed or slow.
The exercises, even if they were stressful, helped me get hands-on practice of the topics that we discussed in the task, and helped keep me engaged with the topics.
Quizzes every day to check understanding, the projects were really useful to learn more hands-on C++
Projects and exercises

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
Not much was challenging, I found that you get the grade that reflects your effort.
Slightly challenging to receive help across the week, with office hours clustered on Tuesdays and Wednesdays.
Not really challenging, well organized
Cold calling was challenging because it was hard to answer on the spot, especially when I didn't fully understand
Most likely the projects, just go to help hours as needed.
Projects lack rubric
Getting cold called on. It was nerve racking sometimes and it seems that he got annoyed occasionally which added to the stress.
The exercises were the most challenging part of this course for me. It often felt as though there was not nearly enough time later in the semester. I think maybe just extending the time based on how much we need to implement in an exercise.
Timed exercises felt very restrictive and discouraging at times, because I felt I needed more time to understand the code.
Quizzes wee pretty complex and hard to understand maybe due to complexity of the subject,
I think exercises were rather tricky since many times it required a lot of initial writing and debugging with a somewhat limited understanding of the actual assignment. I feel as if some more guidance with the exercises and more live feedback on how your input compares to the expected results would help significantly more since I feel sometimes my code becomes hacky just to satisfy a requirement.
The projects were the most challenging. I think it just took me some time to complete them, but we were given adequate help during class and office hours, so I think it was just me and my time management.
I found the exercises quite challenging because a lot of the time it was material that we had just gone over and suddenly we're expected to a quick exercise on it. It just is a me personal thing that I need more time to think on the subject and the exercises made me feel very overwhelmed. I think either more time or leaving the exercises open until the end of the day would have been nicer and given me more time to really learn what I was coding about.
One of the more challenging aspects of the course was the fac that Prof Downing's grading was very nit picky, and you could lose a lot of points over simple mistakes. The grading system could be frustrating at times, and the pop exercises were not well structured.
Hard to follow along since we couldn't take notes
Some quizzes were tricky, and asked about topics we had not finished covered
the projects. some requirements are not specified in the specs such as not using friend class. also small bugs that were my fault

Comments
like not reserving for vector of pointers. i would suggest students to really understand c++ syntax before jumping the gun and writing code like i did.
I thought some of the content is not really relevant or needed. The only OOP part was the papers and I thought that was misleading about the course
Professor Downing required us to completely uproot what we learned in CS314 about object-oriented design. Maybe for the better, maybe for the worse. But it was a difficult transition. I think something that could help would be to give a bit more examples on how to think of a solution that fits the requirements of his class better. It's hard to completely change how you program for just one class.
I think I found some of the exercises challenging, and in order to do well in them you had to be well versed in what happened in the previous lecture, alongside being quite focused towards the beginning of the class. I am appreciative that there are additional exercises towards the end so that my grade does not drop though thank you!
What I found the most challenging was the grading system. In my CS classes I was used to letting somethings that are usually worth little on the grading scale (class activities, quizzes, discussion work, etc) fall to the side while focusing fully on the things that were graded with more weight (exams and projects) and ended up with good grades. However, because of how this class is graded I couldn't do that. I had to make sure to attend every lecture (or as many as I could) and not miss one exercise, paper or blog. This was really challenging and I didn't manage to get as good of a grade as I would've liked
The exercises are pretty tough, but also very doable. Some exercises in particular are hard to complete within the 20 minute time frame, and I feel like 25–30 minutes would be better.
I found the most challenging aspect of the course to be the in-class exercises. I would suggest that future students be allowed to look at code from previous or same lecture to help with the exercise. It seems to be a more of a memory practice than functional when we can't.
I found that the project directions and getting started was difficult. For example when adjusting the makefile for a new project and then having to find and replace key words was difficult and hard to do since there was not much direction.
Weekly blogs and reading the perusal papers were something that I found more agitating than enriching.
I think the exercises were a lot more challenging compared to SWE but as long as you pay attention in class and learn the material you should be fine.
I found keeping up and staying organized to be the most difficult
Probably managing it with OS. If I would've taken this class without OS, I would've learned a ton more
I found the daily quizzes the most challenging part, as at times they seemed to be very complex in relation to the amount of time that we were given for them. One thing future students can do to meet this is collaborating with peers, as well as reviewing previous lecture videos, which I didn't realize we had until midway through the semester.
cold calling, but increased engagement in general
Exercises worried me a little but extras and makeups helped
Unfortunately, I had to deal with personal circumstances. There's not as much leeway in terms of the exercises. Also, with them being completely random, you could miss a week and get docked 2 exercises or 0 exercises. This is something that could help future students.
I was always pressed for time on the exercises. How about giving students 2 minutes to just read the instructions and provided code, then start the timer to write code?
Exercises were challenging, simply because of the time crunch we were under. However, that is just the nature of it, and I feel like the grading system was lenient enough to still ensure you got an A.
I think the grading scheme was hard to get used to at first, but I got used to it.
I thought the last two projects were harder compared to the first three projects that we did. Also, the exercises are pretty challenging, especially under a time limit, so it's good to collaborate with people around you.
N/A
The EMRN grading scale is pretty challenging to deal with. I did well in mostly everything, but once I got docked off one mistake for a project, I couldn't really recover.
I found some of the later projects to be challenging because of the lack of clarity on what was allowed. I think we needed more context behind not using getters and setters, and the specs for those projects should be clearer.
I did not like the attendance scheme since being physically in-person for class versus watching the lecture later does not provide much of a difference in experience for me.
N/A
In class exercises were the most challenging due to the freshness of the knowledge we were applying and the time limit. I would suggest having more time, but let's be honest adding more time may still not be enough because you only really start to feel the stress of finishing the exercise when the clock starts really running out.

Comments
The most challenging aspects of the course were the programming assignments.
The assignments
I think the challenging part is exercise because we are under a time limit. But thankfully, we can collaborate with other students.
The projects, but they weren't too bad and I feel like they gave me some 'experience' with the workflow I will encounter once I graduate.
I found the most challenging was the actual projects like teh memory one, mainly because there is no wiggle room in the grading, but I still enjoyed learning about the projects and solving them.
Not sure, I really liked pretty much everything about the class. Maybe not forcing a project to be with a partner
Name calling and random exercises