Individual Instructor Report Fall 2023 Version A for C S 371P - OBJECT-ORIENTED PROGRAMMING (52975) (Glenn Downing)

Project Title: **Course Evaluations Fall 2023**

Courses Audience: **72**
Responses Received: **64**
Response Ratio: **88.9%**

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.

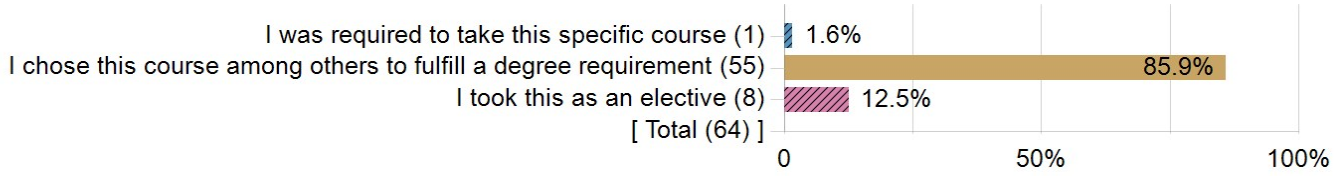
The number of students (e.g. respondents) marking each option is reported for each of the items. These frequency distributions provide information about the level of student ratings and the spread and shape of the class distribution of responses. The distributions thus provide a picture of student perception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

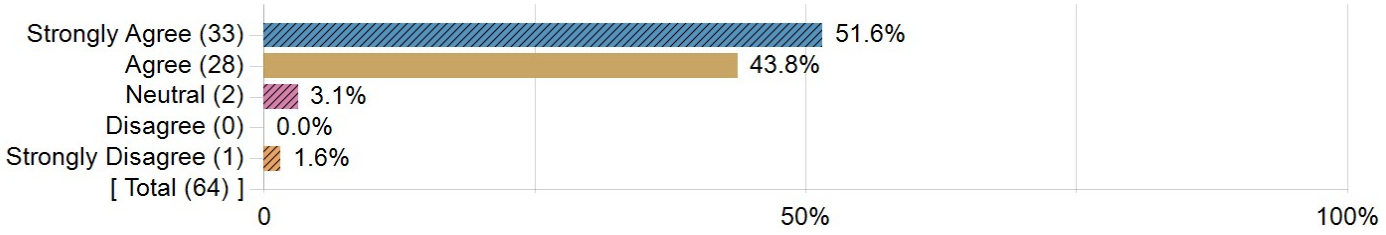
Creation Date: **Monday, December 18, 2023**

Course Questions

Why did you take this course?

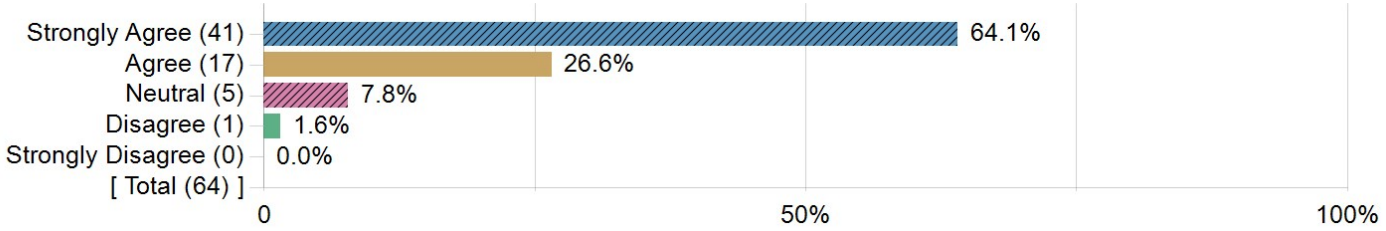


During this course, I gained a deeper understanding of the subject matter.



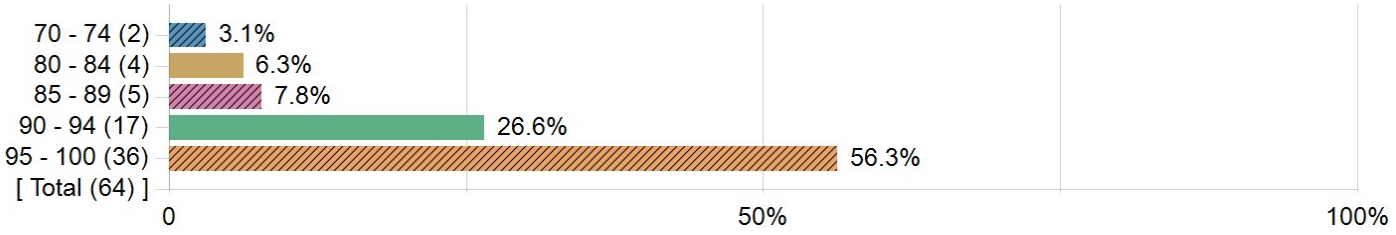
Statistics	Value
Mean	4.44

The course was well organized.



Statistics	Value
Mean	4.53

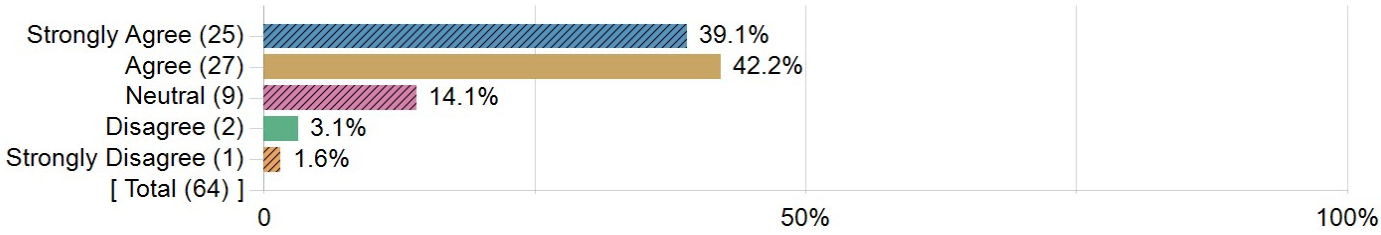
Overall, approximately what percentage of the course meetings did you attend or complete (online, in person, or asynchronously)?



Statistics	Value
Mean	92.70

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

The course assignments improved my ability to perform research or creative work independently.



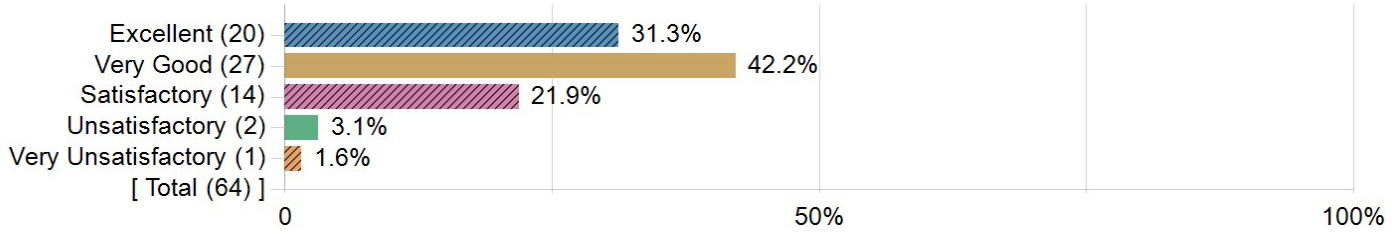
Statistics	Value
Mean	4.14

Instructor Questions

	Strongly Agree	Agree	Neutral	Disagree	Strongly Disagree	Responded	Mean
The instructor clearly explained the course objectives and expectations.	65.6%	32.8%	0.0%	1.6%	0.0%	64	4.63
The instructor fostered an inclusive learning environment.	59.4%	35.9%	1.6%	3.1%	0.0%	64	4.52
The instructor effectively explained the concepts and subject matter in this course.	56.3%	37.5%	6.3%	0.0%	0.0%	64	4.50
The instructional techniques kept me engaged in learning.	53.1%	29.7%	12.5%	3.1%	1.6%	64	4.30
The instructor checked for student understanding of the concepts presented in the course.	60.9%	32.8%	6.3%	0.0%	0.0%	64	4.55

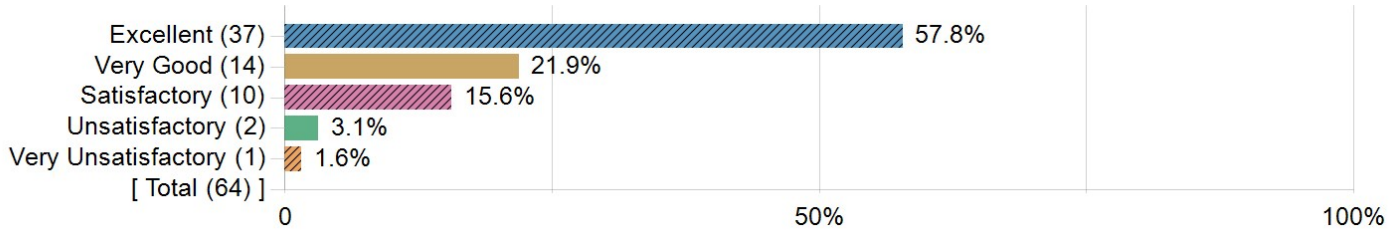
Overall Questions

Overall, this course was



Statistics	Value
Mean	3.98

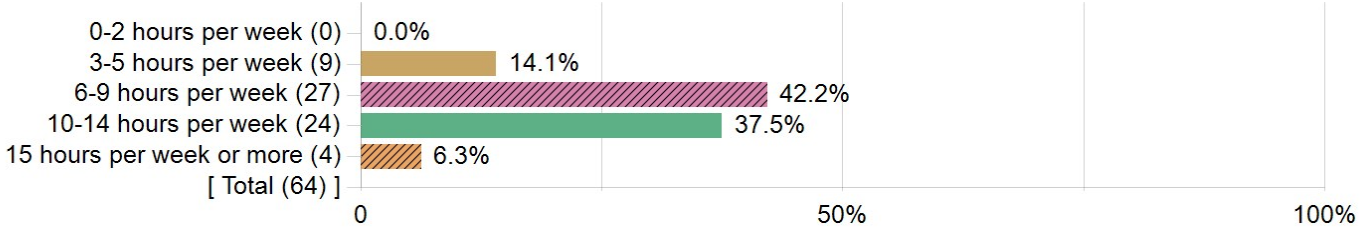
Overall, this instructor was



Statistics	Value
Mean	4.31

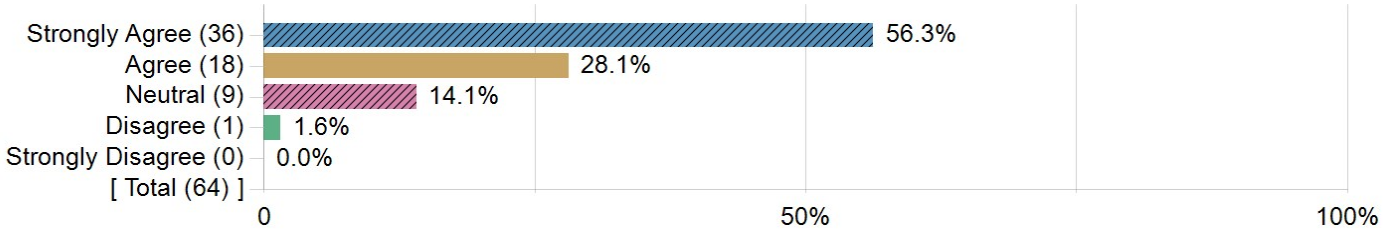
College, School, or Unit Questions

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
Invited Count	72
Response Count	64
Response Ratio	88.9%
Mean	3.36
Median	3.00
Semi-Interquartile Range	0.50
Mode	3
50th Percentile	3.00
Sum Total	0.00
Standard Deviation	0.80
Population Standard Deviation	0.80
Standard Error (base on SD)	0.10
Standard Error (base on PSD)	0.10
Aggregate Frequency 1	0.0%

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



Statistics	Value
Mean	4.39

Comment Questions

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

Comments
The lectures were very engaging and we knew what to expect. I really enjoyed Professor Downings lecture style and appreciated the cold calls.
good TAs
The lectures
In class examples.
Student help on Ed discussion
Professor Downing is very thorough when going through class concepts which is useful because he covers edge cases and "what ifs." I think him going over many different possibilities of how a certain type of code can be written and what they would result in was very helpful and not something I experience in most other classes.
The lectures were informative but repetitive. It was hard to stay focused for extended periods of time. The cold calling was very anxiety–inducing, especially because students were treated poorly.
The most effective thing was the ed discussion where people asked questions. It was helpful to see people run into the same problems I was having so that helped me finish my projects.
The content regarding lvalues, rvalues, copy assignment/constructor, and iterators were very thoroughly explained.
instructor and ta willingness to answer questions both in person and asynchronously
Projects were thought–provoking and interesting. Loved the papers.
Although I wasn't always a fan of the quizzes, they forced me to review each lecture's material right before the next class. The cold calling also helped me stay engaged during the lectures.
The projects were definitely the most helpful. I also found the papers useful in understanding OOP concepts.
I think the most effective aspects were the cold–calling in lectures because they helped me stay focused and pay attention to what was going on. Also, I think that the daily quizzes were a good way of recapping material from previous lectures. I also liked the exercises because we were forced to work in pairs, and I could often learn something new about coding from my partner. Lastly, the recorded lectures were super helpful in reviewing material I had missed or didn't completely understand.
I really enjoyed the projects. It allowed me to have a deeper understanding of C++ and Object Oriented programming in general
The quizzes and going over them helped with understanding the concepts. The exercises were also good at cementing understanding.
I think the daily quizzes were most helpful for my learning. It kept me accountable to understanding and reviewing the material constantly, and the quizzes did not stress me out as much as I think an exam would as there are many opportunities to improve your grade.
I think the cold calling aspect as well as the daily in class quizzes ensured that students were prepared when they came to class.
I would say the cold calling ensured that I stayed engaged in lecture.
I really enjoyed doing the projects after learning different implementations and strategies in class.
The lectures are very helpful, clear, and informative. ED also helped a lot for parts of assignments I was confused about.
Recorded lectures along with the notes being on Git Lab
I thought the projects were useful in gaining broad software engineering skillsets and industry–standard tooling. I learned how to debug using asserts, write unit tests, use docker, CI/CD, and best version control practices.
The projects helped me develop a much better understanding of C++.
I think the projects were the most effective in me learning things that were new to me. I was able to apply what we went over in class in the projects that help lock in the information and set me up for a better understanding.
The cold calling
Projects helped me implement course material on my own
The projects and quizzes were helpful to reinforce knowledge
The projects.
The cold calling kept me engaged and helped me deal with my issue of public speaking. The exercises, quizzes, and projects I feel were all well done and reflected what we were learning in class.
The various lectures that covered a wide variety of topics related to OOD were most effective in my learning.
I liked the projects.

Comments
Having a short quiz at the start of class, and especially going over it afterwards, was a nice way to refresh what was being taught in a bullet–point format.
I think that the lectures and projects were effective in helping my learning. The lectures were in depth in C++ and were clear and concise.
I really enjoyed going over project specifications in class and breaking down how you would go approaching a problem and its optimal solution.
Daily quizzes were very effective for gauging how well I was learning the course material at a given time. They were highly useful in checking if I was on track for that week.
Posting notes online
Having the discussion board and the interactiveness of the lecture was very helpful. People often asked questions on the Ed which helped me gain a better understanding of the assignments. The interactiveness of the class also keep me more enganged than traditional lectures.
I learned the most doing the projects!
The team projects were great.
the quizzes helped reinforce my learning
Each class was very structured and similar in structure, so that helped me catch up on notes. In addition, the cold calling made me pay attention throughout the class.
Ed was very helpful for implementing the projects.
The aspects of the course that were the most effective in helping my learning were the homework assignments where we got to apply the concepts we learned about in class.
Quizzes
How professor Downing conducted his lectures with the cold calling and the projects aligning to the lecture material.
The projects and lectures were effective in demonstrating the topics we need to learn and comprehend.
Professor Downing was a great lecturer and made it to where the content was a lot easier to understand.
The lectures were engaging for the most part. Not the content, necessarily, but I almost always found Professor Downing's manner of lecturing pleasant. The assignments were good for applying OOP concepts. The quizzes were mostly helpful, although some questions were just far too specific. They could sometimes feel like trick questions with how subtle the nuance of syntax is.
Conversation style lectures. I also was hesitant at first but liked the fact that we had to change partners for the projects.
Daily quizzes, lectures were detailed and helped with starting the projects
Cold calling
Downing is a very energetic lecturer and I personally like that he cold calls on students. It was very helpful that the course recently added the 2 Es replace 1 R policy. The projects are interesting, I'm glad that the class encourages us to work with different partners, and Downing and the TAs offer a lot of assistance on projects outside of class if you attend office hours. The course is very well organized –– all notes and exercise solutions are posted on the class website, and the grading system is very well defined from the start.
Doing the projects was very helpful in putting the concepts we learned in class into practice. I also got a lot out of doing the reading assignments, specifically the ones about SOLID and why getters, setters, and extends were bad.

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
Sometimes audio was cut out from lecture recordings or the professor didn't upload notes, I need to study notes better
The projects and the quizzes at the beginning
In class exercises. Time too short and not enough TAs to help the number of students when issues arise.
Projects; not much guidance and you have to do well on all to get an A. Also the grading scheme was stressful. I suggest more lenient grading scheme.
I think the in class exercises were very challenging. I feel like we did not learn all that was required to be successful in the exercises. There were also not enough TAs to help whenever people ran into an issue which added to the stress of doing the exercises because the time limit was quite short.
Cold calling and daily quizzes seemed kind of excessive to me. The daily quizzes also covered extremely specific details, which didn't seem fair at times.

Comments
The most challenging was figuring out some optimizations or edge cases that needed to be covered through running other people's acceptance tests. We could have more hints maybe but this is slowly figured out through the ed discussion.
I would have appreciated some sample practice questions for the quizzes, maybe past quiz questions.
lack of detail in project and exercise specs
Some of the run time complexity aspects of the projects were redundant and focused on things that weren't super relevant to our class. We also didnt really go over the papers in class, which I think is a shame.
The projects were the most difficult aspect of the course but I felt as if the professor did everything within his power to make them doable.
I thought the quizzes were the most challenging part, but they were a good difficulty (i.e. I don't think they should be any easier). This class was helpful for me in learning C++ since I didn't know it very well. However, we did not cover as many OOP concepts as I would have hoped for. I think it might have been better to spend less time on the details of C++ and more time on teaching OOP concepts in class.
One aspect that was challenging was the grading system. This led to a lot of stress because I needed to get perfect scores on every single lab in order to have a chance at an A. I know this is done to encourage students to do all the labs, but perhaps there could be a better way of doing this since I'm not sure asking every student to fully complete every single lab for an A is reasonable.
The vector exercises were definitely the hardest. At times it felt like we were being thrown into the deep end regarding some of that material. I would have enjoyed some other exercises about constructors to warm up to that kind of implementation.
The quizzes were challenging in the fact that I had to get here quickly from my previous class across campus. The exercises were also challenging as I felt more time should've been given for some exercises.
I think that the most challenging part were the exercises. The exercises were intersting at times, but more often than not, time was spent trying to find the correct sysntax for functions, and I don't feel like they were explained well beforehand.
One aspect I found a little challenging was the grading system for the projects. Missing one thing from the project meant that you would not get credit at all for the work put in which I think is unfair. This could be changed to allow a more continuous grading scale rather than discrete.
Exercises have too many minor things that can go wrong and so much to keep track of in not enough time with not enough TA's to help.
I think the assignments were the most challenging part of the course. I did, however, choose to do most of the assignments by myself. Had I worked with a partner on the rest it definitely would have been easier to do, although I do not regret working solo. I think one thing to help students complete the assignments better would be to mandate working with a partner, even though I know that is not preferable for some people.
Some of the exercises were pretty difficult, some of them were even on stuff we havent learned yet. Please make sure that the content is taught in previous classes before doing an exercise (ex. Vector4) (I will admit, it was taught at the last minute of class but the lecture video was cut off so it was impossible to remember.)
The in class activities were too long for the amount of time given to do them. I failed to turn in multiple exercises simply because I needed another 5–10 minutes. I didn't have time to consider a lot of quiz questions thoroughly because just the time to read and comprehend what was being asked took too long The time would be fine if the activities were shorter. Or the activities would be fine with more time.
I found projects really challenging and kind of hard to grasp because I did not have any exposure to C++ before hand, and the work flow was difficult because I had never done anything quite like it.
I thought this class had too heavy of an emphasis on C++ syntax and details. When I first signed up for the class, I was under the assumption we would focus on core OOP concepts like abstraction, polymorphism, encapsulation, and inheritance. Similarly, I thought we would learn about design patterns. While we were assigned papers covering some of these topics, it would have been nice to have covered this in class. Also, we spent way too much time with C++ minutiae, especially with the vector example. Now, I learned valuable skills from the projects and general C++ knowledge, but I don't feel I am any better off regarding object–oriented design.
The daily quizzes felt like they were designed to gatekeep A's from most students. Often times they felt disconnected from previous lectures, and/or contained extremely tricky questions that were designed for more students to get incorrect.
I found the projects the most challenging especially near the end where the content got a bit heavier. I think this was very important to my learning, and the best that could be done to meet the challenge more effectively is to start early and ask questions.
The projects
Quizzes were hard to keep up with
I thought the last 2 projects were particularly challenging because the rubrics were not very specific regarding implementation specifics to follow. I would add more detail for what class is not allowed to handle what information, as it's not apparent from just

Comments
the instructions.
Class attendance quizzes were very specific, although that is't necessarily a bad thing. The notes were generally helpful.
I wish there was more partial credit for the exercises, as I feel that a student can get one 99% correct and still get a 0 for it. Even a 1 goes a long way with the current grading scheme. The quizzes are also not the best in my opinion. Getting a b or c even if you have all the other requirements for a higher grade because you got 30 instead of 31 quizzes is a bit harsh in my opinion, although I really do appreciate the fact that two 3's make up a one.
I thought the in–class exercises were most challenging, as they required a complete understanding of the material, so it was easy to get stumped on some of them. To help students, chances to retake these exercises could be offered; this will ensure that students fully understand the topics before moving on to harder exercises.
The hackerranks were the most challenging. The time limit could be increased to the end of the day.
Having the new points scoring method was disheartening, as it meant that even if a small part of your code was wrong then you wouldn't be awarded anything. The use of code tidbits instead of actual slides also made it much harder to study and review previous lectures.
I think that the most challenging part of this course has been finding people to collaborate with on the assignments. There isn't really a good way to find partners besides meeting people from class. There are the ed discussion posts but that is a bit of a gamble to be honest. As a whole however, this class is easily one of the better classes offered at UT.
I found some of the exercises to be very challenging;. Most of them felt like they required a lot more knowledge than anything we had learned in class. Maybe go a little more in depth of the concept before we start exercises.
While I found the daily quizzes useful, they have also been the most challenging aspect of the course. Not only are they at times rather difficult, they can be a bit unforgiving if you miss too many classes as they can easily result in a much lower final grade for the course than initially expected.
Not covering material mentioned in syllabus
The projects were definitely the most challenging part of the class. Some were more difficult than others. I think the difficults parts were understanding how to build the classes for the later projects. However, I think it was a good experience to struggle. I wouldn't suggest to change anything.
Exercises are the most stressful part of the spec grading system for me since partial credit for them is uncommon.
The calling on people made it hard to hear and follow certain topics.
the projects were the most challenging. in the future, it would be nice if more class time was spent explaining how to approach later projects or designing out how to implement complicated parts of the project as a class.
Nothing was super challenging, except for a part of the projects.
The projects were the most challenging, but I think that what is given in the project specs and through lectures is more than enough to understand and implement the projects.
I guess the aspect of the course that I found the most challenging were the last two projects because of the rules of no getters and setters. I think this rule was particularly confusing because we were told about it, but I don't remember learning about what other valid methods exist for getting information about other classes that exist. I think spending a class period talking about the evil of getters and setters and alternatives to them instead of just having us read a paper on it would be very useful.
GitLab setup and issues
The in–class activities were the most challenging part of the class because of the time constraint and them having a significant impact on your grade and the seeming low amount of room for error.
The quizzes and projects were challenging at times due to their scope and how we were thrown in without being given help at times.
Make the projects easier or give more skeleton code to help the students get an understanding of the project's base.
I feel that in–class exercises could be more lax with time. Sometimes 20 minutes wasn't enough for how complex or specific the exercise was.
More office hours would be helpful for projects, more guidance on projects
The constant quiz taking. Challenging to find time to study.
Specs grading is my biggest gripe about this class, I don't think it furthers students' learning to be so stressed about meeting the cutoff for each category. I actually remember reading a student's blog post who said that he had given up in the class since he missed a project and didn't need to worry about blogs, quizzes, and exercises anymore. Another issue with OOP is the HackerRank exercises. 15 minutes is not enough time to figure out the exercises and feels more appropriate for a competitive programming class than for OOP. Especially since a lot of times it's little C++ syntax issues that screw you up.

Comments
Also, in class I wish we discussed more general object–oriented principles instead of tips/tricks with C++ syntax. A lot of the papers we're assigned to read would make interesting lectures, if we were to go over some of the examples in the papers.
The in–class exercises were often very difficult to finish in the time allotted. Additionally, I had a tough time getting any partial credit for those (i.e.,I either got a 0 or 3) because the most common issue was that my program would not compile. I think that having your exercise not compiling on HackerRank should still be able to give you a 1 if you were almost there overall and only had a few minor bugs.