CMSC 14200 4 - Introduction to Computer Science II - Instructor(s): Borja Sotomayor Basilio, Jesus Almaraz-Argueta

Project Title: **College Course Feedback - Winter 2024**

Number Enrolled: **59**
Number of Responses: **32**

Report Comments

Opinions expressed in these evaluations are those of students enrolled in the specific course and do not represent the University.

Creation Date: **Thursday, March 28, 2024**

What are the most important things that you learned in this course? Please reflect on the knowledge and skills you gained.

Comments
data structures, some algorithms, software design
Object–oriented Programming, Trees, Graphs, Functional Programming, and Software Development
I learned a lot of interesting new concepts that are relevant to my major, and would've been difficult for me to learn on my own
Knowledge of data structures, functional programming concepts, and many software development practices and paradigms. Software development skills are probably the most useful as I've seen a massive improvement in project organization following the frameworks I've learned.
Type annotations, tree rotations, software development basics, pandas, NumPy, git, working in groups.
I learn more programming strategies. Functional programming, coding in groups, using third party libraries.
I don't know. Compared to 141, which I thoroughly enjoyed and was motivated in, the actual material that I've taken away from 142 is probably 10% of the amount I learned in 141.
I learned about graphs, different types of traversals, different types of trees, and how to perform operations on these trees. I learned about inheritance and built off of the previously learned knowledge about object–oriented programming 141. Overall, was an extension of 141's content.
I learned a ton about the data structures and algorithms used in basic programming. I also learned a lot about software development in general
Typing, data structures, algorithms, software development principles
coding in python and fundamental cs concepts
Type attributions, graphs, what it looks like to program in the real world
Data structures, software development
Basic graph theory and some more concepts about trees
Good amount of graphs and data structures.
data structures, efficient algorithms, python in general honestly
More trees, graphs, MinMax, software development, Panda, numpy and click libraries
data structures including self–balancing trees, expression trees, binary trees, AVL, graph traversing algorithms (DFS and BFS), introduction to software development, introduction to Numpy and Pandas libraries
Trees, Object–Oriented Programming, Graphs, Software Design (eg pygame).
Ability to read libraries, basic python, knowledge of graphs and graph algorithms, trees, etc.
I think the most important thing I learned was graph traversal algorithms.
Python graphs, trees, a little bit on queues, traversals (bfs, dfs), Dijkstra's, basic software programming principles, how to work in a team, github, pytest, pygame, python libraries, type checking, debugging
Learned various data structures and algorithms such as trees, graphs, and searching. Developed skills in working with groups on projects as well as software development.
Basic algorithms – dfs, bfs, AVL trees, etc.
Organization
More advanced data structures, Abstract base classes, inheritance, BFS, DFS, Graph, Dijkstra's algorithm

Describe how aspects of this course (lectures, discussions, labs, assignments, etc.) contributed to your learning.

Comments
lectures were very basic, hw was fair / a decent application of what we learned
The homework and in–class examples contributed the most.
The assignments provided a lot of practice, however since we had such a short quarter, the professor couldn't cover almost everything that we had to learn so sometimes the assignments involved a lot of learning on my own instead of the teacher providing us the knowledge we needed
The weekly homework is very good, and I like that it's partially autograded so I can instantly see and correct any mistakes I made. Lectures are great, especially my professor. The final project has probably been a positive experience in the long term, but it's also been very confusing to figure out what we're even supposed to do. I've found discussions pretty pointless. It's basically only 20 minutes of instruction that could easily be covered in class or through video and then the opportunity to ask questions. And I had to stay on campus until 6:30 just for those 20 minutes. I think the discussion sessions should just be Q&A with the TA.
Courses were interesting, notes even better.
Lectures were helpful until half way the quarter
I think reading the class notes and watching a youtube video on djikstra's algorithm are the only things that taught me this quarter.
Lectures were extremely helpful, and this is in no small part due to Professor Sotomayor, who made the content very accessible, explained it very thoroughly, and made lectures very engaging.
The lectures and assignments were really helpful for my overall learning. The discussions were also helpful for big picture things as well as fixing small errors.
Professor Borja's lectures were really engaging and useful for understanding concepts. Discussions helped with the homework.
homework and projects were most helpful
Lectures were extremely helpful and enjoyed Professor Sotomayer's enthusiasm. Class notes posted also helped, and as always the p–sets helped but to a lesser extent than 141.
Assignments were helpful
Lecture notes on Canvas were really comprehensive
I didn’t really go because it’s hard to learn and takes notes in a CS lecture but was a helpful tool on occasion
Lectures were extremely helpful
lectures were very engaging and insightful. Borja is an amazing lecturer and you can sense his enthusiasm for every class on the topic being covered that day. He has a way of explaining complex material in a very simplistic way that is easy to understand. never had to read any sort of textbook except maybe review some lecture notes. discussions were not very helpful
Lectures mostly covered big picture, ways of thinking, and pseudocode. Assignments helped with getting implementation experience with particular concepts. Discussion sessions covered skills / knowledge necessary to projects and overall software development.
The notes were the most helpful. They were comprehensive and made it very easy to learn the material.
I think the office hours was generally very helpful.
The lectures and lecture notes helped me with the midterm. The final project helped me gain "real–life" experience with a project.
Lectures provided a great way to go through the specific algorithms and data structures we were learning at the time. Most times the instructor provided a visual representation which helped greatly, alongside python code which helped us know exactly what was going on.
Lectures were awesome
n/a
Lecture notes were most helpful

Please respond to the following:

	Mean	Median	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
This course challenged me intellectually.	3.77	4.00	6.45%	6.45%	12.90%	51.61%	22.58%
I understood the purpose of this course and what I was expected to gain from it.	4.32	4.00	3.23%	0.00%	3.23%	48.39%	45.16%
I understood the standards for success on assignments.	4.19	4.00	0.00%	3.23%	9.68%	51.61%	35.48%
Class time enhanced my ability to succeed in graded assignments.	3.68	4.00	3.23%	6.45%	35.48%	29.03%	25.81%
I received feedback on my performance that helped me improve my subsequent work.	4.35	4.00	0.00%	0.00%	6.45%	51.61%	41.94%
My work was evaluated fairly.	4.45	5.00	0.00%	3.23%	3.23%	38.71%	54.84%
I felt respected in this class.	4.35	5.00	3.23%	0.00%	6.45%	38.71%	51.61%
Overall, this was an excellent course.	4.13	4.00	3.33%	3.33%	6.67%	50.00%	36.67%

Additional comments about the course:

Comments
If you're not a CS major, don't waste your time with this class. I took the course wanting to develop my skills in programming so I could apply it to other fields I'm interested in. Which seems like a reasonable goal. However, after the midterm, we did a group project that was supposed to teach us skills in software development. One — I'm still not quite sure what I'm meant to have learned. That communication is important when working in a team? That 4 people shouldn't be working on one computer doing the same thing at a time? Two — if the CS department wants to teach these skills (which they stressed were important for those who wanted to become software engineers), they should do it in a class meant for students *majoring in computer science*. Not in an intro course that is taken by students of all majors. My biggest regret in course selection at UChicago. I also went to about 20% of the lectures before the midterm, read the course notes the day before the midterm, and got a 98% (which gives you an A in the class, since your whole grade is determined by the midterm score due to the final project just being graded for being "satisfactory"). Shows you how much content they DIDN'T teach in this course. I was actually interested in computer science before this course (thank you to the 141 instructors for giving me a solid foundation). But now, I'm just confused as to what the field as a whole is, and my programming skills are the same as when I ended 142.
A big step up from 141 in the first half, project helps the class become a lot easier.
I would recommend this CS sequence to actually anyone. I came in with no CS background, I've taken CS 141 and 142 now (along with Discrete and Theory of Algorithms, which also helped ngl) and I've landed a SWE internship at a FAANG company this summer. I went into my technical interviews with no further knowledge than what was taught to me in class and I just tried my best and got the offer! This sequence taught me everything I know and I'm so grateful for it. This obviously won't be the case for everyone loll I'm definitely inclined to this sort of thing, but if you try your best and really learn the material, you might love it as much as me.
Borja is perfect.
felt like lectures on second half on software development could've been condensed. feel like we could've spent more time on the data structures part at the start of the quarter
After midterms, it seemed like students no longer has an incentive to attend class nor discussions as this no longer directly translated to a grade (midterms). Because of this, many people didn't care to learn about libraries and other post–midterm material cause they could simply pick it up for a project task without attending lecture.
Borja is pretty cool, nice t–shirts
HWs took a long time.
Too much focus (thematically or even explicitly) on professional preparedness. Too little focus on the actual computer science. If the class weren't required to take any other CS course, it would be fine. Assignments are never conceptually challenging, but annoying. There's an enormous emphasis on the style of code, as far as grading goes. That's not a horrible thing—in fact, it's pretty great. However, sometimes it goes too far. For instance—getting deducted 8% on a written exam for code written…on paper. (Ranting begins). Yes, I am bitter about it, but it is so absurd. The answer itself was correct—had it been typed and ran on a computer, it would have performed the task. Apparently, the TA found it was too complicated for the task (I have looked over it, and it seems I interpreted the given class differently. I wrote my code to do something I thought the given code wasn't already doing). The exam was on paper—mind you. Not even real code. And they took off half the points on a question on a correct answer. Because it was too "complicated." I read the instructions on the exam, and no where was it emphasized that style would factor into code written on paper. This one question ticked me down an entire letter grade level because of how the exam factored into the grade. If I had done something wrong, I would have understood. That would be fair game, obviously (that's how exams work). But taking so many points off for imaginary code, technically correct code, written by pencil, is so tragically absurd, it's sad. I'm sad for myself, honestly. Also, the TAs seem overrun with work. Grading often seems rushed (like getting something marked wrong or missing on a project because the grader didn't actually read how the code was structured to understand what it was actually doing…oddly specific, I know). There's an enormous amount of students taking this class. You might need to hire more TAs.
The latter half of the quarter had to do with the final project and didn't really contribute to my learning.

I would recommend this course to:

	No	Yes
Highly-motivated and well-prepared students	9.68%	90.32%
Anyone interested in the topic	16.13%	83.87%

Thinking about your time in the class, what aspect of the instructor's teaching contributed most to your learning?

Comments
his lectures were good but kind of simple
The examples he brought up in class, along with his enthusiasm to teach, which made me more willing to learn
Notes by far.
Professor was really nice, always asked for questions and engaged well with students.
Nothing
Professor Sotomayor was extremely engaging and made the course content very accessible. The way he explained the content helped in my understanding of it tenfold and I attended almost every single lecture up until the very end, even after the midterm! You can tell he is very passionate and knowledgeable about the topic, and that made me and my peers happy to learn from him.
He explained things very well and motivated why we were learning what we were learning
Prof Borja's lectures were really engaging. His explanations, especially when explaining algorithms like Dijkstra, BFS, DFS were very clear. In the second half of the course it was interesting to hear his experiences in software development.
Lectures + answering questions was really good
Professor is a good lecturer.
The examples in class where he would show what is going on underneath the hood for a specific function call or algorithm.
really good at explaining concepts
N/A
he was very clear and usually concise
lectures were amazing and very engaging
Blackboard diagrams, live–coding.
I think the visual examples that he made on the board.
The lectures on the graphs and trees, the more technically complicated material that needed explaining.
The lectures were incredibly helpful and broke down the larger ideas in the course into manageable pieces.
Lectures
Lecture notes
Borja's clear explanations during class time was the best aspect

What could the instructor modify to help you learn more?

Comments
maybe go over more complex stuff in class instead of taking a long time explaining simple concepts
Went more in depth into specific topics instead of spending a lot of time on not as relevant stuff
–
Completely rehaul the course again, and consult the 141 instructors on how to design it.
Prof Sotomayor is amazing!!!
Nothing really.
Some of the instructions on the homeworks could be clearer and better organized. Often it was a wall of text.
Sometimes the examples in class felt a little too easy relative to the homework, so going over more difficult examples would be nice.
N/A
Nothing.
Nothing
Dedicate more time to topics like graphs / software development.
Possibly adjusting the way the post–midterm is structured. After midterms people had low incentives to attend class much less discussion sessions because their project had little interaction with the content in class (people could use things like click library without attending class). I think this structure made the end of the quarter more relaxing, but also lost a lot of learning opportunities because of how loosely it was structured. However, the project was a good experience, and I think a group project such as Go was indeed helpful and instructive.
HWs were very rough.
Don’t know

The Instructor . . .

	Mean	Median	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree	N/A
Organized the course clearly.	4.43	5.00	0.00%	0.00%	10.34%	34.48%	51.72%	3.45%
Presented lectures that enhanced your understanding.	4.14	4.00	3.45%	0.00%	20.69%	27.59%	44.83%	3.45%
Facilitated discussions that were engaging and useful.	3.85	4.00	3.45%	3.45%	24.14%	31.03%	27.59%	10.34%
Stimulated your interest in the core ideas of the course.	4.21	4.00	3.45%	0.00%	6.90%	48.28%	37.93%	3.45%
Challenged you to learn.	4.14	4.00	3.45%	3.45%	10.34%	37.93%	41.38%	3.45%
Helped you gain significant learning from the course content.	4.36	5.00	3.45%	0.00%	6.90%	34.48%	51.72%	3.45%
Was available and helpful outside of class.	4.35	4.00	0.00%	0.00%	10.34%	37.93%	41.38%	10.34%
Motivated you to think independently.	4.33	5.00	3.45%	0.00%	10.34%	27.59%	51.72%	6.90%
Worked to create an inclusive and welcoming learning environment.	4.41	5.00	3.45%	0.00%	6.90%	27.59%	55.17%	6.90%
Overall, this instructor made a significant contribution to your learning.	4.39	5.00	3.45%	0.00%	6.90%	31.03%	55.17%	3.45%

Please include the name of the TA/CA/Intern you are evaluating. What aspects of the TA's teaching contributed most to your learning? What could the TA modify to help you learn more? Please include any additional feedback for the TA/CA/Intern.

Comments
Alex Since the topics we discussed weren't brought up during the lecture, I learned quite a lot too, but since the lecture slides can't be taught as much without someone demonstrating the content, I thought how my TA taught was pretty cool. The discussions were optional, so I know a lot of people decided not to go, but I'd recommend going to discussion because the concepts (in my opinion) are cooler than what is taught in lecture
TA didn't do anything wrong but I don't like the way the discussion sessions were run
I don't know—I went to one discussion section then stopped going since they weren't required. My friends who went to a few more said the content covered was mininmal and they generally did not learn much during them. This isn't the fault of theTAs, but the curriculum they're given to teach.
Yiming Su was really helpful in debugging during office hours. It was also fun talking to him about his experience in CS.
Jack. Helpful during discussions and discussion lectures.
Did not attend TA sessions ngl
Alex

The TA/CA or Intern. . .

	Mean	Median	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree	N/A
Facilitated discussions that supported your learning.	4.25	4.00	0.00%	0.00%	0.00%	37.50%	12.50%	50.00%
Gave you useful feedback on your work.	4.67	5.00	0.00%	0.00%	0.00%	12.50%	25.00%	62.50%
Stimulated your interest in the core ideas of the class.	4.33	4.00	0.00%	0.00%	0.00%	25.00%	12.50%	62.50%
Challenged you to learn.	4.33	4.00	0.00%	0.00%	0.00%	25.00%	12.50%	62.50%
Helped you succeed in the class.	4.33	4.00	0.00%	0.00%	0.00%	25.00%	12.50%	62.50%
Was available and helpful outside of class.	4.00	4.00	0.00%	0.00%	0.00%	12.50%	0.00%	87.50%
Overall, this individual made a significant contribution to your learning.	4.25	4.00	0.00%	0.00%	0.00%	37.50%	12.50%	50.00%

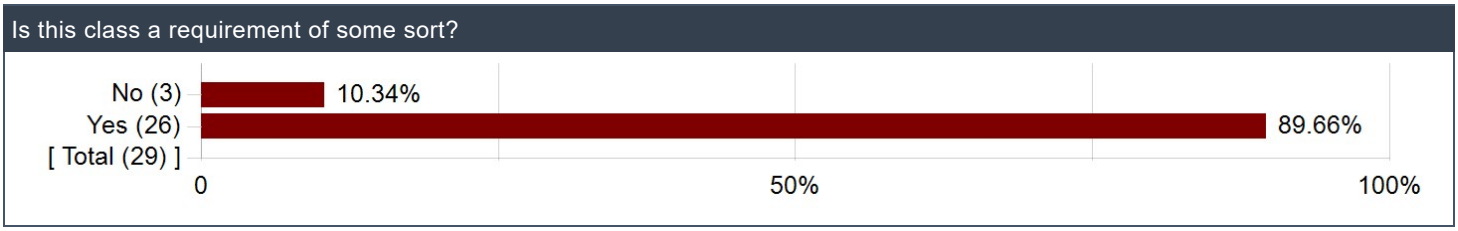
How much did the following elements of the course contribute to your learning gains?

	Mean	Median	No Gain	A Little Gain	Moderate Gain	Good Gain	Great Gain	N/A
Laboratory Experience	N/A	N/A	0.00%	0.00%	0.00%	0.00%	0.00%	100.00%
Field Trips	N/A	N/A	0.00%	0.00%	0.00%	0.00%	0.00%	100.00%
Library Sessions	N/A	N/A	0.00%	0.00%	0.00%	0.00%	0.00%	100.00%
Review Sessions	1.00	1.00	25.00%	0.00%	0.00%	0.00%	0.00%	75.00%
Writing Seminars	N/A	N/A	0.00%	0.00%	0.00%	0.00%	0.00%	100.00%

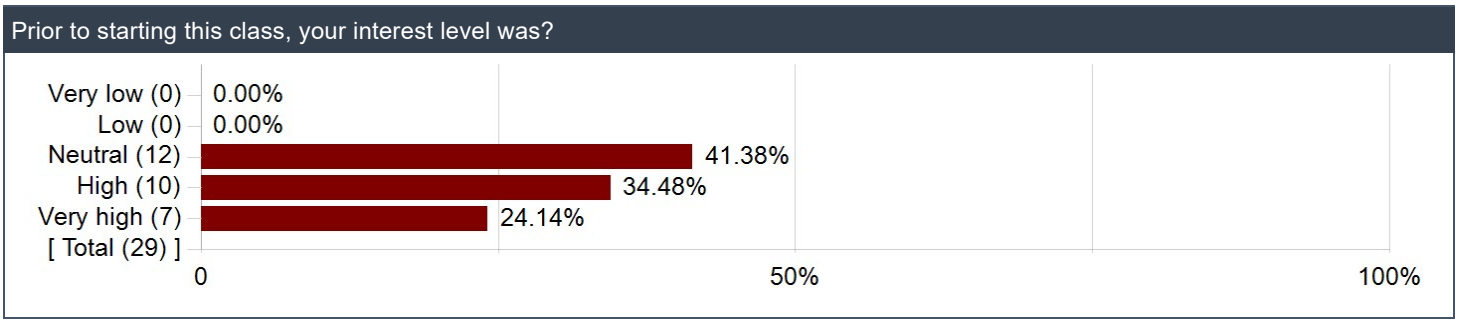
Other course elements not mentioned above:

Comments
We had discussion sessions but due to the timing of them I did not attend often.

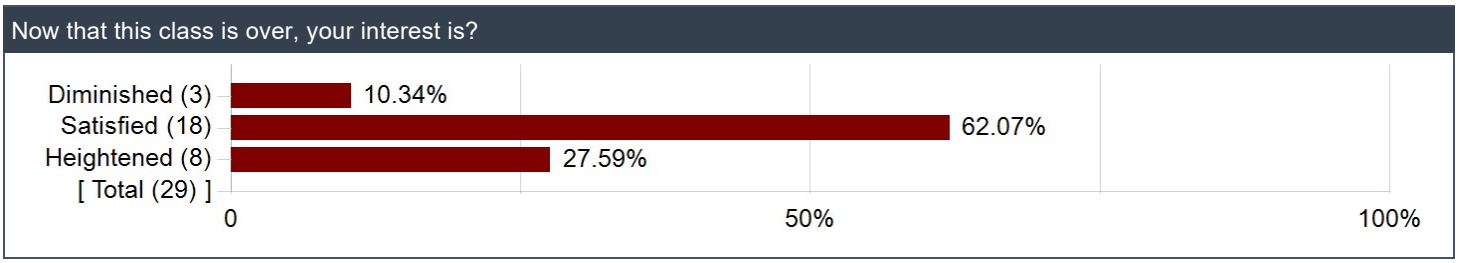
Is this class a requirement of some sort?



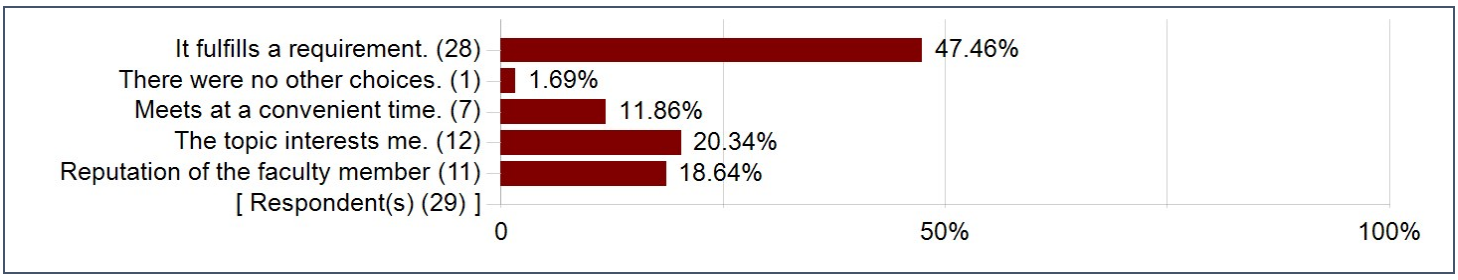
Prior to starting this class, your interest level was?



Now that this class is over, your interest is?

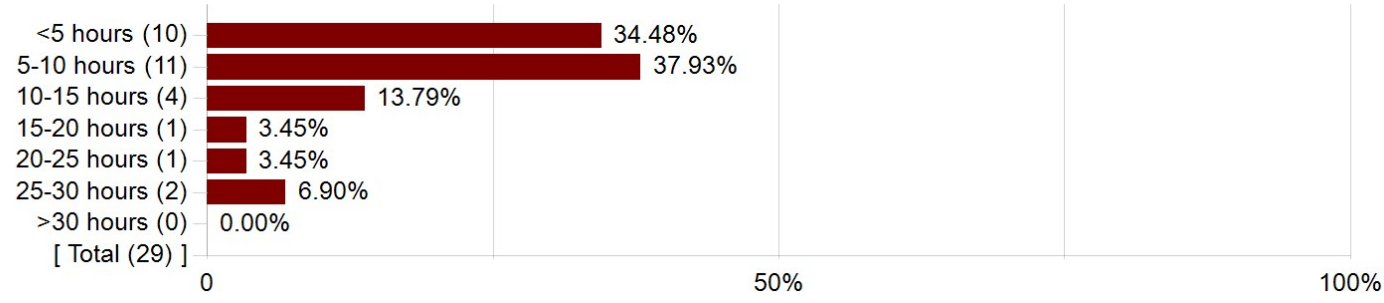


Why did you choose to take this course? (Select all that apply)



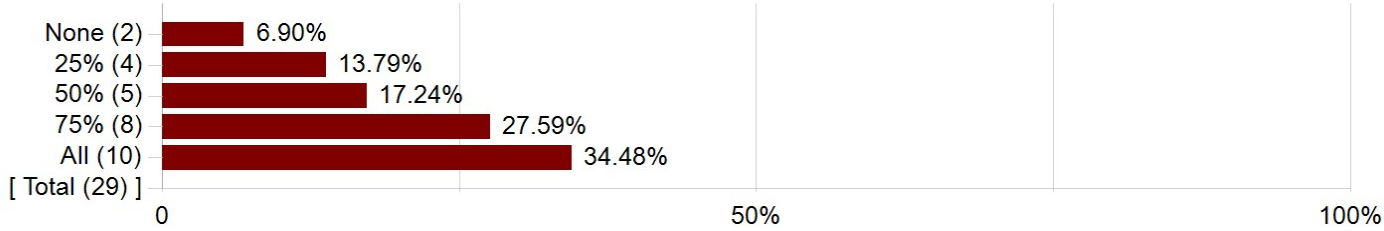
How many hours per week outside of attending required sessions did you spend on this course?

How many hours per week outside of attending required sessions did you spend on this course?



What proportion of classes did you attend?

What proportion of classes did you attend?



Please comment on the level of difficulty of the course relative to your background and experience.

Comments
very easy with coding experience from high school and not taking 141
If you have no prior background in CS, struggled in CMSC 14100, or aren't willing to dedicate a lot of time to learning many of the concepts on your own, then I wouldn't recommend this class.
It was pretty easy for me but I have a lot of python and CS experience so that makes sense. I know it was more challenging for a lot of my less experienced classmates
Defintely challenging.
The course is easier than cs141
Not a cs major, took 1 year in high school then 141. Very easy because they didn't actually teach any content. We weren't even required to know how to code the traversal algorithms and methods we "learned" — you could memorize two bullet points' worth of information then get all the multiple choice questions about these on the exam correct. I think I could've solved the midterm coding problems directly after taking 141.
If recursion was your weakest point in 141 (like it was for me), this class will definitely be tough when it comes to the midterm. Set reasonable expectations for yourself. Overall, it was a well–structured extension of 141 that flowed well, and I think that if you took 141 and found it reasonable, 142 should also be no problem as long as you work hard to understand the content.
I took 141 and learned some algorithms in high school (insertion sort, binary search) and this class was doable. I still learned a lot from the project component!
easy after placing out of 141
Compared to CS 141 I felt the content from lectures and tests was substantially easier because you have the intuition, but the p–sets were substantially harder and often ventured into topics that weren't really covered in class, and was a very big expansion of what we did in class. It felt really hard during the first half, but the project during the second half was definitely much more relaxing, especially being the QA
Have prior experience which helped. The software development part was new.
Skipped out of CS 141, knew Python beforehand, and did some game development in Java. I didn't think it was too hard but the assignments were long.
starts from where CS141 leaves off
was a good step up in level of difficulty from cmsc 141
I have prior programming experience (none with python), and I had skipped 141 via testing out of it. This was not too hard to pick up, but could still be a challenge with harder to understand topics like graph traversals.
I have fair computer science experience, but many class topics were new to me. This class was not horribly difficult but did provide a good challenge at times.
Make sure you study for midterm. Whether you get 90+% determined whether you get an A or not.
Didn’t learn anything new, just reviewed things that I was a bit rusty on. Became more organized at writing code, however.
It could be frustrating at times but manageable if you put in the work.