CSS 343 A
Data Structures, Algorithms, And Discrete Mathematics II
Course type: Face-to-Face

Taught by: Yusuf Pisan
Instructor Evaluated: Yusuf Pisan-Lecturer

Evaluation Delivery: Online
Evaluation Form: D
Responses: 21/37 (57% high)

Overall Summative Rating represents the combined responses of students to the four global summative items and is presented to provide an overall index of the class's quality:

Median	College Decile
4.6	**7**
(0=lowest; 5=highest)	(0=lowest; 9=highest)

Challenge and Engagement Index (CEI) combines student responses to several *IASystem* items relating to how academically challenging students found the course to be and how engaged they were:

CEI: 5.6
(1=lowest; 7=highest)

SUMMATIVE ITEMS

	N	Excellent (5)	Very Good (4)	Good (3)	Fair (2)	Poor (1)	Very Poor (0)	Median	DECILE RANK Inst	College
The course as a whole was:	21	48%	29%	14%	10%			4.4	6	6
The course content was:	21	48%	29%	19%	5%			4.4	6	6
The instructor's contribution to the course was:	21	57%	24%	19%				4.6	5	6
The instructor's effectiveness in teaching the subject matter was:	21	62%	19%	10%	10%			4.7	7	7

STUDENT ENGAGEMENT

Relative to other college courses you have taken:	N	Much Higher (7)	(6)	(5)	Average (4)	(3)	(2)	Much Lower (1)	Median	DECILE RANK Inst	College
Do you expect your grade in this course to be:	20	10%	20%	25%	20%	5%	10%	10%	4.7	2	3
The intellectual challenge presented was:	20	35%	40%	15%	10%				6.1	8	7
The amount of effort you put into this course was:	20	45%	25%	20%	5%		5%		6.3	8	8
The amount of effort to succeed in this course was:	20	25%	45%	15%	15%				5.9	6	5
Your involvement in course (doing assignments, attending classes, etc.) was:	20	50%	30%	5%	10%		5%		6.5	8	8

On average, how many hours per week have you spent on this course, including attending classes, doing readings, reviewing notes, writing papers and any other course related work?

Class median: 12.9 Hours per credit: 2.6 (N=20)

Under 2	2-3	4-5	6-7	8-9	10-11	12-13	14-15	16-17	18-19	20-21	22 or more
				5%	20%	35%	20%	10%	5%	5%	

From the total average hours above, how many do you consider were valuable in advancing your education?

Class median: 10.5 Hours per credit: 2.1 (N=20)

Under 2	2-3	4-5	6-7	8-9	10-11	12-13	14-15	16-17	18-19	20-21	22 or more
			10%	20%	40%	20%	10%				

What grade do you expect in this course?

Class median: 3.5 (N=20)

A (3.9-4.0)	A- (3.5-3.8)	B+ (3.2-3.4)	B (2.9-3.1)	B- (2.5-2.8)	C+ (2.2-2.4)	C (1.9-2.1)	C- (1.5-1.8)	D+ (1.2-1.4)	D (0.9-1.1)	D- (0.7-0.8)	E (0.0)	Pass	Credit	No Credit
15%	45%	10%	25%			5%								

In regard to your academic program, is this course best described as:

(N=20)

In your major	A core/distribution requirement	An elective	In your minor	A program requirement	Other
65%	35%				

STANDARD FORMATIVE ITEMS

	N	Excellent (5)	Very Good (4)	Good (3)	Fair (2)	Poor (1)	Very Poor (0)	Median	DECILE RANK Inst	DECILE RANK College
Course organization was:	20	45%	30%	25%				4.3	5	6
Sequential presentation of concepts was:	20	50%	30%	15%	5%			4.5	6	7
Explanations by instructor were:	20	60%	10%	20%	10%			4.7	7	7
Instructor's ability to present alternative explanations when needed was:	20	65%	5%	10%	20%			4.7	7	7
Instructor's use of examples and illustrations was:	20	50%	25%	20%		5%		4.5	5	5
Quality of questions or problems raised by the instructor was:	20	45%	25%	20%	5%	5%		4.3	4	5
Contribution of assignments to understanding course content was:	20	50%	35%	15%				4.5	6	6
Instructor's enthusiasm was:	20	60%	20%	15%	5%			4.7	4	5
Instructor's ability to deal with student difficulties was:	20	50%	20%	20%	5%	5%		4.5	6	6
Answers to student questions were:	20	50%	30%	15%			5%	4.5	5	6
Availability of extra help when needed was:	20	35%	45%	15%	5%			4.2	3	3
Use of class time was:	20	65%	10%	25%				4.7	8	8
Instructor's interest in whether students learned was:	20	65%	10%	15%		10%		4.7	6	7
Amount you learned in the course was:	20	55%	25%	15%	5%			4.6	6	7
Relevance and usefulness of course content were:	20	70%	15%	10%	5%			4.8	8	8
Evaluative and grading techniques (tests, papers, projects, etc.) were:	20	55%	15%	10%	15%		5%	4.6	6	7
Reasonableness of assigned work was:	20	45%	35%	10%	5%		5%	4.4	5	5
Clarity of student responsibilities and requirements was:	20	60%	25%	5%	5%		5%	4.7	7	7

CSS 343 A
Data Structures, Algorithms, And Discrete Mathematics II
Course type: Face-to-Face

Evaluation Delivery: Online
Evaluation Form: D
Responses: 21/37 (57% high)

Taught by: Yusuf Pisan
Instructor Evaluated: Yusuf Pisan-Lecturer

STANDARD OPEN-ENDED QUESTIONS

Was this class intellectually stimulating? Did it stretch your thinking? Why or why not?

1. This class was intellectually challenging but not interesting to me. The problems presented in class were not stimulating, and it was very difficult for me to apply what I had just listened to to the exercises in class. I feel that we did not have enough examples or problems worked through before attempting it ourselves, which made me feel low and not want to ask for help.

2. I learned so much in this class. It felt relevant to what I want to do with my career and was a great use of my time. The concepts were challenging at first, but I grasped them with working with them.

3. Simply due to the subject matter the class is intellectually stimulating. However, Pisan also encourages his students to engage with the material by using in class activities.

4. Yes, assignments required you to think a lot about program design and efficient ways of implementation. In class exercises encouraged practical application of knowledge learned during lectures and readings.

5. Yes the concepts in this class are difficult to understand and requirement you to pay attention during the entire lecture because if you missed one part then you were out of luck.

6. I found all the homework assignments helpful for understanding in the many topics covered.

7. Yes

8. Yes, it introduced quite a few unfamiliar subjects.

9. Yes, there were new problems I had not faced in my time as a developer and subjects I had never discussed before as a self-taught developer

10. Yes, especially the last assignment because it wasn't very constricting in terms of how to design the program. I was able to apply all the knowledge I learned in class in a creative way that made my program worked as well as exploring other possibilities that could have work/improved my programs

12. Yes, most of the information covered in this course was new to me and it expanded my knowledge what I already knew.

13. Was pretty interesting, especially the architecture parts.

14. Yes this class was intellectually stimulating. I enjoyed learning about new data structures, as well as discussing about Finite State Machines, Turing Machines, and other CS related concepts.

16. Yes, it was very interesting. It helped sharpen my coding and problem solving skills and taught me more software techniques.

17. It was stimulating.

18. Yes, the assignments were challenging

19. CSS 343 is intellectually stimulating because we are learning topics related to computer science that are new and at times difficult to grasp. I believe that the Data Structures two part sequence forces us to think about what we're coding rather than simply coding. There is a lot of material in this course related to logic and theory that I believe self-taught programmers do not learn.

What aspects of this class contributed most to your learning?

1. The assignments and textbook were the only things that helped me learn. I much prefer to work through things on my own. I never felt like I had been given enough practical information to work through an exercise in class, and it was too fast paced for me to get a grasp on any concept.

2. I loved having examples to work on in class with my tablemates. This helped prepare me for the midterm. It helped me confirm that I actually knew how to do the problems. The programs were difficult but adequately stretched my abilities. Slack was a great resource for asking and answering questions and made it so the instructor was accessible.

3. The lectures and in class activities contributed most to my overall understanding of the concepts introduced. However, the way I best learned how to actually apply these concepts to coding the assignments was by google because, we almost never talked about coding or saw examples during class.

4. Class assignments which were larger coding projects that were implemented separately or in groups, and in class exercises which encourage application of knowledge learned in class.

5. Lots of examples of what we were taught really helped. The break in between the 2 hour lecture also helped to get your mind back in learning mode.

6. Assignments upon lecture topics and office hour / Slack explanations.

7. Lecture

8. The lectures contributed the most to my learning of concepts, while the assignments gave me more technical experience.

9. Team/group work. I had to explain things, thus reinforcing my own knowledge

10. Going over examples in class and having questions answered by the professor either in class or in office meetings.

11. The in class exercises were amazing; perfect balance of forcing you to think about what you've learned while still being within your abilities to solve having just learned the material.

12. The coding assignments contributed most to my learning. It forces you to use what you have learned in class and apply it.

13. The in-class activities helped.

14. I love how Professor Pisan explains things in the lectures. It always makes it worth it for me to come to class.

15. Going through examples on the board with input and questions from students, practice questions before exams, group design sessions

16. The varied content and assignments

17. The assignments and book content.

18. The bi-weekly assignments

19. I think this question needs to be answered for two things. The logic and theory of programming concepts was primarily taught in the form of lecture and practice problems in-class. In regard to actual programming, what contributed the most was probably google. During and after a programming assignment, Professor Pisan was great about giving us guidance on how to approach certain problems and in what way we could optimize our code. However, we ultimately are given an assignment that we have to complete with the tools we have available, which I am under the impression is accurate to what being an actual programmer is going to be like. Obviously as we gain more experience, I assume we will google less, but we are learning how to learn as a programmer in this course.

What aspects of this class detracted from your learning?

1. The lectures, as described in the other questions. Sometimes the assignments were vague and confusing. The midterm study guide was useful for the midterm, but many of the problems were even incomplete, which was very confusing.

2. Chatty students sitting near me who would talk during lecture.

3. For the final assignment we were allowed to first work in groups of 4 to design the assignment and then we could choose to pair up and code as team of two. This by far was my best learning experience in the class and I wish we had similar opportunities for the other assignments. I felt like I learned a lot less when I tried to program an assignment by myself. simply having someone to bounce ideas off of while looking at the same code was incredibly helpful.

4. Assignments specifications were sometimes unclear about what exactly was expected from the finished program. More time clarifying these specifications was spent in class.

5. I wish we had a way of asking questions without having to raise our hands in class. People who don't like to talk in front of the classroom suffer because of this.

6. Somewhat abrupt switch in topics from inheritance to FSM/Turing Machines.

7. Assignments

8. Assignments were fairly strictly directed. Makes sense for grading but doesn't allow for very much personal design.

9. Only the people sitting behind me. They constantly were talking

10. I guess the length of the class because my mind can only intake a certain amount of information, but there's nothing you can do with uwb classes having a minimum of 2 hour classes that meet 2 times a week and no one hour daily classes.

12. Nothing comes to mind.

13. Last assignment was a big jump in work compared to precedent.

14. There was not anything specific that detracted from my learning, however I was going through some problems this quarter and that is why I was regrettably unable to give my best effort.

17. Not having ponopto.

18. Nothing

19. Nothing in particular detracted from this course. I actually feel that I learned the most in this course versus the first part of the data structures sequence. Although I struggled with the structure of this data structures course compared to the (what I realize now) lack of structure in my first data structures class. The professor I had in my first class had inconvenient office hours, rarely communicated with classmates questions over email, did not discuss the assignments further than explaining what the assignments was, and did not require that we compile our program on the Linux computers to test for errors and memory leaks. Being in Professors Pisan class helped me learn a lot of skills that I believe are going to be important for me as a software engineer, but having to learn a new class structure made it difficult for me to do as well as I would have wanted. My biggest complaint isn't about this course, but rather that all data structures courses should be structured in the same way. From what I hear, most of them are and my class may have been an exception, but the fact that there are exceptions in a school where we pay a large amount for each course (at least large regarding my financial status), I think there should be consistency.

What suggestions do you have for improving the class?

1. Give more students to work through exercises on their own in-class. Do not walk around the room while they are working; that gave me anxiety as a student when I did not understand something.

2. More practice coding out algorithms, this was especially apparent with the Tree Quiz we took in class.

3. I would love practice sets of code that we could do in class so that we have a chance to interact with some code while the instructor is present. This was how my other classes before transferring worked and I can say that, by far, I learned more and I learned faster than with just lectures and pen and paper practices. Not that the pen and paper doesn't do a good job, but I never learned any actual coding from them so I would like to have both. At the very least it would be nice to have practice sets that we could complete on our own. An example of a practice set that would have been great for the final assignment is a simple class that is polymorphic but then we need to create the subclass that accesses and overrides one or two of it's functions. Overall I feel that I would have learned more if we had interacted with more code, I feel like we very rarely even talked about coding, mostly it was just general concepts such as polymorphic classes or AVL trees.

4. Provide clearer assignment specifications.

5. The one exam and one midterm we had were worth 55% of our entire grade! I don't know how this is really fair if you mess up on one of them (getting below 85) you really have to push in order to get above 90%. The stress of having to receive perfect scores on everything after getting an OK grade on the midterm really distracts from the learning aspect. You cannot learn information/concepts while under stress.

6. Implement Jolly Feedback features earlier.

7. easier assignments

8. I think having open-ended assignments that were smaller in scope might encourage deeper problem solving rather than the ability to follow documentation/provided files. Both are necessary skills, but a balance would be nice.

9. Talked with Professor Pisan already. They are adding source control/git to the class.

10. N/A I like the class just the way it is.

11. More extra credit opportunities.

12. Maybe include another assignment where you can work with a partner. Also some small additional programming practice in class would help with remembering information.

13. n/a

14. Hmm, I would have liked maybe a little more time for Assignment 4 since we ended up losing half a week because of Thanksgiving. I definitely know that if I had a little bit more time, then my partner and I would have been able to do a better job on our assignment.

16. Add a more colorful ass4, maybe GUI because we mainly learned console applications.

17. Please do ponopto for future students. Slightly more easier grading on tests would help as well. Sometimes students know the content but display only partial knowledge.

18. N/A

19. To improve this class, I believe that we could benefit from having in-class programming exercises and if we practiced hand writing code more often. Having a quiz near the end of the quarter where we had to implement code on paper showed me that we are really lacking in that practice. Having the skill to write down our code is helpful for planning the design of a system and how we will approach programming problems. Having in-class exercises could also help with this as I think collaborative problem-solving can be extremely helpful at learning what the best ways to approach a problem are.

IASystem Course Summary Reports summarize student ratings of a particular course or combination of courses. They provide a rich perspective on student views by reporting responses in three ways: as frequency distributions, average ratings, and either comparative or adjusted ratings. Remember in interpreting results that it is important to keep in mind the number of students who evaluated the course relative to the total course enrollment as shown on the upper right-hand corner of the report.

Frequency distributions. The percentage of students who selected each response choice is displayed for each item. Percentages are based on the number of students who answered the respective item rather than the number of students who evaluated the course because individual item response is optional.

Median ratings. *IASystem* reports average ratings in the form of item medians. Although means are a more familiar type of average than medians, they are less accurate in summarizing student ratings. This is because ratings distributions tend to be strongly skewed. That is, most of the ratings are at the high end of the scale and trail off to the low end.

The median indicates the point on the rating scale at which half of the students selected higher ratings, and half selected lower. Medians are computed to one decimal place by interpolation.[1] In general, higher medians reflect more favorable ratings. To interpret median ratings, compare the value of each median to the respective response scale: *Very Poor, Poor, Fair, Good, Very Good, Excellent (0-5); Never/None/Much Lower, About Half/Average, Always/Great/Much Higher (1-7); Slight, Moderate, Considerable, Extensive (1-4).*

Comparative ratings. *IASystem* provides a normative comparison for each item by reporting the decile rank of the item median. Decile ranks compare the median rating of a particular item to ratings of the same item over the previous two academic years in all classes at the institution and within the college, school, or division. Decile ranks are shown only for items with sufficient normative data.

Decile ranks range from 0 (lowest) to 9 (highest). For all items, higher medians yield higher decile ranks. The 0 decile rank indicates an item median in the lowest 10% of all scores. A decile rank of 1 indicates a median above the bottom 10% and below the top 80%. A decile rank of 9 indicates a median in the top 10% of all scores. Because average ratings tend to be high, a rating of "good" or "average" may have a low decile rank.

Adjusted ratings. Research has shown that student ratings may be somewhat influenced by factors such as class size, expected grade, and reason for enrollment. To correct for this, *IASystem* reports **adjusted medians** for summative items (items #1-4 and their combined global rating) based on regression analyses of ratings over the previous two academic years in all classes at the respective institution. If large classes at the institution tend to be rated lower than small classes, for example, the adjusted medians for large classes will be slightly higher than their unadjusted medians.

When adjusted ratings are displayed for summative items, **relative rank** is displayed for the more specific (formative) items. Rankings serve as a guide in directing instructional improvement efforts. The top ranked items (1, 2, 3, etc.) represent areas that are going well from a student perspective; whereas the bottom ranked items (18, 17, 16, etc.) represent areas in which the instructor may want to make changes. Relative ranks are computed by first standardizing each item (subtracting the overall institutional average from the item rating for the particular course, then dividing by the standard deviation of the ratings across all courses) and then ranking those standardized scores.

Challenge and Engagement Index (CEI). Several *IASystem* items ask students how academically challenging they found the course to be. *IASystem* calculates the average of these items and reports them as a single index. *The Challenge and Engagement Index (CEI)* correlates only modestly with the global rating (median of items 1-4).

Optional Items. Student responses to instructor-supplied items are summarized at the end of the evaluation report. Median responses should be interpreted in light of the specific item text and response scale used (response values 1-6 on paper evaluation forms).

[1] For the specific method, see, for example, Guilford, J.P. (1965). Fundamental statistics in psychology and education. New York: McGraw-Hill Book Company, pp. 49-53.