CSS 342 D
Data Structures, Algorithms, And Discrete Mathematics I
Course type: Face-to-Face

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

Evaluation Delivery: Online
Evaluation Form: D
Responses: 34/38 (89% very 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
4.1	**4**
(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	DECILE RANK College
The course as a whole was:	34	35%	35%	21%	9%			4.1	4	5
The course content was:	34	38%	38%	12%	12%			4.2	4	5
The instructor's contribution to the course was:	34	41%	35%	21%		3%		4.2	3	4
The instructor's effectiveness in teaching the subject matter was:	34	29%	38%	26%	3%	3%		4.0	2	3

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	DECILE RANK College
Do you expect your grade in this course to be:	33	12%	6%	21%	27%	18%	12%	3%	4.1	0	1
The intellectual challenge presented was:	33	36%	24%	12%	18%	6%	3%		5.9	7	6
The amount of effort you put into this course was:	33	30%	30%	12%	27%				5.8	5	5
The amount of effort to succeed in this course was:	33	39%	36%	6%	15%	3%			6.2	8	7
Your involvement in course (doing assignments, attending classes, etc.) was:	33	27%	36%	18%	15%	3%			5.9	4	4

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: 11.7 Hours per credit: 2.3 (N=33)

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
		9%	12%	12%	15%	15%	6%	6%	15%	9%	

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

Class median: 9.7 Hours per credit: 1.9 (N=33)

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
	6%	12%	21%	9%	15%	15%	9%	3%	3%	3%	3%

What grade do you expect in this course?

Class median: 3.3 (N=33)

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
18%	24%	12%	15%	15%	12%		3%							

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

(N=33)

In your major	A core/distribution requirement	An elective	In your minor	A program requirement	Other
61%	27%	3%		9%	

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:	33	33%	36%	27%		3%		4.0	4	4
Sequential presentation of concepts was:	33	39%	39%	21%				4.2	5	6
Explanations by instructor were:	33	30%	30%	36%	3%			3.9	2	3
Instructor's ability to present alternative explanations when needed was:	33	33%	30%	30%	6%			4.0	2	3
Instructor's use of examples and illustrations was:	33	36%	30%	21%	9%	3%		4.0	3	3
Quality of questions or problems raised by the instructor was:	33	42%	45%	9%		3%		4.3	5	5
Contribution of assignments to understanding course content was:	33	42%	42%	9%	6%			4.3	5	5
Instructor's enthusiasm was:	33	55%	36%	9%				4.6	3	4
Instructor's ability to deal with student difficulties was:	32	31%	47%	12%	6%	3%		4.1	4	4
Answers to student questions were:	33	39%	42%	12%	6%			4.2	4	5
Availability of extra help when needed was:	33	42%	33%	21%	3%			4.3	3	4
Use of class time was:	33	42%	27%	24%	3%	3%		4.2	4	5
Instructor's interest in whether students learned was:	33	39%	36%	12%	9%	3%		4.2	3	3
Amount you learned in the course was:	32	31%	34%	28%	3%	3%		4.0	3	3
Relevance and usefulness of course content were:	32	47%	41%	12%				4.4	5	5
Evaluative and grading techniques (tests, papers, projects, etc.) were:	33	33%	30%	30%		6%		4.0	3	3
Reasonableness of assigned work was:	33	48%	36%	9%	6%			4.5	5	6
Clarity of student responsibilities and requirements was:	33	42%	30%	21%	6%			4.2	4	5

CSS 342 D
Data Structures, Algorithms, And Discrete Mathematics I
Course type: Face-to-Face

Evaluation Delivery: Online
Evaluation Form: D
Responses: 34/38 (89% very 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. Yes. The assignments were challenging.

2. I am not sure what other students' backgrounds are; however, I felt that the first part of the quarter was focused heavily on things that were taught in the prerequisites programming classes. For instance, on classes, or the syntax of C++ language. Outside of that, I did learn quite a bit of new things after midterms. Furthermore, the programs - even the "easy" ones - were quite stimulating. I spent a lot of time on them. In addition, Professor Pisan uses clang-tidy, which has made me a better programmer overall.

3. lectures+practicing activities, assignments, reading homework

4. Yes, some concepts and examples were very useful in learning about problem solving.

5. Yes, because there are tons of newly introduced data structures, which plays around with nodes, where we need to carefully think where the node goes.

6. Yes it was. Yes it certainly stretched my abilities with valuable assignments and great in-class puzzles.

7. Yes, it challenged me more than any other class I've taken so far.

8. First half of course felt like it was a C++ (i.e. syntax/language details) course. CS content mostly review from 143. Only challenging assignment was the final assignment, mostly for time reasons.

9. I thought that learning about C++ and how memory management works was interesting. I think it would have been useful to compare C++ to other languages like Java - especially since so many people in the class had a background in Java. I think the data structures, algorithms and discrete math part of the course was somewhat neglected because there was so much focus on C++. I've been trying to do interviews for internships during this quarter, and I can't say that the course material was deep enough to be helpful for that.

10. Yes because it was difficult

11. Yes, it required different ways to look at something, and think through a problem in ways that I had not done before.

12. Yes the assignment stimulate

13. Yes, the mathematical aspects such as big-O and induction were very cool.

14. Lots of algorithm examples presented and given time to process and work them out individually

15. yes, i am learning new materials each week, and the assignment did make me think a lot about the material we taught in class.

16. Yes, the assignment projects incorporated the lectures from class very well.

17. yes

18. Yes, but I felt that demonstrations could have been better laid out. For example, the towers of Hanoi problem is common enough that many videos exist that are in the public that describe recursion in early accessible and digestible forms.

19. The new information I learned in this class included testing my programs on linux, writing script to compile, test, and run my programs, information about how programs are compiled, searching for language-specific implementation tips online, and C++ specific details about memory and syntax. I found this information very useful because it was unlike the information I learned in 110, 142, and 143. I was expecting to learn new information about data structures, algorithms, and descrete math, but most of what was taught in those areas, I had already learned in previous courses or through independent study.

20. I found the class to be intellectually stimulating mostly through the programming assignments because they encouraged us to approach problems in new ways.

21. Technical concepts were challenging, due to their abstraction, but I came away with a better sense of how to solve problems than when I started the course.

22. Yes it did!

23. I've already taken a similar class before so not really.

24. This class was extremely intellectually demanding, and required me to stretch my thinking due to the complex topics being discussed.

What aspects of this class contributed most to your learning?

1. Lectures.

2. The programs and the lectures

3. assignments

4. Going through problems together to find solutions.

5. The assignments

6. In-class questions and puzzles along with office hour questions.

7. The assignments.

8. I met a nice friend in this class :]

9. In-class exercises. I think these were useful for applying the material. I think it would be helpful to have more of them. I liked the LeetCode exercises, but I think it would be more useful to talk about general techniques for approaching these problems, rather than just going straight to a solution. For example, the maximum sum in an array problem would have been a great opportunity to introduce the sliding window algorithm and dynamic programming in general.

10. Lectures and in class exercises

11. The lecture contributed the most. The first few projects weren't quite as to data structures so much as it helped learn c++

12. Opened answers for the assignment

13. The homework and in-class challenges contributed most to my learning.

14. Good balance between individual and group work.

15. n/a

16. The assignment projects.

17. assignments and activities

18. Projects were the most useful, readings were less used.

19. I found great value in the programming assignments. They sounded easy in the requirements document, but in practice were fun and challenging.

20. The programming assignments contributed most to my learning in this class

21. Assignments were most beneficial.

22. I liked the assignments and they helped me prepare for the exams.

23. The practice work.

24. The homework and book

What aspects of this class detracted from your learning?

1. I guess jumping from learning the basics in lectures to applying them in advanced assignments was a little hard.

2. N/A

3. No. They're all good and important to learn

4. Some of the material in the midterm was not covered thoroughly enough in class, such as the pointer's question.

5. The midterm sample

6. Quizzes didn't really help my learning at all. The physical classroom location made it hard to focus with the odd layout of the room.

7. The difficulty of the Maze assignment.

8. Pisan is nice and seems to genuinely care about students, but not the best lecturer (perhaps average for CS instructors). Sometimes slow, and I found it difficult to keep listening.

9. * Reading quizzes weren't useful for learning the material: The questions often seemed to test whether or not you had memorized the author's exact wording (i.e, specific details on the Bag ADT, verbatim definitions of terms like "operation contract" which is often called an "invariant" in other DS&A courses, formula for number of leaf-nodes in an n-ary tree). * The bureaucracy of submitting assignments was painful: I used a laptop that runs a Linux OS for the classes' assignments, but it wasn't very helpful for getting the assignments to conform to the Linux Lab requirements. The clang-tidy, CMake, valgrind, gdb and clang++ versions that I used on my laptop were often different on the Linux labs. Before each assignment submission, it would be necessary to reserve 3 - 4 hours making sure to double-check all of these things so it would compile. * Assignment specifications not in one place Every assignment had requirements that were spread out between Canvas, the README.md of the assignment's GitHub repository, and the comments in each file. Having to hunt for requirements is really annoying. * Assignment specifications had many typos. There were some typos that persisted all the way from Assignment 1 to Assignment 4. * Assignment specifications sometimes changed Assignment 1 was painful, because there was a PDF that was posted that ended up having its requirements mostly changed. * Formulas in PowerPoints were often difficult to read Whenever we had math formulas in the PowerPoint slides, they were often painful and difficult to read because they were typed like this: (1 + 2 / 3^5) * x^2 / x^5 + 5 * y

10. Not much

11. The quizzes. Im not sure if there was a better way, but quizzes often ecouraged me to skim the book more than really read it.

12. Opened answers for the assignment (hard to determine which direction to study)

13. Sometimes I felt like we focused to much on the minutiae of c++ instead of data structures and algorithms.

14. Exams sometimes ask questions not related project material

15. n/a

16. Nothing I can think of at the moment.

17. none

18. Midterm, was the only way to show memorization of concepts. This made it disproportionately effect my final grade and made my feelings on over learning more negative.

19. We were given about two weeks for each programming assignment, but I did not have time to use the full two weeks and usually had to spend two full days to complete it. Due to my own time constraints, I was not able to spend time optimizing and thoroughly testing my code. This is no fault of professor Pisan or the class structure, but more my time constraints and the course load / work that I do outside of this class. I think I would have benefited greatly from more time to analyze and improve my code.

20. I don't think any aspects of this class detracted from my learning

21. None

22. Nothing did.

23. The mandatory participation attendance.

24. The instructor. Essentially a self-taught class.

What suggestions do you have for improving the class?

1. Adding an answer key for the practice midterms and finals would help a lot.

2. I think mainly the first few weeks felt much like review to me. Perhaps, before classes start, a "Get to Know You" quiz could be issued. This way, you can see where everyone is at involving their knowledge; and gear the review towards that.

3. clarify assignments. For example, banking assignment needs to clarify how errors are handled.

4. The guidelines for the exams need to be more specific so we can properly prepare.

5. Give a more detailed list of what to learn for midterm and finals

6. Maybe cut 1 assignment and quizzes and do weekly puzzles / challenges worth small points towards assignment grade.

7. Be a little more specific on what you are looking on midterm questions

9. * Reading quizzes weren't useful for learning the material: I would have preferred more challenging questions that were textbook agnostic and require you to apply the concepts, rather than questions that only required you to memorize the formulas or descriptions from the textbook. This would also make it possible for students to use other textbooks such as Algorithms by CLRS or Algorithms by Sedgewick if they so prefer. * The bureaucracy of submitting assignments was painful: I would consider looking into how UW Seattle does virtual machines for their classes. If you google "UW CSE Linux Home VM" you will find it. * Assignment specifications not in one place Keeping the assignment specification in one place (preferrably the Canvas page), rather than having students hunt for requirements would make it easier to meet the requirements of the assignment. *Assignment specifications had many typos Consider getting someone to proofread the assignment specifications. *Assignment specifications sometimes changed Don't post an assignment's specification until you're sure about the requirements * Formulas in PowerPoints were often difficult to read This is probably because Google Slides does not have an equation editor. I think it would be worth looking into using some of the Chrome extensions like EquatIO, or using a LaTeX compiler online like Overleaf to generate images of typeset equations that you can insert into your slides. * Other notes: I think the starter code in the GitHub repositories is an unnecessary crutch. Just having method signatures that students should implement is enough. It's hard to get familiar with how C++ programs should be organized if you don't make them yourself. I think it would be helpful to have some way to check your answers for the midterm exam and final exam practice, rather than having students to either come to office hours or wait until we discuss them in class.That, or have more practice problems for the exams rather than just the in-class exercises or practice exam questions (i.e, selected textbook problems)

10. No suggestions really

11. Please find a way to test things in linux more seamlessly.

12. More example coding question

13. Be less strict with c++ programming questions on exams, "spot the error" problems.

14. Expanded access to more practice work

15. spend more time in study, try to write the code instead of imagining it on your brain

16. Nothing I can think of at the moment.

17. More sample before midterm and finals

18. Create or find better demonstrations and examples for common algorithmic problems. For example video or physical demos. Also make class more project based and less dependent on high stakes test.

19. The only issue I came across with this class was that some of the questions my peers asked were not understood by the professor. As a result, the wrong question was answered multiple times and the students eventually accepted that they needed to find the answer elsewhere. This could have been from the professor not actively listening or asking clarifying questions, or it could have been because my peers were not being clear as they asked their questions. I am not certain. All my questions were answered clearly and completely.

20. I don't have any suggestions

21. May be worthwhile to have coursework that involves open-source projects, to give students hands on experience with messy code.

22. Its a great class! Maybe provide answers to practice exams.

23. N/A

24. Create better slides that students can reference when studying/reviewing.

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.