JHU - Krieger School of Arts & Sciences / Whiting School of Engineering
ASEN.2016.Fall

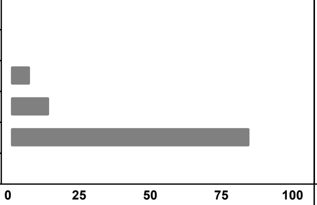
Course: EN.600.437.01.FA16: Distributed Systems

Instructor: Yair Amir *

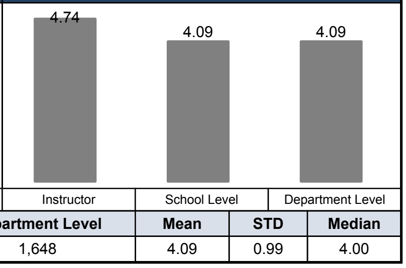
1 - The overall quality of this course is:

Response Option	Weight	Frequency	Percent
Poor	(1)	0	0%
Weak	(2)	0	0%
Satisfactory	(3)	2	6.45%
Good	(4)	4	12.9%
Excellent	(5)	25	80.65%
N/A	(0)	0	0%

Percent Responses



Means



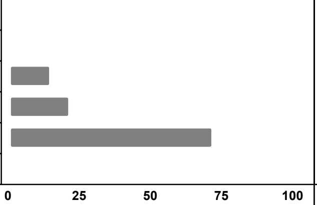
Return Rate	Mean	STD	Median	School Level	Mean	STD	Median	Department Level	Mean	STD	Median
31/35 (88.57%)	4.74	0.58	5.00	9,737	4.09	0.96	4.00	1,648	4.09	0.99	4.00

2 - The instructor's teaching effectiveness is:

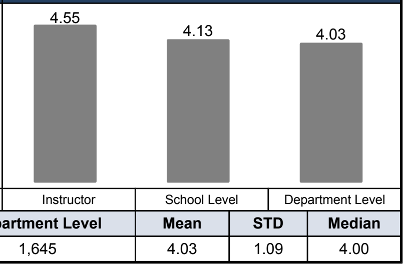
Yair Amir

Response Option	Weight	Frequency	Percent
Poor	(1)	0	0%
Weak	(2)	0	0%
Satisfactory	(3)	4	12.9%
Good	(4)	6	19.35%
Excellent	(5)	21	67.74%
N/A	(0)	0	0%

Percent Responses



Means

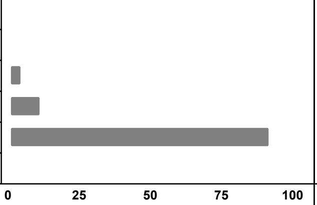


Return Rate	Mean	STD	Median	School Level	Mean	STD	Median	Department Level	Mean	STD	Median
31/35 (88.57%)	4.55	0.72	5.00	9,695	4.13	1.01	4.00	1,645	4.03	1.09	4.00

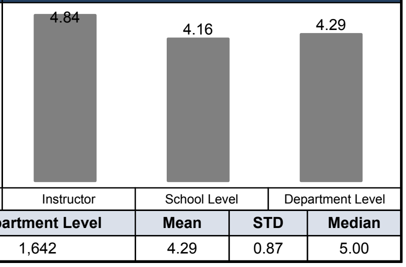
3 - The intellectual challenge of this course is:

Response Option	Weight	Frequency	Percent
Poor	(1)	0	0%
Weak	(2)	0	0%
Satisfactory	(3)	1	3.23%
Good	(4)	3	9.68%
Excellent	(5)	27	87.1%
N/A	(0)	0	0%

Percent Responses



Means

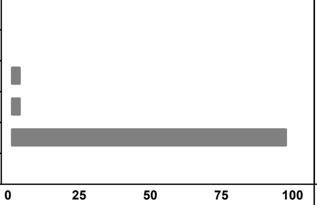


Return Rate	Mean	STD	Median	School Level	Mean	STD	Median	Department Level	Mean	STD	Median
31/35 (88.57%)	4.84	0.45	5.00	9,673	4.16	0.89	4.00	1,642	4.29	0.87	5.00

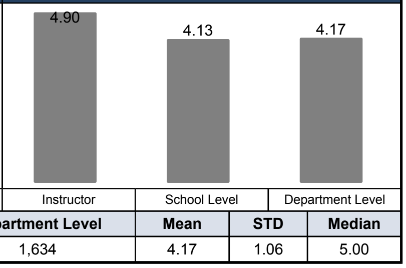
4 - The teaching assistant for this course is:

Response Option	Weight	Frequency	Percent
Poor	(1)	0	0%
Weak	(2)	0	0%
Satisfactory	(3)	1	3.33%
Good	(4)	1	3.33%
Excellent	(5)	28	93.33%
N/A	(0)	0	0%

Percent Responses



Means



Return Rate	Mean	STD	Median	School Level	Mean	STD	Median	Department Level	Mean	STD	Median
30/35 (85.71%)	4.90	0.40	5.00	9,659	4.13	1.04	4.00	1,634	4.17	1.06	5.00

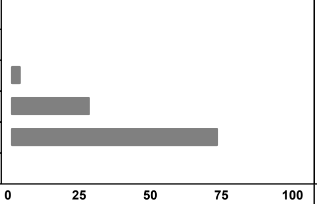
5 - Please enter the name of the TA you evaluated in question 4:

- Tom Tantillo, Amy Babay, Emily Wagner
- Tom Tantilo
- Tom, Amy, Emily
- Tom, Emily, Amy
- Tom tantillo
- Tom
- Tom
- Tom
- Tom, Amy, Emily
- Tom Tantillo, Emily Wagner, Amy Babay
- TOM
- Tom Tantillo
- Tom Tantillo
- Tom
- Tom Tantillo
- Tom
- Tom
- Tom Tantillo
- Tom Tantillo
- Tom Tantillo, Amy Babay, Emily Wagner
- Tom Tantillo
- Tom
- Tom Tantillo, Amy Babay, and Emily Wagner
- Tom Tantillo
- Tom Tantillo

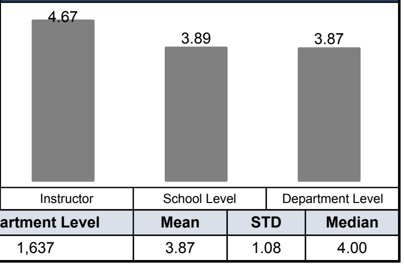
6 - Feedback on my work for this course is useful:

Response Option	Weight	Frequency	Percent
Disagree strongly	(1)	0	0%
Disagree somewhat	(2)	0	0%
Neither agree nor disagree	(3)	1	3.33%
Agree somewhat	(4)	8	26.67%
Agree strongly	(5)	21	70%
N/A	(0)	0	0%

Percent Responses



Means

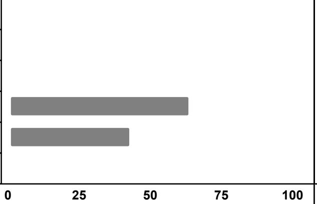


Return Rate	Mean	STD	Median	School Level	Mean	STD	Median	Department Level	Mean	STD	Median
30/35 (85.71%)	4.67	0.55	5.00	9,640	3.89	1.07	4.00	1,637	3.87	1.08	4.00

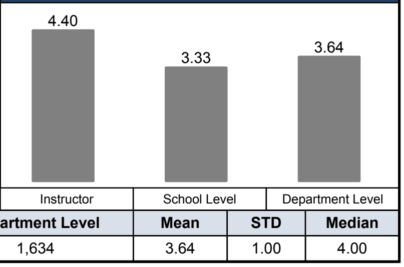
7 - Compared to other Hopkins courses at this level, the workload for this course is:

Response Option	Weight	Frequency	Percent
Much lighter	(1)	0	0%
Somewhat lighter	(2)	0	0%
Typical	(3)	0	0%
Somewhat heavier	(4)	18	60%
Much heavier	(5)	12	40%
N/A	(0)	0	0%

Percent Responses



Means



Return Rate	Mean	STD	Median	School Level	Mean	STD	Median	Department Level	Mean	STD	Median
30/35 (85.71%)	4.40	0.50	4.00	9,659	3.33	1.04	3.00	1,634	3.64	1.00	4.00

8 - What are the best aspects of this course?

- You get to build a lot of distributed system protocols from the ground up which gives the student a greater feel for how things work. Also the instructor/TAs put an insane amount of work into the class and they respond to emails very quickly.
- Excellent lectures, good exercises that teach a comprehensive concepts of distributed systems.
- The final project, although very long, was very fun to work on. The professor and TAs are very engaged with the students and the course, and definitely show that they care. I really liked the BFT/Cheating husband/PPcast stuff - do more of that!
- You can not only learn the theoretical part, but also have many opportunities to practice these and put what you learned in class into practice.
- It is an interesting class!!! The topic of this course is including many aspects of distributed systems. And the project is a little hard but worth to do it.
- rewarding and challenging
- Very passionate and knowledgeable teacher. Good material. Assignments were a good difficulty: not overly complex, but hard enough so that it pushed you to really understand the material.
- The course teaches you to implement real distributed systems, from the basics of multicast to an actual mail system.
- The projects are very useful
- course staff puts a lot of effort into project feedback lectures give historical background, intellectual progression, and currently developing applications of distributed algorithms which contextualizes class concepts in terms of real-world impact
- The course builds distributed systems from scratch, which is really a good practice.
- Yair is very passionate about the subject and does his best to convey everything we need to know about 21st century distributed computing. Because the course material is coupled with his own real world experiences, it's never to hard to forget the /relevancy/ of this course in industry.
- Students learn a lot from this course. The knowledge of this course is quite useful in real life. Course staffs talk with students about their designs before each project, which is pretty helpful.
- Thought provoking and unexpected insights.
- The professor is amazing and his philosophy of wanting students to come out of the course with the knowledge and confidence to build an enterprise ready distributed system really rang true.
- This course will teach you a lot from the ground up about distributed systems. You will be capable of building something real when you are complete with the course. It was also very cool to see work being done that affects things we use every day (like the power grid or CNN video streams). It was nice to see how this is relevant and how we can use this in the real world to build strong software.
- Excellent presentation PDF slides and good lectures in general, along with a few fun "games" played in-class. The TAs are very helpful and there is a lot of feedback. The mix of theoretical and practical assignments are intellectually challenging but engaging at the same time. The final project was really cool, and we are proud of what we did.
- The meeting for every design one on one is very helpful.
- Really cool topic, and projects that really make you think and learn the material. No tests! Professor was quite engaging and cared a lot about the success of his students. The professor wasn't afraid to share about his research, which was really cool too.
- Professor is awesome. Homework and final project are interesting. TA is nice and really helpful.
- The course leads us to develop creative solutions for problems in distributive systems. It is self-containing and doesn't need specific external knowledge to complete.
- Interesting field, very knowledgeable and approachable professor and staff

9 - What are the worst aspects of this course?

- Nothing
- The first two assignments seemed rather dry and I didn't feel we had the best guidance on what to do The feedback on the theoretical exercises weren't particularly useful Lectures were often dry Course was too large The professor wanted to incorporate a lot of aspects of research into the course. I agree with the principle, but it often seemed as if classes devolved into bragging about their group's research. Also as a result, I'm not sure what the "fundamentals" are for distributed systems
- It will be difficult for students who are not good at programming C/C++.
- not enough teaching on theory and analysis, lectures are not sufficient, sometimes hard to understand Yair
- None.
- Too much debug work
- concepts in lecture are not reinforced except for the theoretical assignments in the beginning of the course
- none.
- Sometimes he spends large portions of the class giving us life advice, and, while I'm sure it is useful to many in the class, I think it took away from precious class time. The professor sometimes does not realize when it's time to move on from a topic - he assumes too quickly that a student's' expression means that he doesn't understand the material, when, in fact, he's just waiting for the professor to move on. He also extends the deadlines too readily.
- Personally speaking I don't really like C compare to other object oriented languages.
- None.
- C is hard.
- The course is difficult - there is a lot of work in C, which can be a real challenge if you have no C/C++ programming experience.
- Some lectures are only briefly explained, especially the intrusion-tolerant parts. Homework takes a very long time to complete, and there are no late days.
- The professor could sometimes become condescending when asked questions that he thinks are dumb or irrelevant.
- Workload is heavy, slides are not useful as expected.
- Nothing to be specially mentioned.

10 - What would most improve this class?

- Nothing
- Getting to use the C standard libs. People who make distributed systems for real I imagine use this. I agree with the choice to not use Python More guidance on proj 1 and 2, or more applied projects like the final
- more transparency with algorithms and how they are proved
- Nothing.
- Add some review sessions
- more theoretical-type assignments to connect with lecture content
- Maybe it would help if the TA develop some functions encapsulating the lower level operations, and then the students can directly call these functions instead of implementing them.
- The theoretical exercises were incredibly helpful - maybe one more of those may have solidified some concepts. Doing maybe one or two check-ins for the projects (during class time perhaps?) might encourage students to do a better job on them/start on time.
- Maybe we can try to talk about some popular distributed frames?
- Providing a further readings list to expand on class material.
- Not much - I enjoyed the course, even though it was very difficult.
- Allow late days in the course. Have an assignment or two about the later part of the course.
- Better distribution of time for the 4 projects would have helped with the workload. Assignment 3 did not need so much time.
- Homework could be implemented in C++, which may easy the homework.
- Nothing to be specially mentioned.

11 - What should prospective students know about this course before enrolling? (You may comment on any aspect of this course such as assumed background, readings, grading systems, and so on.)

- This class will have one of the largest workloads you can have at Hopkins.
- Good background programming on c is needed.
- Prepare to spend a ton of time with your partner, so pick a good one :)
- You need to be good at C/C++ programming and think about problems logically.
- It uses C and linux.
- very, very challenging. huge time commit. worth every bit of struggle. incredibly rewarding. come prepared.
- Assignments are a bit time consuming, but can definitely be done in time if you manage time well.
- This course is a good amount of work. Coming in without any knowledge of socket programming may make the first assignments difficult, and can make for a steep learning curve.
- Have intermediate debug ability
- you will probably spend more time than you expect on some of the projects, so if you are not interested in systems implementation you will be frustrated with the amount of effort required spend time on detailed designs for the project and update your design document as you code
- Grades not curved.
- Start your assignments on time! This is a tough course so you need to stay on top of things. Knowledge of C is very important before starting class. Make sure your design documents for your projects are as detailed as possible BEFORE you start so that implementation goes smoother. Working with a partner can speed things up immensely.
- Students should be familiar with C.
- Projects can be long but are interesting.
- Only take this course if you are prepared to do a lot of work. You will need to seek office hours when you are lost, and homeworks take a lot of time (even with a partner). Only take it if you are confident you can handle the extra load, and you are confident you can program in C or C++.
- You need to spend maybe five days per practical assignment, so be ready for that! Intermediate C/C++ programming skills and some data structures knowledge are required, but nothing else. There is some networking code, but the instructors will teach you and they will give you a tool to use later on.
- If you are not that into distributed system, do not take it.
- This course is really difficult. It helps a lot to have a partner that you know you can work well with, since all of the projects are partnered. You should be familiar with C, and taken data structures. Other systems courses would be helpful too.
- All the homework are done in C language. Workload is heavy. Grade is strict, no curve.
- Good course for anyone interested in thinking through complex problems. All the theoretical works and exercises should be done with careful thoughts. Projects can only be done in C or C++ at Linux machines (ugrad) for best efficiency, while technical concepts like sockets will be covered in class. Because of the heavy workload, start your project early or wait until painful hours before deadlines.
- Knowledge of C or C++, Important to stay on schedule (don't fall behind), attend every lecture (attendence is part of participation grade)

JHU - Krieger School of Arts & Sciences / Whiting School of Engineering
ASEN.2016.Fall

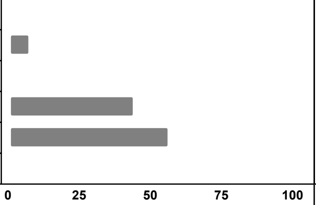
Course: EN.600.337.01.FA16: Distributed Systems

Instructor: Yair Amir *

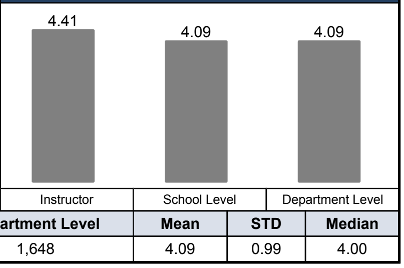
1 - The overall quality of this course is:

Response Option	Weight	Frequency	Percent
Poor	(1)	0	0%
Weak	(2)	1	5.88%
Satisfactory	(3)	0	0%
Good	(4)	7	41.18%
Excellent	(5)	9	52.94%
N/A	(0)	0	0%

Percent Responses



Means



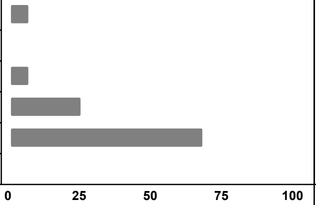
Return Rate	Mean	STD	Median	School Level	Mean	STD	Median	Department Level	Mean	STD	Median
17/18 (94.44%)	4.41	0.80	5.00	9,737	4.09	0.96	4.00	1,648	4.09	0.99	4.00

2 - The instructor's teaching effectiveness is:

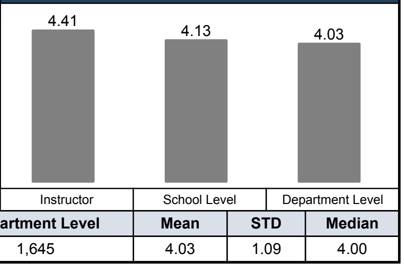
Yair Amir

Response Option	Weight	Frequency	Percent
Poor	(1)	1	5.88%
Weak	(2)	0	0%
Satisfactory	(3)	1	5.88%
Good	(4)	4	23.53%
Excellent	(5)	11	64.71%
N/A	(0)	0	0%

Percent Responses



Means

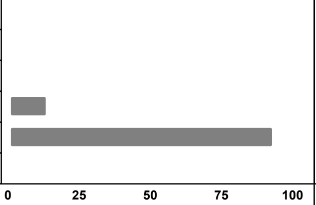


Return Rate	Mean	STD	Median	School Level	Mean	STD	Median	Department Level	Mean	STD	Median
17/18 (94.44%)	4.41	1.06	5.00	9,695	4.13	1.01	4.00	1,645	4.03	1.09	4.00

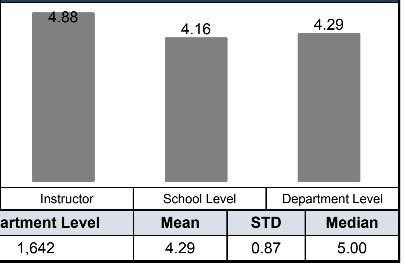
3 - The intellectual challenge of this course is:

Response Option	Weight	Frequency	Percent
Poor	(1)	0	0%
Weak	(2)	0	0%
Satisfactory	(3)	0	0%
Good	(4)	2	11.76%
Excellent	(5)	15	88.24%
N/A	(0)	0	0%

Percent Responses



Means

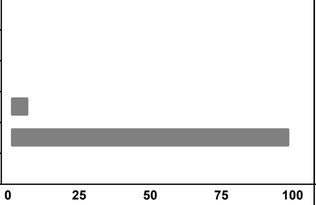


Return Rate	Mean	STD	Median	School Level	Mean	STD	Median	Department Level	Mean	STD	Median
17/18 (94.44%)	4.88	0.33	5.00	9,673	4.16	0.89	4.00	1,642	4.29	0.87	5.00

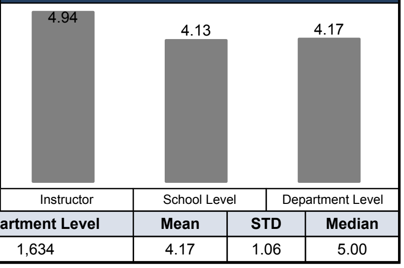
4 - The teaching assistant for this course is:

Response Option	Weight	Frequency	Percent
Poor	(1)	0	0%
Weak	(2)	0	0%
Satisfactory	(3)	0	0%
Good	(4)	1	5.88%
Excellent	(5)	16	94.12%
N/A	(0)	0	0%

Percent Responses



Means

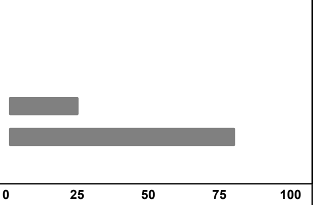
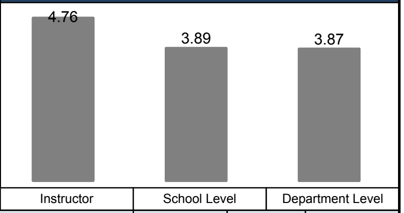


Return Rate	Mean	STD	Median	School Level	Mean	STD	Median	Department Level	Mean	STD	Median
17/18 (94.44%)	4.94	0.24	5.00	9,659	4.13	1.04	4.00	1,634	4.17	1.06	5.00

5 - Please enter the name of the TA you evaluated in question 4:

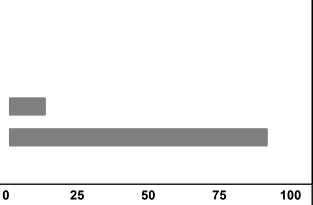
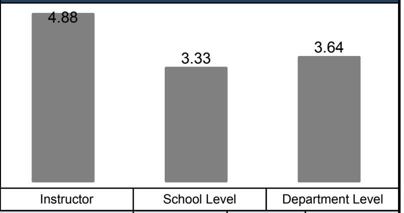
- Tom Tantillo, Amy Babay, Emily Wagner
- Tom
- Amy
- Emily Wagner, Tom Tantillo
- Tom Tantillo
- Tom Tantillo
- Tom Tantillo
- Tom
- Tom, Emily, Amy
- Tom, Amy, and Emily
- Tom Tantillo
- Tom Tantillo, Amy Babay, Emily Wagner
- All - Tom, Emily, and I forget the other woman's name
- Tom, Emily & Amy

6 - Feedback on my work for this course is useful:

Response Option	Weight	Frequency	Percent	Percent Responses	Means
Disagree strongly	(1)	0	0%		
Disagree somewhat	(2)	0	0%		
Neither agree nor disagree	(3)	0	0%		
Agree somewhat	(4)	4	23.53%		
Agree strongly	(5)	13	76.47%		
N/A	(0)	0	0%		

Return Rate	Mean	STD	Median	School Level	Mean	STD	Median	Department Level	Mean	STD	Median
17/18 (94.44%)	4.76	0.44	5.00	9,640	3.89	1.07	4.00	1,637	3.87	1.08	4.00

7 - Compared to other Hopkins courses at this level, the workload for this course is:

Response Option	Weight	Frequency	Percent	Percent Responses	Means
Much lighter	(1)	0	0%		
Somewhat lighter	(2)	0	0%		
Typical	(3)	0	0%		
Somewhat heavier	(4)	2	12.5%		
Much heavier	(5)	14	87.5%		
N/A	(0)	0	0%		

Return Rate	Mean	STD	Median	School Level	Mean	STD	Median	Department Level	Mean	STD	Median
16/18 (88.89%)	4.88	0.34	5.00	9,659	3.33	1.04	3.00	1,634	3.64	1.00	4.00

Course: EN.600.337.01.FA16: Distributed Systems

Instructor: Yair Amir *

8 - What are the best aspects of this course?

- This course is-- by far-- the most challenging, useless, and important class that I've taken at Hopkins. It's useless in that i'll probably never touch distributed systems with a 10 foot pole ever again, and it's essential in that it'll shape the way you think and approach writing programs--more like essays, meticulously planned out and mulled over, rather than quick, imperfect scripts constantly being rewritten and iterated over. The TA's are also phenomenal.
- Exceptionally challenging to the point of forcibly expanding your brain's intelligence
- great professor
- the material and projects
- You will learn a lot by doing this course. The assignments, particularly exercise 2 and the final project will give you a good understanding of distributed systems at a practical level. In addition, the lectures are interesting and give a good background in the state of the art in the field. Yair and his teaching staff are very competent and helpful.
- Most of the projects were highly challenging and created a very solid understanding of the material. The level of feedback on my work was higher than any other course I've taken in the department.
- Interesting material, practical assignments
- The professors and TAs put in a lot of effort to help students understand the material and accomplish the tasks.
- You will learn a lot. The material is actually very interesting (once you understand it)
- The course gives an extremely in-depth look at distributed systems. A lot of time and effort is put into each assignment, on both ends. The assignments for the course take an extremely long time, but they are all followed through with Prof Amir and the TAs giving as much guidance as they possibly can. The TAs were especially helpful - they always had an answer that was comprehensive and useful, and were very quick to respond to emails. Overall, this course was one of the most rewarding courses I've ever taken, if only because I had to devote so much time and energy thinking, talking about, and programming the material.
- Learning a skill that's very in-demand right now.
- I improved immensely as a programmer and as a student. As a coder, I became more comfortable with programming in C and understanding how complex algorithms are implemented. I also learned how effective and important planning prior to coding is. As a student, I learned how to effectively prioritize my time to work on large projects while still keeping up with work for other classes. More than that, the professor and the TAs are amazing. The professor, Yair Amir, truly cares about his students, and is interested in not only helping them perform better in the class, but also to grow as people. The TAs put an immense amount of time into answering questions and helping students with the assignments.
- This course brings students close to the cutting-edge of protocol design.
- Two Words: Yair Amir. Yair is fantastic. I have never before met a man who cares so deeply about his students and his research area. After spending a semester with Yair, it is abundantly clear to me why he is the best person to act as Chair of the Computer Science department. After only a few lectures, he knew everyone by name (including even the disrespectful students that sat in the back and didn't pay attention) and wanted to engage as many people as possible in lecture. If you did poorly in distributed, then then you really need to look deeply at yourself, because Yair does absolutely everything in his power to help his students succeed. I'm convinced that Yair is secretly a superhero, because somehow he manages to balance the massive workload that he takes on, yet appear as if he has dedicated all of his time to the course. Yair is everything I aspire to be as a productive adult, computer scientist, and teacher. By the end of the semester, he had invested hours of his time into helping each group individually, and it really showed. He along with his TAs spent over a week of non-stop meetings to evaluate and discuss the final projects in the course. This level of dedication is unparalleled to any I have seen in my nearly two decades of being a student. I get little sleep and I suspect I incessantly whine about it, but Yair must get none given all of the responsibilities he assumes, and somehow he manages to still be incredibly prepared and attentive. Yair Amir is an incredibly impressive man. I have the utmost respect for his work ethic, teaching ability, and knowledge. Thank you so much for such an amazing semester, Yair! I must also say that the course TAs, Tom, Amy, and Emily are fantastic as well. They are incredibly helpful and approachable and gave great feedback and support throughout the course. They too indeed contributed significantly to how fantastic the course was.

9 - What are the worst aspects of this course?

- I wanted a "challenge" when I enrolled in this course. I've since found it a bit like drinking water from a garden hose. This course is extremely difficult. It is demanding. It is soul-sucking. It probably takes an an average of 5 years off a student's life. It is depressing how much you'll realize you don't know. It is also imperfect. Yair is quite capricious and his intelligence is very intimidating; I think concepts appear obvious to him and I believe that it results in a difficult classroom environment for students to ask questions in. The assignments are also vaguely worded, and while I understand that "real-world" distributed systems problems are full of ambiguity, a more thorough introduction to networking and its syntax would be beneficial for students who lack a background in systems (of which there are a sizable number).
- Is only 3 credits...should be 5.
- it's a lot of work
- This class requires a significant time investment. Yair makes it a point to emphasize this at the beginning of the class, and you should listen. Make sure you start projects early, especially exercise 2 and the final.
- The workload fluctuated a lot over of the semester, and was especially high around the time of project 2 and the theoretical exercises.
- Time commitment
- Lectures could get a little tedious and classes sometimes ran overtime.
- It's extremely difficult, and is only made more difficult by the way it is taught. Yair clearly understands the material really well, and I often felt as if he wasn't able to dumb it down enough to teach it to students who were unfamiliar with the topic of distributed systems. The assignments were often vague and expectations were not specified. Also, I felt uncomfortable asking questions in class because when Yair addressed questions he would often interrupt the student halfway through the question and/or give some sort of condescending response.
- This course requires a huge time commitment - there are no gaps in-between assignments since you'll want the most time available for each one.
- Professor Amir is a poor instructor. I say this after much deliberation, having now taken two courses taught by him (the previous one being intermediate programming). He certainly knows the material, but he gets visibly frustrated any time students ask questions, and then he implies that they aren't trying hard enough. I found attending his lectures very frustrating. There were several occasions in which he interrupted students before they could finish their questions, to the point where I couldn't piece together what the student was asking before they were cut off. Professor Amir asks for feedback at the end of the course, but any time someone provides some feedback, he gives a long and drawn out explanation of why he disagrees with them, and the students aren't given a chance to respond. He refuses to allow students to use the standard libraries of the C++ language, making us implement basic data structures like linked lists from scratch. I can say with certainty that implementing these data structures for the homework assignments was a distraction from implementing the actual distributed systems-specific code. That said, he is passionate about the material and cares about the students succeeding. It's just that this passion gets in the way of him being an effective instructor. He is proud of his lab's work, but that's no reason for him to spend about a quarter of each lecture displaying charts that compare his lab's work to others', especially when the previous 45 minutes of conceptual explanations were difficult to follow and no text is provided to supplement them. I left lectures feeling like I hadn't learned much and like I was at fault for that being the case. However, most of the classmates I spoke to appeared to feel similarly, except for the ones that had a preexisting interest in distributed systems and had (I assume) read lots of literature on the subject before attending the class.
- In some ways, this class becomes a lifestyle. There are days where the distributed systems exercise will be the only thing you work on. Despite this, you may still find it difficult or even impossible to succeed. You should know what you're getting yourself into. If you've never used C, don't take this course. If you've used C but aren't the best at it, don't let that deter you. Talk with Yair about it and he will give you what you need to become a good enough C programmer. That being said, I was pretty bad at C and still did well in the course because I have a strong understanding of the concepts that were taught in the course. It's more important to understand the concepts taught than to be able to code something that works; on the final project, even though my program didn't work entirely as it was supposed to, I still did better than some other teams that had working projects with inefficient designs.
- The vast majority of my time spent on this course was dealing with frustrating minutia.
- There are none. At all. Period. End of discussion. I cannot see what anyone else wrote for this section, but whatever it is I can confidently say it's 100% wrong.

Course: EN.600.337.01.FA16: Distributed Systems
Instructor: Yair Amir *

10 - What would most improve this class?

- Being more patient with students, being more receptive to answering questions thoughtfully, candidly, and without indignation, holding class so that it adheres to the allotted time slots and does not consistently go over time which might be unfair to students with classes afterwards.
- N/A
- More details about using spread for the final project.
- A nicer classroom! It was difficult to focus in the windowless room for the entire 75 min. lectures.
- Late days for assignments
- Having more ugrad servers to test the projects to avoid traffic.
- More guidance (at least with the first project and when Spread is introduced); a friendlier classroom environment; getting rid of those days where the class would be separated into 4 groups and a TA or Yair would lead each of those groups in a discussion about design (they were extremely chaotic and seemed like a waste of everyone's time); clearer expectations for the projects; adding diagrams/designs/demonstrations to lecture slides to teach all of the different protocols
- I felt that the theoretical assignments were not especially difficult to think through, but required a sort of specific structure that wasn't defined that well in class, especially for the very first one.
- I don't think Professor Amir should teach this course unless he makes some drastic changes to its structure and to his lecturing style. I have never come close to being this disappointed and irritated by a Hopkins CS course. I will not recommend this course to any classmates. I am strongly considering meeting in person with Professor Amir to express this disappointment in person, since I perhaps owe it to him after speaking so ill of his teaching style.
- For me, probably a crash course in C at the start of the course. That being said, Yair did ensure that grad students from other schools were skilled enough with C by teaching them before the start of the course; however, as an undergrad, he assumed I was skilled enough in C from Intermediate Programming. Normally this would be true, but I was very out of practice with C and was never the best with it to begin with.
- Provide more methods to the students for standard operations in c. I think it is ridiculous that we had to implement circular arrays and linked lists ourselves. There are other similar tasks that I learned nothing from and was overall a large waste of my time.
- I can't think of a way to improve this class. However, you could improve other classes by cloning Yair and have him teach everything.

11 - What should prospective students know about this course before enrolling? (You may comment on any aspect of this course such as assumed background, readings, grading systems, and so on.)

- There are few things more potent, more painful, than a distributed systems assignment. Doing one is a lot like floundering in a vast sea of confusion, with several rocks tied to your ankles for good measure. On the other hand, there are fewer kinds of ecstasy than getting one of those programs to work. Yair and his team are brilliant, and they will give you guidance as you explore the lowest of lows you will ever feel during your time at hopkins. You will both curse them (for inciting such horrifying assignments on you) and thank them profusely (for sitting with you for hours debugging your program). Honestly, distributed systems is a different beast altogether. It will consume your life, THOROUGHLY. It will ask the world of you, and the world you will gladly (or grudgingly) give. But if you and your partner (and make sure you have a good partner, or else you'll both be miserable and sad) make it out, you’ll have an appreciation for the people who actually make these massive, intricate, improbable things work. I don’t know how much genius is present on this campus, but it is in abundance in the dsn lab. Enroll with extreme caution, but, should you choose to do so, know you will suffer and learn more than you ever have— and probably ever will— during your time here. This class is truly the stuff of legends.
- Students should be prepared to put lots of time into assignments, but it will be a rewarding experience!
- it's a lot of work. if you put in the work you learn a lot
- You should only take this class if you are certain you will have the time for it. Make sure to brush up on C/C++ beforehand. Don't be shy to ask for help in this class.
- Huge workload!
- Projects take a significant amount of time but teach the material well. Be prepared to spend a lot of time on the course.
- This class will take a lot of your time. Be prepared to spend time on it.
- This class is HARD. Come in with a good partner (all the coding assignments are partnered) and with the expectation that this class will consume your life for the semester.
- If you decide to take the class, it will be one of the "main" classes of your semester - it's requires a large time commitment, so you'll need to really have an interest in learning distributed systems. The assignments, in my opinion, require the work of 2 people, even though you are allowed to work by yourself. So you'll not only need time to program, but you'll also need to be available to work with a partner often. Also, all the exercises are written in C, so you'll need a background of it to do the course.
- You should be familiar with C. It greatly helps to have taken algorithms and discrete math. An amazing thing about this class is that despite getting graded harshly on practical exercises and theoretical assignments all semester, you can still end up with a good grade because Yair determines grades based on how much he perceives you've learned from the class. If you put the effort in, you will most likely be fine. Just don't enroll in this class expecting an automatic A (it's definitely possible to get an A, but you have to be a very strong C programmer and good at understanding complex concepts).
- A strong C or C++ background is a MUST. Some network programming experience helps, but is certainly not required. No required readings. There are four large coding projects and two theoretical written homeworks.