CS 3240-100 **Advanced Software Development** - Spring 2010

ENGR (10321)

INSTRUCTORS: **Sherriff, Mark (mss2x)**

Respondents: 56 / Enrollment: 104

Summary: CS 3240-100 Advanced Software Development - Spring 2010 (10321)

Overall Course Rating

CS-3240-100 Mean 3.83
CS-3240-100 Std Dev 0.87
CS-3240-100 Response Count 280

Difference from Category Mean, Expressed in Category Standard Deviations	-2 -1 0 1 2
	-0.24

SEAS, 3000-level courses Mean 4.05
SEAS, 3000-level courses Std Dev 0.91
SEAS, 3000-level courses Response Count 9407

Overall Instructor Rating

INSTRUCTOR: Sherriff, Mark
Mean 4.38
Std Dev 0.72
Response Count 391

Difference from Category Mean, Expressed in Category Standard Deviations	-2 -1 0 1 2
	0.26

SEAS, 3000-level courses Mean 4.14
SEAS, 3000-level courses Std Dev 0.88
SEAS, 3000-level courses Response Count 15550

~ QUESTIONS AND DETAILS ~ | *~ ANSWER MATRICES ~*

1. How accurate is this statement for you: The project helped me better understand the phases and intricacies of software development.

~
Question Type: Likert
~
contributed by Sherriff, Mark (mss2x)

Results for CS-3240-100, Sherriff, Mark

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
55	4.31	0.66	23 (41.82%)	26 (47.27%)	6 (10.91%)	0 (0.00%)	0 (0.00%)

Results for SEAS, 3000-level courses

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
55	4.31	0.66	23 (41.82%)	26 (47.27%)	6 (10.91%)	0 (0.00%)	0 (0.00%)

2. How accurate is this statement for you: The project was of acceptable difficulty.

~
Question Type: Likert
~
contributed by Sherriff, Mark (mss2x)

Results for CS-3240-100, Sherriff, Mark

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
56	4.16	0.76	19 (33.93%)	29 (51.79%)	6 (10.71%)	2 (3.57%)	0 (0.00%)

Results for SEAS, 3000-level courses

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
56	4.16	0.76	19 (33.93%)	29 (51.79%)	6 (10.71%)	2 (3.57%)	0 (0.00%)

3. How accurate is this statement for you: The project was of acceptable length.

~
Question Type: Likert
~
contributed by Sherriff, Mark (mss2x)

Results for CS-3240-100, Sherriff, Mark

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
56	4.23	0.69	20 (35.71%)	30 (53.57%)	5 (8.93%)	1 (1.79%)	0 (0.00%)

Results for SEAS, 3000-level courses

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
56	4.23	0.69	20 (35.71%)	30 (53.57%)	5 (8.93%)	1 (1.79%)	0 (0.00%)

4. Which topic/lecture in this course was your favorite and why?

~
Question Type: Short Answer
~
contributed by Sherriff, Mark (mss2x)

Results for CS-3240-100, Sherriff, Mark

Total	Individual Answers
38	*See below for Individual Results*

Powerpoint karaoke because it was awesome.

~ QUESTIONS AND DETAILS ~	~ ANSWER MATRICES ~
	Didn't really like the lectures until the end
	Powerpoint karaoke. It will be invaluable in my career.
	Risk Management. It was interesting.
	Lecture karaoke!
	Agile vs plan driven, it helped to turn two buzzwords into well defined processes and laid down when to use on over the other.
	design patterns because I always thought about these kinds of patterns in an abstract way in my mind when brainstorming how to go about a particular programming assignment but never heard of them as actual design patterns. it was a nice eye opener and was cool to learn about the different approaches to programming and their advantages and disadvantages.
	Not counting powerpoint karaoke, I really enjoyed talking about plan-driven vs. agile development.
	Static Analysis: the design and pattern recognition for the software to find bugs within a system was interesting to me
	I really liked the powerpoint karaoke. Awesomeness in one word! :D
	POWERPOINT KARAOKE, followed by the tangent about the StarPony(R)
	Powerpoint karaoke. This should be something that is done for the whole department, and put on as a show sometime. It was a ton of fun for the audience, and it looked like the presenters were having fun too. As far as the actual lectures, I really enjoyed the INOVA team coming in and giving a lecture on real-life applications of the scrum technique.
	I enjoyed hearing about what software development in the real world would be like
	Both projects were very fun. It was my favorite course.
	Talking about different methods of agile development and how people use these new innovative ideas that I've never thought about in order to work effectively to finish projects.
	I liked design patterns. Seeing things that are directly applicable was the most helpful.
	Besides "Powerpoint Karaoke" (of course), I liked the "Survey of Agile" lecture since it discussed the processes that they place in industry.
	Not one in particular.
	The agile stuff seemed the most interesting to me, particularly because it seemed most applicable / useful to me given my style of effort.
	Looking back, I liked the topics that had to do with requirements elicitation and such. Those topics helped me understand the business and social side of development that is often forgotten in CS.
	My favorite was looking at the different testing programs (DJUnit, MUclipse ect.). I thought the different testing programs were really interesting. I think I liked them because they kind of brought up some interesting theoretical point about testing, which I find more interesting that the less theoretical, more practical stuff.
	powerpoint karaoke - self explanatory
	The extra topics at the end, they addressed things that people were interested in and were well covered.
	powerpoint kareoke! :)
	Powerpoint Karaoke, enough said
	The lecture on human-computer interaction was pretty interesting, because I realized the importance of having good interfaces from that lecture
	powerpoint karaoke
	Risk management was probably pretty useful; I don't have a strong opinion on any of them.
	Testing. I learned about tools and techniques I hadn't heard of before
	Requirements analysis. I thought it was the most applicable.
	Power point karaoke. Entertaining.
	I like building the robot. Most of the topic an the lecture, I already know it before, so it's not that much fun.

~ QUESTIONS AND DETAILS ~	~ ANSWER MATRICES ~
	My favorite lecture was Powerpoint Karaoke :). Other than that I really enjoyed the lectures pertaining to team management, because we don't learn that in any other class. I honestly don't remember much about the lecture component of this course. Favorite Lecture: PPT Karaoke because it was a great way to end the semester. John Knight is a 'must-have-back'. I also found the lectures on testing to be interesting especially when you showed the different eclipse plugins. Team management and risk lego robots...because legos are awesome Other than ppt karaoke, probably testing I most enjoyed the guest lectures from the LED clock company (sorry forgot the name). It was interesting to get the perspective of an actual company rather than just reading about certain methodologies.
5. Which topic/lecture in this class do you think you will find the most useful in the future? ~ Question Type: Short Answer ~ *contributed by Sherriff, Mark (mss2x)*	Results for CS-3240-100, Sherriff, Mark Total: 37 Individual Answers: *See below for Individual Results*
	The most useful thing from this class is learning how to manage code when you have a Corporation of 8 people. Jeez, probably all of them. Team management Which programming language to use Risk Mitigation I really like the lecture on how to choose which programming language to use for different projects, because I think it is a very useful skill to learn. I do think that lecture is too brief and should have more details. Again, risk management seems pretty relevant in Reality. Requirements analysis. I think the most useful lectures for the future were the ones dealing with the design phase. I don't really see myself being the one to go and elicit requirements and feel like I will be more heavily involved in the design stages of development. I think the lecture on which programming language to use was very useful probably the design patterns. those patterns can be applied to any programming project and have a huge impact on the progression and outcome of the project documentation and teamwork The class in general was very useful to me, mostly due to the fact that this class taught me how to design and develop a large scale project. No comment. Really depend on what will you do in the future. Design patterns, more than anything else. Software life cycle...? Agile development, for the above reasons Agile development, because its a fad in software development right now and gets products to market quickly. Not one in particular. The lectures on risk management were helpful. More likely than not, the entire project to work on is a large risk built of smaller risks. Things seem to go wrong in every project, so mitigating the risks are crucial. PPT Karaoke

~ QUESTIONS AND DETAILS ~	~ ANSWER MATRICES ~
	We always go over it, but the types of programming and how they are used (agile and plan-driven). Both have their applications for projects, and I can see companies using both types depending on what they are working on. Requirements Engineering. It was good to learn how to think about everything that is really involved in the project. I may have even taken it a bit far in designing the project, but it ended up clear, and easy for everybody to figure everything out. The agile development methods. The group work for the project the lectures we spent on requirements will probably prove to be very useful, especially the ones on requirements elicitation where we got to practice during class Probably the design lectures, since they deal solving the problems rather then understanding the problem or making that solution a reality. When we brought in the fols from Innova to talk abotu Scrum, I thought that was the most useful. Requirements elicitation Risk Management and Requirements Elicitation: having worked a few internships, these areas are very crucial towards successful software development. testing how do do the documentation The corporation work was very rewarding because it actually forced us to use asynchronous techniques. Requirement elicitation. In general, mentioning features in IDEs was helpful Powerpoint karaoke Powerpoint karaoke
6. What lecture/topic(s) in this class "did not work" or were not seen as useful in the long run? ~ Question Type: Short Answer ~ *contributed by Sherriff, Mark (mss2x)*	Results for CS-3240-100, Sherriff, Mark Total: 29 Individual Answers: *See below for Individual Results*
	I don't think is any. no comment I'm not convinced the more esoteric testing stuff (like, mutation) would be all that relevant. But, generally, this class seemed pretty on the mark, though I hate documentation. Design Patterns Requirements elicitation, mostly because the big take-home seemed to be, "it's difficult, but try your best." The lectures on the different kinds of software development (Agile, etc.). I do not believe that any team chose a type and stuck with it. They sort of set up the project as it came, and did not do a lot of fore planing. This was the case even if one of the team members tried to encourage the rest of the team to do so. To much focus on extreme programming/scrum/whatever... drop all the terms and get to the meat of things. There was a lot of time spent covering requirements. I feel like we were sort of beating a dead horse when we finally moved on. A more even discussion of all phases would have been better. We spent a really, really long time on requirements elicitation. I understand how important it is, but we probably didn't need as many lectures on that. They all fit with the general scope of the class. Using the crystal meth joke multiple times

~ QUESTIONS AND DETAILS ~	~ ANSWER MATRICES ~

N/A

Not one in particular.

Such a high level overview of design patterns was not very useful.

I think we could have stood to hear more about Software Maintenance. I thought it was brushed through pretty quickly.

Requirements elicitation, it's a skill most of us already understand.

none

Many of them. Although Prof Sherriff presents the material really well and in an interesting manner, I have heard it all three times before. It is the same information presented in 201, repeated in 216 and repeated again 4240. I am just tired of it by now, the only reason I came to class was that Sherriff was interesting and I wanted to hear about the project we were working on.

Some lectures about the minor and major projects are pretty useless.

everything worked well

The ones I can't remember.

Felt like testing was too much of a recap

Stuff that we had covered in other classes (CS 201, 216, etc.)

Agile vs plan-driven maybe could be approached differently to avoid "good" and "bad" connotations... as Professor Sherriff remarked in lab (when people spoke with disdain about either approach) that it needs to be clear that they are two valid solutions depending on the specific development paradigm.

"What type of programming language to use". It just seemed there wasn't much in the lecture that wasn't common sense.

None really

I didn't think the Design Patterns were presented well, and were slightly confusing.

Only the classes where students asked retarded questions about the requirements for 50 minutes. That's the kind of thing that should be done during lab time

none - all of the lectures seemed useful and like the material will prove to be useful at some stage in the "real world", especially for us going into software development. Some of the material might not seem as useful to people not going into software development since it's kind of specific, but it is a software development class, and Professor Sherriff taught us well about that subject

7. How accurate is this statement for you if you used the podcasts from this class: Podcasts were useful to catch up on material that I missed due to absences.

~

Question Type: Likert

~

contributed by Sherriff, Mark (mss2x)

Results for CS-3240-100, Sherriff, Mark

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
54	3.86	1.08	8 (14.81%)	5 (9.26%)	8 (14.81%)	0 (0.00%)	1 (1.85%)	32 (59.26%)

Results for SEAS, 3000-level courses

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
54	3.86	1.08	8 (14.81%)	5 (9.26%)	8 (14.81%)	0 (0.00%)	1 (1.85%)	32 (59.26%)

8. How accurate is this statement for you if you used the podcasts from this class: The podcasts were useful to review material that I was unclear on.

~

Question Type: Likert

~

contributed by Sherriff, Mark (mss2x)

Results for CS-3240-100, Sherriff, Mark

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
56	3.62	1.02	4 (7.14%)	8 (14.29%)	7 (12.50%)	1 (1.79%)	1 (1.79%)	35 (62.50%)

Results for SEAS, 3000-level courses

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
56	3.62	1.02	4 (7.14%)	8 (14.29%)	7 (12.50%)	1 (1.79%)	1 (1.79%)	35 (62.50%)

~ QUESTIONS AND DETAILS ~	~ ANSWER MATRICES ~

9. How often did you listen to the podcast for a lecture?

~

Question Type: Multiple Choice

~

contributed by Sherriff, Mark (mss2x)

Results for CS-3240-100, Sherriff, Mark

Total	Every lecture (NA)	Nearly every lecture (NA)	Whenever I needed to review a topic (NA)	Only when I missed a class (NA)	Randomly just to see what it was like (NA)	Never (NA)
55	0 (0.00%)	1 (1.82%)	9 (16.36%)	6 (10.91%)	4 (7.27%)	35 (63.64%)

Results for SEAS, 3000-level courses

Total	Every lecture (NA)	Nearly every lecture (NA)	Whenever I needed to review a topic (NA)	Only when I missed a class (NA)	Randomly just to see what it was like (NA)	Never (NA)
55	0 (0.00%)	1 (1.82%)	9 (16.36%)	6 (10.91%)	4 (7.27%)	35 (63.64%)

10. Do you have any suggestions/comments that we should take into account for future projects for this course?

~

Question Type: Short Answer

~

contributed by Sherriff, Mark (mss2x)

Results for CS-3240-100, Sherriff, Mark

Total	Individual Answers
32	*See below for Individual Results*

Please make the project relevant to the course. You keep saying that it is and you keep brushing off any and all criticism that it is not, but I can PROVE that it is not. Every time we apply the course material to the project (all those documents), it holds us back and prevents progress. That is a VERY CLEAR indicator that the project does not match up with the course material. If a math instructor gives an arithmetic problem and insists that students solve it with calculus, SOMETHING IS WRONG. My suggestion is this: drop the LEGOs and assign a huge software project to the entire class. An endeavor of this scale would necessitate the use of all that stuff you talk about in lecture. As it stands now, I just zone-out in lecture and ignore the Course Pack, because we don't need the material for the project and aren't being tested on it (aside from a pathetically-easy midterm, worth only 5% of our grade). I don't think that the current project demands an unreasonable amount of effort, but it is just a poor choice for this class.

Tone down on the length/frequency of presentations

It was just perfect projects for me.

Although it is true that more people can do more work than less people, it doesn't exactly scale, since there can be a lot of overhead involved in coordinate amongst a lot of people. It is true part of this class is about learning firsthand and secondhand how to minimize that overhead, but is it false to say that it won't be there.

In honesty these projects worked pretty well, but there was a bit of a breakdown; if you weren't already good at Lego Mindstorms, there was a surprising learning curve for just building an NXT bot. Conversely, if you were a Lego rockstar, you were at an advantage. Maybe provide some baseline chassis instructions?

While there's a lot to be learned from using Lego robots, not being allowed to take the robots outside of Olsson was a real limitation. Future projects should involve components that can be worked on outside of the lab. Overall, the project introduced us to more Java, Bluetooth, Android and phone development, and other peripherals. Future projects should definitely use those technologies, but the Legos really aren't as relevant.

Using outside devices such as the Droid phone was pretty neat

None, the checkpoints forced us to adhere to goals which was important.

I feel like the two projects we have this semester involves too much lego robot building. My team members are really inexperienced in building lego robot, and it is really time consuming and frustrating for us to build these robots with all of these features. I will like the robot to focus less on robot building and more on programming. Also, the amount of code needed for these projects are really small. The documentations needed for the major project really taken away a lot of time from my team in working on the major project. This caused my team to have very little progress during the first few weeks of the project.

Don't require us to use a device that is not avaliable to work with for the fist 3/4ths of the project and with no real connection to the project.

Better lecture on Android...

Teams = wonderful idea Corporation = terrible idea

Ditch the robots. This is a course about software engineering, not troubleshooting hardware that just refuses to work.

~ QUESTIONS AND DETAILS ~	~ ANSWER MATRICES ~

Stricter checkpoints??? It would have been nice to have my team working as hard as I was throughout the project (or just working more so I didn't have to work so hard.)

Both projects were a ton of fun, and I thought were well thought out. I don't know why so many corporations failed the Major Project final run (mine included, though I obviously know why in our case), as it seemed there was ample time. We were finished, we need to do integration, if we had a team that actually worked on their project, we would've been good.

Organize/Plan more. Too much documentation on things we haven't started yet, presentations on things we haven't started yet.

I don't like having to work with the phone. That was more annoying than anything.

Should have a little more concrete lecture.

Tell people to learn about their partners and really define their roles. That wasn't really given emphasis.

Let us take the robots out of the lab Try to even out the part boxes before assigning them to teams, some got shafted and others struck gold The android component seemed like it was completely superfluous and yet was worth the plurality of the points for the final project grade.

I LOVE YOU PROFESSOR SHERRIFF. TEACH MORE CLASSES. ALL THE CLASSES.

Consider providing sample networking code, and make the other programming challenges harder. Having the big challenge of the course be networking is kind of a drag.

Team division needs to be planned out better, coding and LEGO experience need to come into account just like personalities. The major project should be due before Finals start otherwise the people who procrastinate (almost every student in the class) will continue to wait until the last minute when other teammates are no longer willing to work on the project because other classes have finals they need to study for.

The progress checkpoints were a good idea for the major project. Documentation was a lot of work, but worthwhile.

Either go full-out with android or abandon it. Having just a little part of the project be android makes an effort/points ratio that is < 1. Which kinda makes me not want to do the android part at all.

The course's quirks are also real life problems in the software industry, any flaws I've noticed (software compatibility issues, team communication issues, difficulty) all occur and must be accounted for in real life.

Evaluate code as well as performance. Have same check points for each group. Setting different check points for different groups where the check points are worth 30 points of your grade is unfair. I had a friend who's group had nothing working until the morning of the final exam period and my group was working all along and worked a lot better than the friend's major project group and yet we got the same grade.

The project was acceptable.

I think the projects were great. Challenging yet fun. When you describe your stance on the projects in class I think you hit the nail on the head. Keep with it

Ensure teams are balanced. My first team had 3 women that did not do any work. They wasted time and effort. They lost some of my parts. They shouldn't get anything above a D in this class. Yes, I realize this sounds angry, however, I coded 95% of the minor project. That should have been no more than 40%. On top of this, the final part of the minor project changed during exam day. The board warped and my robot couldn't complete the course, even though it made it through the night before during test day. I really hate my team. They didn't do jack. That said, my major project team was the best I've seen. And my old team failed the major project since I was reassigned.

Don't use Android! Possibly make it three teams working together?

loved the project.

11. During the project, how many hours per week did you dedicate specifically to project work?

~

Question Type: Multiple Choice

~

contributed by Sherriff, Mark (mss2x)

Results for CS-3240-100, Sherriff, Mark

Total	0-2 (NA)	3-5 (NA)	6-8 (NA)	9-12 (NA)	13-16 (NA)	17 or more (NA)
56	0 (0.00%)	25 (44.64%)	20 (35.71%)	10 (17.86%)	0 (0.00%)	1 (1.79%)

Results for SEAS, 3000-level courses

Total	0-2 (NA)	3-5 (NA)	6-8 (NA)	9-12 (NA)	13-16 (NA)	17 or more (NA)
56	0 (0.00%)	25 (44.64%)	20 (35.71%)	10 (17.86%)	0 (0.00%)	1 (1.79%)

~ QUESTIONS AND DETAILS ~	~ ANSWER MATRICES ~

12. The subject matter was challenging.

~

Question Type: Likert

~

contributed by Dean of the School of Engineering and Applied Science

Results for CS-3240-100

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
56	3.89	0.73	9 (16.07%)	35 (62.50%)	9 (16.07%)	3 (5.36%)	0 (0.00%)	0 (0.00%)

Results for SEAS, 3000-level courses

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
1886	4.14	0.79	635 (33.67%)	938 (49.73%)	235 (12.46%)	58 (3.08%)	10 (0.53%)	10 (0.53%)

13. The objectives of the course were clearly stated and accomplished.

~

Question Type: Likert

~

contributed by Dean of the School of Engineering and Applied Science

Results for CS-3240-100

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
56	4.27	0.73	20 (35.71%)	34 (60.71%)	0 (0.00%)	1 (1.79%)	1 (1.79%)	0 (0.00%)

Results for SEAS, 3000-level courses

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
1874	4.14	0.81	644 (34.36%)	953 (50.85%)	187 (9.98%)	70 (3.74%)	16 (0.85%)	4 (0.21%)

14. There was a reasonable level of effort expected for the credit hours received.

~

Question Type: Likert

~

contributed by Dean of the School of Engineering and Applied Science

Results for CS-3240-100

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
56	4.02	0.80	13 (23.21%)	35 (62.50%)	5 (8.93%)	2 (3.57%)	1 (1.79%)	0 (0.00%)

Results for SEAS, 3000-level courses

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
1887	3.96	1.01	574 (30.42%)	941 (49.87%)	159 (8.43%)	133 (7.05%)	74 (3.92%)	6 (0.32%)

15. The homework assignments helped me learn the subject matter.

~

Question Type: Likert

~

contributed by Dean of the School of Engineering and Applied Science

Results for CS-3240-100

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
56	3.51	0.92	5 (8.93%)	23 (41.07%)	14 (25.00%)	6 (10.71%)	1 (1.79%)	7 (12.50%)

Results for SEAS, 3000-level courses

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
1879	4.15	0.88	657 (34.97%)	780 (41.51%)	173 (9.21%)	71 (3.78%)	28 (1.49%)	170 (9.05%)

16. The textbook increased my understanding of the material.

~

Question Type: Likert

~

contributed by Dean of the School of Engineering and Applied Science

Results for CS-3240-100

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
56	3.28	0.91	2 (3.57%)	15 (26.79%)	17 (30.36%)	4 (7.14%)	2 (3.57%)	16 (28.57%)

Results for SEAS, 3000-level courses

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
1881	3.83	0.99	401 (21.32%)	686 (36.47%)	291 (15.47%)	125 (6.65%)	42 (2.23%)	336 (17.86%)

~ QUESTIONS AND DETAILS ~	~ ANSWER MATRICES ~

17. The course material was well organized and developed.

~

Question Type: Likert

~

contributed by Dean of the School of Engineering and Applied Science

Results for CS-3240-100, Sherriff, Mark

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
56	4.27	0.75	22 (39.29%)	29 (51.79%)	4 (7.14%)	0 (0.00%)	1 (1.79%)	0 (0.00%)

Results for SEAS, 3000-level courses

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
2223	4.14	0.86	812 (36.53%)	1030 (46.33%)	243 (10.93%)	81 (3.64%)	32 (1.44%)	25 (1.12%)

18. The instructor was knowledgeable about the subject matter.

~

Question Type: Likert

~

contributed by Dean of the School of Engineering and Applied Science

Results for CS-3240-100, Sherriff, Mark

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
56	4.46	0.66	30 (53.57%)	23 (41.07%)	2 (3.57%)	1 (1.79%)	0 (0.00%)	0 (0.00%)

Results for SEAS, 3000-level courses

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
2224	4.53	0.61	1259 (56.61%)	838 (37.68%)	65 (2.92%)	11 (0.49%)	7 (0.31%)	44 (1.98%)

19. The instructor was well prepared for class.

~

Question Type: Likert

~

contributed by Dean of the School of Engineering and Applied Science

Results for CS-3240-100, Sherriff, Mark

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
55	4.51	0.54	29 (52.73%)	25 (45.45%)	1 (1.82%)	0 (0.00%)	0 (0.00%)	0 (0.00%)

Results for SEAS, 3000-level courses

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
2219	4.34	0.76	1007 (45.38%)	944 (42.54%)	142 (6.40%)	43 (1.94%)	18 (0.81%)	65 (2.93%)

20. The instructor (not Teaching Assistants) was accessible for individual assistance.

~

Question Type: Likert

~

contributed by Dean of the School of Engineering and Applied Science

Results for CS-3240-100, Sherriff, Mark

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
56	4.41	0.57	23 (41.07%)	26 (46.43%)	2 (3.57%)	0 (0.00%)	0 (0.00%)	5 (8.93%)

Results for SEAS, 3000-level courses

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
2219	4.07	0.87	691 (31.14%)	926 (41.73%)	313 (14.11%)	73 (3.29%)	27 (1.22%)	189 (8.52%)

21. The grading policy was fair.

~

Question Type: Likert

~

contributed by Dean of the School of Engineering and Applied Science

Results for CS-3240-100, Sherriff, Mark

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
56	4.12	1.01	25 (44.64%)	20 (35.71%)	4 (7.14%)	7 (12.50%)	0 (0.00%)	0 (0.00%)

Results for SEAS, 3000-level courses

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
2223	3.86	1.01	580 (26.09%)	973 (43.77%)	365 (16.42%)	146 (6.57%)	75 (3.37%)	84 (3.78%)

~ QUESTIONS AND DETAILS ~	~ ANSWER MATRICES ~

22. The instructor responded adequately to in-class questions.

~

Question Type: Likert

~

contributed by Dean of the School of Engineering and Applied Science

Results for CS-3240-100, Sherriff, Mark

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
56	4.58	0.50	32 (57.14%)	23 (41.07%)	0 (0.00%)	0 (0.00%)	0 (0.00%)	1 (1.79%)

Results for SEAS, 3000-level courses

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
2221	4.23	0.78	838 (37.73%)	1042 (46.92%)	182 (8.19%)	49 (2.21%)	21 (0.95%)	89 (4.01%)

23. As a teacher, this instructor was better than most others in this School.

~

Question Type: Likert

~

contributed by Dean of the School of Engineering and Applied Science

Results for CS-3240-100, Sherriff, Mark

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
56	4.29	0.78	25 (44.64%)	24 (42.86%)	5 (8.93%)	2 (3.57%)	0 (0.00%)	0 (0.00%)

Results for SEAS, 3000-level courses

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
2221	3.84	1.02	630 (28.37%)	843 (37.96%)	471 (21.21%)	164 (7.38%)	63 (2.84%)	50 (2.25%)

24. The average number of hours per week I spent outside of class preparing for this course was:

~

Question Type: Multiple Choice

~

contributed by Office of the Provost

Results for CS-3240-100

Total	Less than 1 (NA)	1 - 3 (NA)	4 - 6 (NA)	7 - 9 (NA)	10 or more (NA)
56	5 (8.93%)	18 (32.14%)	22 (39.29%)	7 (12.50%)	4 (7.14%)

Results for SEAS, 3000-level courses

Total	Less than 1 (NA)	1 - 3 (NA)	4 - 6 (NA)	7 - 9 (NA)	10 or more (NA)
1884	108 (5.73%)	559 (29.67%)	728 (38.64%)	246 (13.06%)	243 (12.90%)

25. I learned a great deal in this course.

~

Question Type: Likert

~

contributed by Office of the Provost

Results for CS-3240-100

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
55	3.98	0.80	13 (23.64%)	31 (56.36%)	9 (16.36%)	1 (1.82%)	1 (1.82%)

Results for SEAS, 3000-level courses

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
1875	4.05	0.88	604 (32.21%)	916 (48.85%)	236 (12.59%)	92 (4.91%)	27 (1.44%)

26. Overall, this was a worthwhile course.

~

Question Type: Likert

~

contributed by Office of the Provost

Results for CS-3240-100

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
55	4.35	0.75	26 (47.27%)	24 (43.64%)	3 (5.45%)	2 (3.64%)	0 (0.00%)

Results for SEAS, 3000-level courses

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
1876	4.04	0.94	643 (34.28%)	848 (45.20%)	243 (12.95%)	98 (5.22%)	44 (2.35%)

~ QUESTIONS AND DETAILS ~	~ ANSWER MATRICES ~

27. The course's goals and requirements were defined and adhered to by the instructor.

~

Question Type: Likert

~

contributed by Office of the Provost

Results for CS-3240-100, Sherriff, Mark

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
55	4.35	0.64	23 (41.82%)	29 (52.73%)	2 (3.64%)	1 (1.82%)	0 (0.00%)

Results for SEAS, 3000-level courses

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
2212	4.19	0.72	750 (33.91%)	1192 (53.89%)	215 (9.72%)	47 (2.12%)	8 (0.36%)

28. The instructor was approachable and made himself/herself available to students outside the classroom.

~

Question Type: Likert

~

contributed by Office of the Provost

Results for CS-3240-100, Sherriff, Mark

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
55	4.38	0.65	25 (45.45%)	27 (49.09%)	2 (3.64%)	1 (1.82%)	0 (0.00%)

Results for SEAS, 3000-level courses

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
2217	4.06	0.86	723 (32.61%)	1030 (46.46%)	365 (16.46%)	65 (2.93%)	34 (1.53%)

29. Overall, the instructor was an effective teacher.

~

Question Type: Likert

~

contributed by Office of the Provost

Results for CS-3240-100, Sherriff, Mark

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
56	4.46	0.60	29 (51.79%)	24 (42.86%)	3 (5.36%)	0 (0.00%)	0 (0.00%)

Results for SEAS, 3000-level courses

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
2229	4.09	0.89	785 (35.22%)	1046 (46.93%)	261 (11.71%)	95 (4.26%)	42 (1.88%)

30. Please make any overall comments or observations about this course:

~

Question Type: Short Answer

~

contributed by Office of the Provost

Results for CS-3240-100

Total	Individual Answers
31	*See below for Individual Results*

I guess I just don't understand the point of this class, it repeats so much from previous semesters. I don't know. And it just seems like they are trying to mess with us with the projects and limitations of where we can take them to work and keeping everything locked up. . .maybe if we had our own facility we would actually be able to work effectively. I am pissed off about the random hardware and software that doesn't work with my machine or the lab stuff and jumping through six hoops to get a beep to sound. I am tired of this project and the papers that are not effectively explained to us. I write papers, just what is asked for and what I get back is Not What I Was Looking For, you are picky with grading and treat different students differently. I don't get it, this class was frustrating as hell. I am so glad it is done. This started off level headed and balanced but my day just got worse, and I am tired of it all so what really pisses me off is coming out.

Project was awesome... Sherriff is awesome... Ignore the haters.

I enjoyed 3240 a lot; I think it's improved a lot since the last time it was taught.

This was a very fun course, giving us a taste of working in groups to get things done, and also as a corporation of two groups. It just shows you that sometimes you have to push the other team more to catch to yours, otherwise you both fail in the end. It's most definitely a team assignment, and without everyone playing a role, you can't succeed.

Prof. Sherriff is amazing. I'd take another class from him.

~ QUESTIONS AND DETAILS ~	~ ANSWER MATRICES ~
	Sherriff is a great professor, the project focus of this class is unique and effective. I like that we are forced to give two presentations. My one complaint is that listening to other team's presentations is completely useless and taught me very little. I guess there isn't much that can be done to change this since presenters need an audience. In the future, maybe change presentation topics to be more interesting to watch. To put how awesome Professor Sherriff is in perspective: there is approximately zero reason to come to class - no tests (besides the "midterm), no quizzes, it was all the project and coming to class didn't really effect how you did on the project too much - yet EVERYONE still came. That does not happen in college. It is not hard at all to realize that Prof. Sherriff cares about this course, cares about what he is teaching, and really cares about all of his students. This was a typical Sherriff course: fun and entertaining lecture moments, some hard days (or nights) of project work and an overall great learning experience. The projects were well thought out and conveyed the ideas of the software development accurately. Professor Sherriff consistently proves to be the best teacher I've had in all of UVA so far. Grading is fair, he answers his email quickly like a champ and the projects are grueling but awesome. CS 3240 was an outstanding class, very true to the real world of software development as I've already experienced in an internship. Death by documentation is annoying, but realistic, and the joy of seeing those darned robots finally do their thing right after an all nighter is awesome. I have plenty of reason to vent frustration here, I spent lots of late nights trying to get the stinking bluetooth to work, or trying to make that demon possessed android do my bidding, but when I step back and look at this course as a whole, it's without a doubt the kind of course I had hoped I would take when I used to think about college in my high school days. On a side note, I recently showed a prospective CS student around UVA, and when he asked about the CS department, I was able to say without a doubt that the big three of CS (Sherriff, Bloomfield, Horton) make CS a great experience, even early in the program. High honors, give Sherriff a raise, cut the STS department :) Please read my response to #10. That is my main criticism of the course. This has actually been one of my favorite classes, despite being a lot of work relative to some others. I really liked how we essentially had free reign over our learning experience. Working on the projects was a great way to simulate the real world experience of software development. You gave us an assignment, told us when it had to be completed by, and we were in charge of the rest. I loved that. I very much enjoyed doing the work for this class knowing it wasnt spoon-fed to us. I actually felt like a design champion and it made the class very satisfying. The grading in the class was questionable at best. The first assignment, you tell us to make our own assumptions about how the program should work and then have the TA take off most of the points because it didnt work under their assumptions. If you are going to give us an assignment like that, give us more of a chance to explain how it works than just two sentences in the write-up. And have the TA perhaps read the write-up to see what our assumptions were and how to operate the program. I wont go into detail about the midterm, but suffice it to say that if an answer is not what you wanted, but is correct according to what the question actually asked for, that is not grounds for taking off points. The labs were not configured for NXT development from what I could tell. Any NXT code that I loaded onto the computers would light up with red underlines and it wasnt worth dealing with. If we are going to be constrained to working in the lab, make sure that the facilities are adequate for what we need to do. Also, it would be nice if we could have worked on the robots elsewhere. I realize it is a big liability and whatnot, but honestly, when teams are leaving the cabinets unlocked and the lab open overnight, its worse. As for you personally, whoever left that comment before hit it spot on. You are awesome in class: funny, down to earth, knowledgeable, and an overall cool guy. Outside of class, though, you are a big asshole. You dont smile, you look like you are always pissed off, and your responses to questions and anonymous feedback are extremely condescending. Its like you are two completely different people in and out of class. And unless every time is a bad time, its not just bad timing and youre not just having a bad day as you suggested. On presentations, one way to increase productivity in the lab sections might be to dedicate some sections solely to presentations, and others solely to robot building. The leftover time after presentations was often not enough to be productive, but of course the presentations are an important portion of the course, too. Separating them out could mitigate the setup/cleanup time overhead which plagued the current (presentation, then a little time for building) arrangement. Of course, attendance would have to be taken to ensure that people didn't just skip the presentation sections. Great class, great professor. I learned the most about team management from this course. There's a lot to discover beyond coding and tech work. Overall, it was an effective class and the grading was very fair. Like I said above, all the components for the project should allow students to take everything out of the lab. Also, it might be interesting to add some other light homework assignments that connect to current events. For example, reading one article per week on things like Flash and the iPhone, copyright issues, or some other big topics in modern software development, and then tie them back to course material.

~ QUESTIONS AND DETAILS ~	~ ANSWER MATRICES ~
	On the days that I could actually make it to class, I felt like I was watching Jon Stewart, minus the excessively long pauses for the latter to make a stupid face. This is a good thing; it could only be made better if the minefield of idiots had allowed me to take a Westley Colbert class immediately afterward. Here is a list of lecture content items I enjoyed: tangents about being a grad student, tangents about that one guy's crazy parties and XP-ness, tangents about the reluctant object of Sherriff's sexual attention known as the Star Pony, etc. Items I didn't like: software engineering, working with people, people in general, and thinking I might be the autistic kid on the playground and the prospect of maybe deserving it too sometimes. I'll be honest, it hurts a little inside to bring the bad stuff up considering the projects were organized well and quite useful. But apologies aside, thanks to the foibles of pursuing widespread technology without standardization our team organization boils down to one person writing code, since he has the only laptop which actually connects to the NXT Robot, despite the fact that the powers-that-be who brought you the cripple-train degree requirements have introduced the additional insult to humanity's sense of reason that is 40 odd computers right there. Despite my disdain for this cavernous space in the Olsson basement, it was the source of my weekly bonus nap during the lab section when four or five woefully-lost students would stand in the front of the room and compete for the Richard Stallman award in mindless grandstanding in the face of Sherriff's occasional but insatiable grumpiness. Fuck yeah Cheerwine. Great class! I learned a lot about working and managing a larger group of people than we have done before. I really enjoyed the class - lectures and projects. Labs were slightly boring because of the presentations, but they were bearable. Great class! Selecting courses in the CS department is like walking through a minefield of geniuses. Fortunately sometimes you're called on to perform a service to your country and your sense of honor and duty (omg doodie) and whatnot tickles your conscience until you man up and take a ride on the bullet train/fulfill your degree requirements. I'll be honest, it feels good to be this nice since it feels a little bit like praising the autistic kid on the playground, since you know, even doing simple things can be difficult for them. It's easy to believe that this is somehow considered equivalent to the cs capstone given the utterly riveting lecture content, outed for the Richard Stallman award in pant-wettingly exciting material only by the student presentations, which seem to function as Sherriff's training program to indoctrinate students into the secret community of CS ingenuity which meets weekly in the incredibly full space where his brain most certainly resides under normal anatomical conditions. Though I must admit I did learn a great deal from his penetrating inquiries into why a certain woefully lost group of students organized their use case diagram around their robot and control laptop rather than threads of execution, and I have no doubt I will carry the wisdom I gained during this line of questioning well into my professional career. That being said, the projects were organized well and quite useful. Seriously, seriously, they were the greatest. Here's a list of everything we had to rely on Sherriff on for the projects: The Bluetooth dongle (haha, dongle), the "temple," and the "corridor." (lawl, I temple'd your corridor with my dongle ???). Heres a list of everything that turned to gold and the students had to love. The Bluetooth dongle (lol, dongle), the temple, the corridor, my sense of moving towards a worthwhile goal in college, etc. Out of all the valuable lessons to be learned about software development and team organization, we discover firsthand the strengths of pursuing widespread technology with standardization through TCP. Our team organization boils down to one person writing code, since she has the only laptop which actually connects to the more-than-willing object of Sherriff's sexual attention known as the NXT Robot. Fortunately, this configuration balances well since the other three members of the group are busy furiously evaluating the critical risks our corporation faces in the reliability of the Olsson basement electrical grid for our weekly RAF report. Bless Cheerwine. wait. wut? but really, there should be a third lecture hour during the week for presentations so we don't spend more than half the allocated lab time not working on the projects. I really liked the course. Not every lecture was directly applied to the project, but they were all important to hear. I am not sure what is going on in Professor Sherriff's personal life, but he needs leave those issues at home and not take out his frustration on his students. If that isn't it, perhaps he would benefit from counseling. I had him for Intro to Software Development a year ago and liked him a lot. He is not the same professor I had back then. When I went of his office to ask him questions this semester, he always seemed liked he was not in a good mood. I always felt like I was bothering him every time I asked him a question. He was not very receptive to questions. My group had some severe technical difficulties with the equipment he provided us for the project and he either didn't seem to believe us or didn't seem to care. I realize there was limited bricks, but it was not fair that my group. Our faulty njx brick cost us a LOT of time in the lab. His grading scheme was not fair either. He gave us a break down and asked us to send him any question or concerns we had but he didn't listen to what we had to say. He shot all of my groups concerns down like we had personal attacked him. He told us one thing the day of the demonstration and then graded us differently. When we tried to explain our concerns he didn't want to hear them. He gave my group a zero and the team we were working with got full credit on a component of the project that was completely independent of either group. The error that occurred was user error (a member of the other team) not a failing in my team's coding but Professor Sherriff could not bothered to hear this. He also lost his temper and was not very professional on multiple occasions. I can now tell by the wording of his emails when he has lost his temper with a student(s). As a lecturer he is good and he knows his stuff, but I do not feel comfortable going and talking to him. I work as a substitute teacher and I understand that there are times when you get frustrated and pissed off with your students, but I have never had a teacher or professor react the way he has when he gets upset with students. There are so little amount of lectures for this class. Many of classes were focused on projects. Because of this, I did not learn that much from this class that is really new and interesting. I think the professor should spend less time in talking about projects during lectures, because it wastes time. Also, I don't like the fact that professor did not have slide shows for some of the materials. Not having slide shows made some students like me harder to study for exams. Bluetooth stuff is just awful in general. Some computers may never be compatible, even with Windows 7 Professor Sherriff made it seem like the midterm was "not a big deal" and gave me the impression that I didn't need to spend a lot of time studying for it. Then I got a bad grade on it :/. I guess thats my fault tho, so whatever

~ QUESTIONS AND DETAILS ~	~ ANSWER MATRICES ~
	I really liked this course because it taught me how to work with people on complicated projects. More importantly, I learned more about how I like to do work and where I believe my ideal place would be in a team. I didn't enjoy some of the people in my major project group and this class gave me that experience (however frustrating it was). This was a fun (though complex) class nonetheless, that taught me some interesting things. Sometime the lecture are pointless. It's not like I hate it but ..... The projects will become the focuses of your life the weeks they are due. There is nothing that can be done about it. Everyone will put off the work while doing their other assignments until it must be done. Even if you are the person who has everything done ahead of time, the projects are of great enough scope that you need other people to work on it. If you are the best coder on the team, you will not enjoy the project. You will do the vast majority of the work. The papers that go along with the projects are ridiculous. They are supposed to simulate how a computer science project is managed in real life, but with the size of this project, the fact that we are students in college, and that we have other commitments in the form of classes and assignments, we just can't dedicate that kind of time to the papers and the project simultaneously. Enough about the course. Professor Sherriff is a great teacher. He knows his information and conveys it to the students well. He definitely makes this class his own. He does not take notes on grades that he records though. On multiple occasions I asked him to review a grade that I thought was incorrect and all he could tell me was "At the time, I wrote down a XX" which was worthless to me at the time and, while I ended up with the grade I wanted, it caused problems for some of my teammates. Also, rather than try to sort out the problem the moment it was mentioned, he would say "We can talk about it more if it turns out that it affects your grade" Really, that's the only time that we care about a couple of points here and there, but why defer it to a time when we're stressed and taking exams instead of when it's fresh in our minds and we can show that we completed the assignment? Good course. Takes a ton of time and energy. Expect at least one all-nighter. Papers suck. Solid course. Enjoyed the project. I was never offered cheerwine or any other illegal substance by the professor Sherriff is one of the best teachers the CS program has. I loved this class for its focus on application of skills. I really really hated how much it focused on the ability to build with legos and how dependent it was on the strength of a group. In the future you should have people rate their skills with legos / code / subversion / bluetooth / control and try to more evenly distribute abilities among the teams. While I understand that this class is about learning these skills, it really sucks when some teams are almost done by week 2 and others are struggling throughout the entire process. One of the better classes that I have taken, taught me group work skills that I had not learned yet and how important planning is in software development. Allow more time for actual completion of the project (especially major project). We were writing documentation and presenting on topics we haven't even gotten to yet because there was so little time to actually make progress on the project. Presentations were overall just not worth it, maybe 1 per semester would have sufficed. NO MORE BLUETOOTH! Loved the course, honestly feels like more of an elective than a required course though. I suppose the way I work is I get the most out of courses like this by cramming the information into my brain before a big exam. For example, in CS201 (which I took with Sherriff as well) I made heavy use of the podcasts and reviewed all (or most) of the lectures before exams to make sure I understood all of the material. There was essentially no midterm or final exam to study for, which was very very nice, and made the course fun, but I don't think I got a lot of what I was supposed to get out of the course because of this. There was no reason for me to force myself to learn the information. In fact I think I learned a lot more about software development methods out of CS201 and the elective taught by Bloomfield last semester CS4240. I love the podcast idea though, even in CS4240 where he did not record lectures, I recorded them myself with my laptop to review later. I think classes like this are a perfect format for it (unlike other subjects like math).