

CS 3240-001 Advanced Software Development - Fall 2018

ENGR (16489)

INSTRUCTORS: Sherriff, Mark (mss2x)

Respondents: 73 / Enrollment: 144

Summary: CS 3240-001 Advanced Software Development - Fall 2018 (16489)	
Overall Course Rating CS-3240-001 Mean 3.93 CS-3240-001 Std Dev 1.07 CS-3240-001 Response Count 363 SEAS, 3000-level courses Mean 4.07 SEAS, 3000-level courses Std Dev 1.03 SEAS, 3000-level courses Response Count 12492	Overall Instructor Rating INSTRUCTOR: Sherriff, Mark Mean 4.27 Std Dev 0.95 Response Count 507 SEAS, 3000-level courses Mean 4.24 SEAS, 3000-level courses Std Dev 0.95 SEAS, 3000-level courses Response Count 19768

~ QUESTIONS AND DETAILS ~ ~ ANSWER MATRICES ~

<p>1. How accurate is this statement for you: The project was of acceptable length.</p> <p style="text-align: center;">~ Question Type: Likert ~ contributed by Sherriff, Mark (mss2x)</p>	<table border="1"> <thead> <tr> <th colspan="8">Results for CS-3240-001, Sherriff, Mark</th> </tr> <tr> <th>Total</th> <th>Mean</th> <th>Std Dev</th> <th>Strongly Agree (5)</th> <th>Agree (4)</th> <th>Neutral (3)</th> <th>Disagree (2)</th> <th>Strongly Disagree (1)</th> </tr> </thead> <tbody> <tr> <td>73</td> <td>4.32</td> <td>0.88</td> <td>36 (49.32%)</td> <td>29 (39.73%)</td> <td>5 (6.85%)</td> <td>1 (1.37%)</td> <td>2 (2.74%)</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th colspan="8">Results for SEAS, 3000-level courses</th> </tr> <tr> <th>Total</th> <th>Mean</th> <th>Std Dev</th> <th>Strongly Agree (5)</th> <th>Agree (4)</th> <th>Neutral (3)</th> <th>Disagree (2)</th> <th>Strongly Disagree (1)</th> </tr> </thead> <tbody> <tr> <td>73</td> <td>4.32</td> <td>0.88</td> <td>36 (49.32%)</td> <td>29 (39.73%)</td> <td>5 (6.85%)</td> <td>1 (1.37%)</td> <td>2 (2.74%)</td> </tr> </tbody> </table>	Results for CS-3240-001, Sherriff, Mark								Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	73	4.32	0.88	36 (49.32%)	29 (39.73%)	5 (6.85%)	1 (1.37%)	2 (2.74%)	Results for SEAS, 3000-level courses								Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	73	4.32	0.88	36 (49.32%)	29 (39.73%)	5 (6.85%)	1 (1.37%)	2 (2.74%)
Results for CS-3240-001, Sherriff, Mark																																																	
Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)																																										
73	4.32	0.88	36 (49.32%)	29 (39.73%)	5 (6.85%)	1 (1.37%)	2 (2.74%)																																										
Results for SEAS, 3000-level courses																																																	
Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)																																										
73	4.32	0.88	36 (49.32%)	29 (39.73%)	5 (6.85%)	1 (1.37%)	2 (2.74%)																																										
<p>2. How accurate is this statement for you: The project was of acceptable difficulty.</p> <p style="text-align: center;">~ Question Type: Likert ~ contributed by Sherriff, Mark (mss2x)</p>	<table border="1"> <thead> <tr> <th colspan="8">Results for CS-3240-001, Sherriff, Mark</th> </tr> <tr> <th>Total</th> <th>Mean</th> <th>Std Dev</th> <th>Strongly Agree (5)</th> <th>Agree (4)</th> <th>Neutral (3)</th> <th>Disagree (2)</th> <th>Strongly Disagree (1)</th> </tr> </thead> <tbody> <tr> <td>72</td> <td>4.22</td> <td>0.94</td> <td>32 (44.44%)</td> <td>31 (43.06%)</td> <td>4 (5.56%)</td> <td>3 (4.17%)</td> <td>2 (2.78%)</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th colspan="8">Results for SEAS, 3000-level courses</th> </tr> <tr> <th>Total</th> <th>Mean</th> <th>Std Dev</th> <th>Strongly Agree (5)</th> <th>Agree (4)</th> <th>Neutral (3)</th> <th>Disagree (2)</th> <th>Strongly Disagree (1)</th> </tr> </thead> <tbody> <tr> <td>72</td> <td>4.22</td> <td>0.94</td> <td>32 (44.44%)</td> <td>31 (43.06%)</td> <td>4 (5.56%)</td> <td>3 (4.17%)</td> <td>2 (2.78%)</td> </tr> </tbody> </table>	Results for CS-3240-001, Sherriff, Mark								Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	72	4.22	0.94	32 (44.44%)	31 (43.06%)	4 (5.56%)	3 (4.17%)	2 (2.78%)	Results for SEAS, 3000-level courses								Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	72	4.22	0.94	32 (44.44%)	31 (43.06%)	4 (5.56%)	3 (4.17%)	2 (2.78%)
Results for CS-3240-001, Sherriff, Mark																																																	
Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)																																										
72	4.22	0.94	32 (44.44%)	31 (43.06%)	4 (5.56%)	3 (4.17%)	2 (2.78%)																																										
Results for SEAS, 3000-level courses																																																	
Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)																																										
72	4.22	0.94	32 (44.44%)	31 (43.06%)	4 (5.56%)	3 (4.17%)	2 (2.78%)																																										
<p>3. How accurate is this statement for you: The project helped me better understand the phases and intricacies of software development.</p> <p style="text-align: center;">~ Question Type: Likert ~ contributed by Sherriff, Mark (mss2x)</p>	<table border="1"> <thead> <tr> <th colspan="8">Results for CS-3240-001, Sherriff, Mark</th> </tr> <tr> <th>Total</th> <th>Mean</th> <th>Std Dev</th> <th>Strongly Agree (5)</th> <th>Agree (4)</th> <th>Neutral (3)</th> <th>Disagree (2)</th> <th>Strongly Disagree (1)</th> </tr> </thead> <tbody> <tr> <td>73</td> <td>4.19</td> <td>1.10</td> <td>38 (52.05%)</td> <td>21 (28.77%)</td> <td>8 (10.96%)</td> <td>2 (2.74%)</td> <td>4 (5.48%)</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th colspan="8">Results for SEAS, 3000-level courses</th> </tr> <tr> <th>Total</th> <th>Mean</th> <th>Std Dev</th> <th>Strongly Agree (5)</th> <th>Agree (4)</th> <th>Neutral (3)</th> <th>Disagree (2)</th> <th>Strongly Disagree (1)</th> </tr> </thead> <tbody> <tr> <td>73</td> <td>4.19</td> <td>1.10</td> <td>38 (52.05%)</td> <td>21 (28.77%)</td> <td>8 (10.96%)</td> <td>2 (2.74%)</td> <td>4 (5.48%)</td> </tr> </tbody> </table>	Results for CS-3240-001, Sherriff, Mark								Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	73	4.19	1.10	38 (52.05%)	21 (28.77%)	8 (10.96%)	2 (2.74%)	4 (5.48%)	Results for SEAS, 3000-level courses								Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	73	4.19	1.10	38 (52.05%)	21 (28.77%)	8 (10.96%)	2 (2.74%)	4 (5.48%)
Results for CS-3240-001, Sherriff, Mark																																																	
Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)																																										
73	4.19	1.10	38 (52.05%)	21 (28.77%)	8 (10.96%)	2 (2.74%)	4 (5.48%)																																										
Results for SEAS, 3000-level courses																																																	
Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)																																										
73	4.19	1.10	38 (52.05%)	21 (28.77%)	8 (10.96%)	2 (2.74%)	4 (5.48%)																																										
<p>4. Which topic/lecture in this course was your favorite and why?</p> <p style="text-align: center;">~ Question Type: Short Answer ~ contributed by Sherriff, Mark (mss2x)</p>	<table border="1"> <thead> <tr> <th colspan="2">Results for CS-3240-001, Sherriff, Mark</th> </tr> <tr> <th>Total</th> <th>Individual Answers</th> </tr> </thead> <tbody> <tr> <td>58</td> <td>See below for Individual Results</td> </tr> </tbody> </table>	Results for CS-3240-001, Sherriff, Mark		Total	Individual Answers	58	See below for Individual Results																																										
Results for CS-3240-001, Sherriff, Mark																																																	
Total	Individual Answers																																																
58	See below for Individual Results																																																
	<p>Really liked the software patterns lecture. It's a baseline for different patterns we can expect to see when we go into sw development!</p> <p>I thought the lectures regarding modularity in Software design were interesting. Modularity is a very useful concept that can be generally applied to many areas of SwE and life.</p>																																																

Software Architecture. I suppose I just really like elegant software architecture solutions. I feel that often the implementation of software solutions is rather straightforward, and there is more room for innovation at the architecture level

Architecture

Learning about the different SW development methodologies (Agile vs plan driven). I was really interesting to me how a company could structure the work schedule and objectives of their software team based on the nature of the company. I really hadn't thought about that before. I've interned at a few companies, but we've only ever done agile. I never knew that there was anything else, and furthermore that these methodologies really meant something about the nature of the work being done.

The one where we decided projects because we actually got to discuss as a large group the pros and cons of possible bumps in the road when it comes to choosing a project.

Security & maintenance

Software process; because of how applicable it is in terms of how teams should function depending on different factors such as size, experience, criticality

Software Architecture - most technical and applicable

Software Process -- it was interesting to learn about the intricacies of the many different ways you can approach software development

Verification and validation because I didn't realize how much of a role testing actually played

I think of all the lectures my favorite was going over programming models because that's what's most useful for internships/jobs.

MVC because it was the most interesting topic for me.

The lecture on ethics was my favorite because we talked about the ethical responsibilities of a software engineer.

Agile vs. Plan Driven

Maintenance, I enjoyed spending time on what is probably the most often overlooked portion of software engineering.

I thought architecture/design was the most interesting

Architecture. It addresses the logic model of a maintainable and scalable software.

Design patterns. You don't learn much about it in industry or in classes here at UVA.

My favorite topic was modularity as it gave me a new opportunity to think about how pieces of the software system interact and how to best implement these interactions

Software Design Patterns

I like the lab mostly because I can actually work on the project with my group and focus on the interaction between the front-end and the back-end.

I enjoyed the topic on architecture because I didn't know there were so many different, well-known, models.

Requirements and user stories, I plan on pursuing a career in product management and that material was directly relevant to my interviews.

-

Professional Issues, There are quite a few people who are after the big bucks willing to cost lives.

architecture and design. made me think about my knowledge in software development in a much broader scope and realize that some of the habits I have when coding are not ideal

The security topic. As someone like me who wishes to be part of the cybersecurity field in the future, this topic really piqued my interests. There is so much that one doesn't think about when developing an application.

Architecture and design because it was most involved with programming

Learning about MVC development was very interesting as I had only a little bit of experience with it before and it's something I'd like to understand before entering a more serious workplace where it may be a critical practice.

I enjoyed software architecture because it explained more about the structure of software and how to divide a large project into different components and understand how they work together.

~ QUESTIONS AND DETAILS ~

~ ANSWER MATRICES ~

Agile software development because I actually did it in my internship

Since I don't really have a course that stuck out to me, I'll stick with the early material on basic software processes, which was quite helpful and informative compared to most others.

Professional Development

The one one security and maintenance, I have an interest in cyber security.

Software Process because I understood it the best and it is important to have a thorough plan and process for development.

Requirements with the user stories

My favorite lecture was probably the Software Architecture - Patterns lecture

Design patterns: useful for my interviews!

Ethics, because that's a topic I care a lot about.

I liked the lecture about the decorator patterns since I've used Unity in the past. I also liked the video game references

Architecture: I find design patterns very interesting

Requirements because I understood it best, i guess

Ethics because this is something that is not directly learned, but is definitely something that I feel needs to be thought about.

frameworks, patterns and architecture

Methodologies, since I hadn't heard of the terms before

I enjoyed the talk about the code of ethics because it is important for my future

V&V, I enjoyed the way the information was given

software architecture bc it was interesting and practical. also enjoyed design patterns

Testing, mainly because I focused a lot on Quality Assurance it one of my past internships and it was interesting to see what the class teaches and what I was taught in the industry

Code Smells because they were a very accurate representation of what most programmers experience.

Requirements

Learning about MVC and REST. Because I didn't know before how state was stored and it is so widely applicable.

Architecture and Design, because it gave a ton of insight into building and designing future software projects

anything that individual work

Software Architecture was pretty cool because it is about how almost all modern web applications are made

Security; the material is interesting

software process

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-3240-001, Sherriff, Mark	
Total	Individual Answers
61	See below for Individual Results

Software design patterns

Unsure.

Scrum principle

~ QUESTIONS AND DETAILS ~

~ ANSWER MATRICES ~

Software Process

Requirements and user stories

Security; this is important in developing software systems

The entire project probably

Architecture

I can't think of a single lecture that I find useful given the fact that all of the lectures were beyond useful. I've been in internships and most of what was taught to me in this class aligns to what the world expects of you.

django, heorku and travis skills

Probably Software architecture

The architecture section. Some interview ask about this topic.

Software development process

Just the ability to use github.

The programming models would be the most useful I think.

Probably software architecture as well because it can be applied to anything. I think talking about security was also important because its something we don't necessarily acknowledge early enough but is becoming more and more important.

MVC and using Django.

AGILE

SW patterns

I think the Software Architecture topics will be the most useful

MVC/Rest/Software Design Patterns

ALL OF THEM,

MVC and RESTful development.

Requirements Engineering. When it comes to building your own solutions from scratch, this is the most important aspect

Verification/Validation

software rprocess

Methodologies for sure.

emphasis on good beginnings (req/elicitation) helps a LOT later on

-

Details about good software design like modularity, coupling, cohesion, etc. These are fundamentals that they will always notice as long as they continue in the industry.

Most if not all of them

I think learning how to elicit requirements was useful as well as incorporating design patterns into your architecture.

How to work with an agile team.

I think the software process lecture was the most useful in learning about how different environments work and their tradeoffs

Agile software development

I think I will find Software Process most useful

Architecture and Design probably

I find the lectures on maintenance to be the most useful because I now understand that software is something that people must continuously update.

Testing, mainly because it's such an important component to verifying a program and the numerous ways programs can be attacked/evaluated.

Security and Maintenance, because it will likely be used the most and be the most important to remember in the future

Ethics

The Software Process topic. That seems the most applicable across the board.

The way the agile method works and the experience of working on a software project in a group where everyone takes a different role but also works together.

Professional code of ethics was probably most helpful because these are difficult situations that you can't always ask management about.

Architecture: I find design patterns very interesting

^ same as above, either that or system architecture and design (which seems obvious).

I think security will be the most useful topic because security is always changing and that lecture topic helped bring to light the issue for people. Security needs to be considered as you are developing rather than just an afterthought.

using github!

Git/Github, a need to know industry standard.

Design patterns. It's one of the more complex topics from the courses that I wish we dived deeper into.

Learning about Github

Design patterns

Design decomposition

Requirements

Design of modules for good cohesion and coupling

software architecture bc it was interesting and practical, maybe design patterns?

Maintenance

software ethics

Software process

Agile methodologies

Maybe process or architecture -- they seem like they would have practical applications

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-3240-001, Sherriff, Mark	
Total	Individual Answers
52	See below for Individual Results

requirements as this topic can vary as it depends heavily on the client and user.

Professional Issues: One cannot teach a person to be good in a couple lectures. It takes a lot more to convince someone of ethics, but still loved the lecture!

Architecture

I did not think that the topics on some of the team stuff was relevant

Everything else

Professionalism

I can't think of a topic we talked about that was not useful. They are all important to consider for a software developer.

~ QUESTIONS AND DETAILS ~

~ ANSWER MATRICES ~

Ahhh, the SVM part? Well, it is useful, but I don't like the way it is taught.

Nil

Anything with groups

Design patterns: I'm sure it's useful but it seems like it would be something you have to recognize as you go along in your career. I guess the main point of the topic was to introduce this topic of having these commonly accepted solutions

Nothing

I fully understand the importance of ethics, so I don't necessarily want that part of the class to change, but I didn't exactly "learn anything new" in that segment (other than the fact that there's literally an SE code of ethics).

N/A

N/A

N/A

I found the software ethics topic to be the least useful only because it seemed a bit obvious that your code should above all else cause no harm. However, it was interesting that there is an established code of ethics.

-

Can't think of any!

Ethics? I think they all kinda worked but ethics left the least impact on me of all.

Professional issues - very basic and straightforward. Don't need more than a vague understanding of how to be a decent person/employee to understand this module

I dont know of any

The lecture where we learned about adaptors

Most if not all lectures seemed useful.

professionalism seemed like a waste of time

NA

The decompositions

I personally thought the MVC stuff was confusing.

the ethics one seemed kind of unnecessary and silly

Requirements likely won't stick with me

VnV

The adapter lectures.

I cannot think of an example of a topic that I found not useful as all were relevant to the course

UML diagrams or anything related to creating diagrams for software projects.

Code of Ethics lecture-- I think ethics are important but found the lecture to be along the lines of having a document which I've had to read 3 or 4 times now read to me. Good content, bad delivery.

None were non-useful, just some were super boring. SW testing, for example.

None

None

I think that the software processes topic is not as useful as other topics because that is a topic that varies very greatly from company to company so I don't think learning that is something that will help when we start working. That will always be something learned in real time because it is purely dependent on company culture.

I would have liked to spend more time on github and django itself. I really struggled with learning django. More in class time spent on it would be helpful.

~ QUESTIONS AND DETAILS ~

~ ANSWER MATRICES ~

Design patterns. I felt that it was nice to know that they existed, but I didn't feel like I cam away with any useful knowledge regarding them. I think it would have been more useful to look at more patterns, but at a higher level and then relating them to common software engineering problems.

The team based project

I don't think we delved deeply enough into security to make it worth discussing.

Requirements

I think that they were all pretty useful, but least useful was probably VnV

n/a

I felt like the VCS lecture wasn't very helpful because it wasn't applied. I wish that we had had a class where we actually learned how to use GitHub and then GitHub was related to generalized VCS. The VCS lecture was a lot of "you can use VCS to do this useful thing" but then I didn't know how to do it.

I think all the topics were worthwhile since all of them will be used in actual industry.

I think most things were necessary

I think the professional standards lecture could use more hands-on activities otherwise it may feel too obvious.

We spent a lot of time going over REST. It would've been nice to have been given a tangible way to apply that knowledge, I suppose. Not really a big deal.

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-3240-001, Sherriff, Mark								
Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
72	4.11	0.98	19 (26.39%)	16 (22.22%)	7 (9.72%)	2 (2.78%)	1 (1.39%)	27 (37.50%)

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)
72	4.11	0.98	19 (26.39%)	16 (22.22%)	7 (9.72%)	2 (2.78%)	1 (1.39%)	27 (37.50%)

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-3240-001, Sherriff, Mark								
Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
72	4.05	1.03	18 (25.00%)	15 (20.83%)	7 (9.72%)	3 (4.17%)	1 (1.39%)	28 (38.89%)

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)
72	4.05	1.03	18 (25.00%)	15 (20.83%)	7 (9.72%)	3 (4.17%)	1 (1.39%)	28 (38.89%)

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

Question Type: Multiple Choice

contributed by Sherriff, Mark (mss2x)

Results for CS-3240-001, 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)
73	3 (4.11%)	1 (1.37%)	16 (21.92%)	15 (20.55%)	11 (15.07%)	27 (36.99%)

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)
73	3 (4.11%)	1 (1.37%)	16 (21.92%)	15 (20.55%)	11 (15.07%)	27 (36.99%)

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-3240-001, Sherriff, Mark	
Total	Individual Answers
49	See below for Individual Results

N/A.

1. Instead of working through the tutorials how about implementing a mini project at the beginning of the semester. Perhaps something like creating a custom user. I think this would be better use of lab time before the start of the main project. Something like this: <https://wsvincent.com/django-allauth-tutorial-custom-user-model/> I would send out an email before the beginning of the semester Also perhaps ask students to look at the django tutorial before the beginning of class.

see #6

Better prepare people for web development. Perhaps some sort of guideline for HTML/CSS/Django coding.

no

Nothing I can think of

I was a scrum master, and on my team two team members fell behind a little, and didn't understand what was going on to the same extent as the rest of the team, which meant that it was difficult for me to assign them any work because it would be done poorly, or it would take way less time for another group member to do the same task. This compounded as they simply didn't gain any experience since they weren't assigned to more complex tasks. I am not what a good solution to this problem might include

nah, but the next question was all over the place, one week 0, another >17 hours

I think the class should be more focused around the project whatever it is. So the class will be on Django or Ruby on Rails for example, and during that you will learn about requirements and the software process and stuff.

I think there needs to be higher accountability on how much team members contribute. I really don't think that peer evals work well just because no one wants to be the person who causes you to fail. I think if peer evals were factored into something else that determined your grade it might work better (less pressure to give others good evals). For example, if instead of peer evals being directly on the spec grading chart it was part of the professional requirements or something it might work better.

I think it would be cool to have the 3 project ideas at the start of the semester and 1 extra project idea that the TAs or the Professor comes up with.

I think it's pretty unfair to give everyone the same grade on the team project. I don't know if that's how grades will be decided in the end, but based off the grading description for the course that's what it seems like it will be. I think that kind of grading is only fair when everyone in the group is equally competent and equally hard-working, which is almost never the case. Personally, I felt like I was doing lots of work but still having to rely on my teammates for a grade and at points, I felt like my work wasn't going to be worthwhile because if my teammates didn't pull through on their end, I would still get a bad grade. If this isn't how grades are decided, that should be more transparent. Also, this project was supposed to teach us how to go through the software process, but because we were told exactly which steps to complete and when, I felt like we didn't learn this lesson at well. At one point, our scrum master suggested creating a schedule for all of us to follow but we concluded it wasn't worthwhile for our team to do that, because we basically already had a schedule set up by the course. I think it would have been worthwhile to let us decide our own schedule, and force us to figure out how to plan on our own.

Make groups 1 person smaller for better distribution of work

Too hard - people who were super gods at coding took over and I couldn't help out much because I was slower at coding and not as good but that lead me to having no committees even though I wanted to help. Unfair distribution.

Nope

Have fewer people on a team

Maybe use a more structured grading system?

Maybe include video of the recording or at the least record the screen so I know which slide you're talking about without second-guessing.

smaller teams

N/A

Maybe a tutorial for those who don't know html, css, javascript, etc.

Have better TAs.

I wish the scope was more well laid out and the term "well reviewed" was more clearly defined.

We got a fair amount of autonomy for the group project. I think that is good in that we were able to figure things out on our own and develop skills and learn from and with each other. However, I think some more direction from the instructors might have been helpful especially at the beginning. I don't think throwing a 7 part tutorial at us to learn Django in one week was necessarily the best way to introduce such a project.

I really think there should be a bit more guidance for heroku if you're going to continue using that. Also I would even recommend that you make Django optional and let students choose a web framework.

Give some guidance for Django, if you're the only person on your team that knows web development you end up carrying the team

Better work distribution. 3 people on my group did most of the work, even though I wasn't one of them, I disliked it because I couldn't do the work even if I was able because of the pace others did it at and one of theirs willingness to collaborate

Please give the scrum masters more power. Three out of 6 people did basically nothing for our project, forcing me and two others to do twice as much work. This made the project incredibly difficult. There was nothing as a scrum master I could do to make them work so we were just screwed by these lazy people.

Have the start of the project itself be earlier as the jump from implementing features to near finished version was fairly quick.

1) have TAs that are more effective across the board. Some TAs appeared to be operating on different rubrics from each other, and often weren't sure themselves about deadlines and course expectations

Provide more thorough feedback throughout the semester. i feel like if we are given more feedback throughout the semester, there is no reason that a team should present a product that is anything less than well-reviewed after the beta. However, if sufficient feedback is not given before the beta, teams must scramble to meet expectations and I feel that this decreases the integrity of the project.

I would have liked to have been able to pick my group. Normally I don't care but the combination of specification grading with groupmates that did absolutely nothing made the project a lot more stressful than it probably needed to be.

Have clear guidelines for the project. TAs were essentially able to abuse the lack of clarity in the project descriptions to fail people or criticize people of opinions rather than some set fact.

None

The project was of fair difficulty, but there definitely should have been more guidance on it during the course, instead of just on Piazza and during lab. While we used the concepts that we learned during class to apply to the process of creating the project, they didn't really help with the programming aspect itself, which is what we are ultimately judged on. There should have been a lot more guidance on how to use heroku, django, postgres, etc during the lecture in order to support us in the project.

The project went pretty smoothly for my group and others that I've talked to. I would be in favor of this class doing projects the EXACT same way next year (6 people per team, have class decide projects, etc.).

Stricter sprint checks to ensure that we stayed on track. The TAs should more closely monitor the progress of the group

Stop partner evals, stop spec grading

The class should emphasize learning how to do the project a bit more. It was basically no guidance

No

No

n/a

Try to balance the project options or standardize more

Include screen recordings in the podcast

Make the teams smaller and only pick the scrum master after you met a few times. We picked the most irresponsible person to be our scrum master so our team had no structure.

I was a scrum master and used a git project board to create/assign tasks. I never heard it mentioned from TA's, but it is used in any scrum/agile team and you can set required code reviews. I recommend this for any team to equally split tasks.

~ QUESTIONS AND DETAILS ~

~ ANSWER MATRICES ~

Ahhhh maybe make the in-class activities not all crammed in the second half of the class? Also, I was lucky that I got a really supportive and nice group, but my friends' groups all seem bad.

Start sooner. We would have liked more time to come up with a functional beta (it was sort of "feature 1", "feature 2", "BETA"). I think Django is appropriate for this course.

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-3240-001, Sherriff, Mark						
Total	0-2 (NA)	3-5 (NA)	6-8 (NA)	9-12 (NA)	13-16 (NA)	17 or more (NA)
73	7 (9.59%)	40 (54.79%)	15 (20.55%)	7 (9.59%)	4 (5.48%)	0 (0.00%)

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)
73	7 (9.59%)	40 (54.79%)	15 (20.55%)	7 (9.59%)	4 (5.48%)	0 (0.00%)

12. How would you rate the availability of TAs?

Question Type: Likert

contributed by Sherriff, Mark (mss2x)

Results for CS-3240-001, Sherriff, Mark							
Total	Mean	Std Dev	Excellent (4)	Good (3)	Average (2)	Weak (1)	Very Poor (0)
73	2.73	0.90	15 (20.55%)	30 (41.10%)	21 (28.77%)	7 (9.59%)	0 (0.00%)

Results for SEAS, 3000-level courses							
Total	Mean	Std Dev	Excellent (4)	Good (3)	Average (2)	Weak (1)	Very Poor (0)
73	2.73	0.90	15 (20.55%)	30 (41.10%)	21 (28.77%)	7 (9.59%)	0 (0.00%)

13. How would you rate the helpfulness of the TAs?

Question Type: Likert

contributed by Sherriff, Mark (mss2x)

Results for CS-3240-001, Sherriff, Mark							
Total	Mean	Std Dev	Excellent (4)	Good (3)	Average (2)	Weak (1)	Very Poor (0)
71	2.54	1.07	14 (19.72%)	25 (35.21%)	19 (26.76%)	11 (15.49%)	2 (2.82%)

Results for SEAS, 3000-level courses							
Total	Mean	Std Dev	Excellent (4)	Good (3)	Average (2)	Weak (1)	Very Poor (0)
71	2.54	1.07	14 (19.72%)	25 (35.21%)	19 (26.76%)	11 (15.49%)	2 (2.82%)

14. How often did you make use of the TA office hours?

Question Type: Multiple Choice

contributed by Sherriff, Mark (mss2x)

Results for CS-3240-001, Sherriff, Mark					
Total	Every week (NA)	Every other week (NA)	Once per assignment (NA)	Rarely (NA)	Never (NA)
72	3 (4.17%)	5 (6.94%)	6 (8.33%)	36 (50.00%)	22 (30.56%)

Results for SEAS, 3000-level courses					
Total	Every week (NA)	Every other week (NA)	Once per assignment (NA)	Rarely (NA)	Never (NA)
72	3 (4.17%)	5 (6.94%)	6 (8.33%)	36 (50.00%)	22 (30.56%)

15. Any specific comments about the TAs you would like to share?

Question Type: Short Answer

contributed by Sherriff, Mark (mss2x)

Results for CS-3240-001, Sherriff, Mark	
Total	Individual Answers
39	See below for Individual Results

All the TAs were excellent. Paticularly Shabad.

Nothing other than that all the TAs were really nice and helpful.

many of them just told me to google my question even after I clearly struggled with it for a while.

The TA for our project (Matt Casias) was always very clear in his expectations and was never unreasonable in assessing us (we always got "pass", and he would explain what he liked/didn't as necessary, as well as why we passed).

no

I felt like they were too strict on grading (e.g. failing a team for a sprint because they didn't like how something looks, or a 5-minute fixable issue)

We never really got any feedback on our project during sprints until the very end. We were able to keep ourselves on track so this made the project way less stressful because we knew what needed to be done but sometimes couldn't get it done and the TA being lax gave us ease. It was however slightly frustrating/annoying to get feedback on demo day on something that we have been showing the TA almost every sprint.

I was in lab section 103. I have a lot of issues with how TAs Roman and Shabad went about TAing this course. Specifically speaking on those two TAs- 1) They had a very weak amount of knowledge when it came to basic branching structure in Git, how Django worked, and in general how the project was supposed to work. About half of the questions asked were given the response - 'we'll ask Sherriff and get back to you'. 2) We were consistently given high remarks on every sprint check then on the final two, everything went sour. For some reason Roman had decided to state that our project was lacking a lot of features and he didn't want to approve of it. This was during the beta check. Initially, we clearly stated what our features would be and he approved of them. He commended us on our progress every week, then decides on the beta day that it looked like we did the project in a week (his specific remarks) and that we don't have enough features. This is a fundamental misunderstanding of the sprint check system. The system exists to ensure that this isn't possible. To add on to the ridiculousness, my team had presented the project to Sherriff a couple of days before, and Sherriff was more than pleased with what we had. I wouldn't have minded if Roman had suggested some few things to tweak, but going back on his comments from previous weeks and throwing degrading comments towards our project was unprofessional and honestly gave the course a bad taste. 3) On the final check for lab section 103 TAs did not follow what TAs in previous sections had done (I know cause I was in the room for the previous sections). We were supposed to demo our project and then they were supposed to play around with it. Instead of that, the TAs in 103 took our Heroku link and started clicking the most random set of buttons to try and 'break it'. It was absolutely ridiculous and disorganized. The first couple of teams were graded for quite a long period of time. Due to this lack of time management, the last two teams were graded within 10-15 minutes. The reason why this is an issue is it's clearly unfair treatment. TAs spent considerably less time trying to 'break' those teams projects. To add onto this unfair treatment, my team didn't get a well reviewed due to not having a working mobile version for iPhone6 specifically. TA Shabad specifically remarked to one of the last two teams graded that although they don't have a working mobile version, he wouldn't tell TA Roman so that they would receive a full pass. Absolutely ridiculous that he would do such a thing and that he would think that students from that team wouldn't brag about that to others. Class was going so well until these last two weeks. Please choose better TAs.

They don't know enough about heroku, django, and databases to help with technical details of the project.

Hold more spread office hours

My TA was constantly moving the goal post; We had fulfilled all our requirements 3 weeks early and he kept saying "I don't think there's enough. Add this." Why did we design requirements at the beginning of the semester and receive a grade for them if they were useless in the end?

Sometimes it would be hard to find the TAs.

Mike Chang is a Django wizard. He was very courteous and super helpful in office hours.

Gave good feedback, very willing to give suggestions on how to fix problems or offer their time if we need it.

None

None

I personally liked my TA a lot, he was fair, approachable, and actually gave us good feedback every week that helped us improve the product (Sam was my TA).

I asked a TA a question once and they interrupted me and told me what to do before I had even explained my problem. I felt like that was not only rude, but ultimately unhelpful because they didn't understand my issue. As a TA, I understand that people make common mistakes and there are common solutions to those mistakes, but you should always at least listen to a question before drawing a conclusion. Also, when my group was having difficulties, we were told to simply "follow a tutorial" as though we had not already done that on our own. Overall, I think the TAs had good intentions and were nice. They gave valuable feedback, but when my group was actually struggling, they didn't seem willing to actually help. I don't expect TAs to debug for me, but I expect them to listen to my questions and actually give valuable feedback, which wasn't always done. That said, we always asked for help during lab. If that wasn't an appropriate time, we should have been told so so that we didn't expect to get help then.

They were unusually harsh for no reason.

N/A

N/A

~ QUESTIONS AND DETAILS ~

~ ANSWER MATRICES ~

No
 No
 No
 No
 n/a
 n/a
 My TA is not knowledgeable about Django, Heroku, and Travis at all. He's a nice person but lacks expertise in CS3240.
 Please show up during office hours when you are supposed to.
 -
 As mentioned previously, I would prefer that the TA's are more critical throughout the semester rather than just during the beta and the final presentation.
 The TA's were awesome and super helpful
 Nope.
 Nope.
 not particularly
 My group's TA was great.
 They were mean
 My TA was unprepared for sprint checks. He usually checks other teams first so that he can base our pass/no pass based on the other teams. He once gave us a pass and came back at the end of lab and gave us a no pass because he found something in the rubric he didn't check first. It felt unfair and rude at times.
 They should give more constructive feedback instead of sitting around for half the lab

No
 No
 No
 No
 n/a
 n/a
 My TA is not knowledgeable about Django, Heroku, and Travis at all. He's a nice person but lacks expertise in CS3240.
 Please show up during office hours when you are supposed to.
 -
 As mentioned previously, I would prefer that the TA's are more critical throughout the semester rather than just during the beta and the final presentation.
 The TA's were awesome and super helpful
 Nope.
 Nope.
 not particularly
 My group's TA was great.
 They were mean
 My TA was unprepared for sprint checks. He usually checks other teams first so that he can base our pass/no pass based on the other teams. He once gave us a pass and came back at the end of lab and gave us a no pass because he found something in the rubric he didn't check first. It felt unfair and rude at times.
 They should give more constructive feedback instead of sitting around for half the lab

16. What other topics do you wish we had time to cover or which topics did we cover that you wish we could have covered more deeply?

Question Type: Short Answer

contributed by Sherriff, Mark (mss2x)

Results for CS-3240-001, Sherriff, Mark	
Total	Individual Answers
45	See below for Individual Results

Cant think of any
 Django specific topics
 not sure
 Nothing I can think of.
 software best practices/code smells
 Testing (the actual implementation and how to do it, rather than just high level)
 Software development- practice actually writing code in class, starting with frontend (html & css), then covering backend (databases), then going over frameworks (django) then covering servers (heroku). Learn these things instead of listening to garbage lectures.
 I would have liked more discussion about hooking up apis, getting tokens, using OAuth.
 Architecture
 I wish more time was spent on professional issues, since we only spent a lecture (and a guided practice) on it, and the conversation was interesting!
 Design patterns & real UMLs!
 Django

Cant think of any
 Django specific topics
 not sure
 Nothing I can think of.
 software best practices/code smells
 Testing (the actual implementation and how to do it, rather than just high level)
 Software development- practice actually writing code in class, starting with frontend (html & css), then covering backend (databases), then going over frameworks (django) then covering servers (heroku). Learn these things instead of listening to garbage lectures.
 I would have liked more discussion about hooking up apis, getting tokens, using OAuth.
 Architecture
 I wish more time was spent on professional issues, since we only spent a lecture (and a guided practice) on it, and the conversation was interesting!
 Design patterns & real UMLs!
 Django

Ethics?

More software testing.

Django Practical implementation of SW Designs - what is not good practice?

Finding customer needs

I think it would have been useful to spend some time lecturing about specifics of the project, or different Django features. It was cool that we got to learn and discover that on our own, but it also felt disconnected where we were learning about software techniques and general things about software engineering while also having to worry about an actual project, but not having much guidance in that.

I don't know if this ties in too closely with other classes to be necessary, but CS business practice maybe? I didn't really have a problem with class structure.

It would be better if we could cover a little bit of the software structure? Like the front-end and back-end stuff? I mean I'm ok because I already have prior experience but many students come in with no experience in software development and have no idea of what is backend and frontend and how to code for that.

none

None

None

architecture patterns and good practices

Docker

IDK

I wish that we had covered how to use github because most of my team did not know and I had to teach them.

I wish we had been taught how to use Django (or whichever framework we used) and GitHub. Then, we could be taught about MVC and VCS in the context of those examples. This would be helpful so that all team members had the same competency with the technical frameworks (or at least more similar competency). It would also be helpful so that we were using the frameworks and GitHub well. Since we weren't taught these things, people with previous experience had to take on more of the burden and team members with less experience didn't learn as much. I understand that we are supposed to be able to teach ourselves, but in team projects where you rely on others, you can finish a project without learning as much as if you were doing assignments on your own. I also felt like we all figured out how to get Django and GitHub working but didn't necessarily use best practices for these tools. I would have preferred to have written better code, but it's genuinely hard to teach yourself best practices when you have short sprints and 4 other classes.

design patterns would have been fun to spend more time on, but not necessary

SW patterns. You could teach a whole course on "what are some of the most common problems you are most likely to run into as a software developer."

More on Security

Nothing off the top of my head to be honest.

I think we went into reasonable depth on each topic we did cover.

N/A

N/A

N/A

Going more in depth with ethics- how to handle negative situations with direct management

Each topic was covered adequately

n/a

n/a

unsure

Would have been nice to get a TL;DR of how websites work(databases, clients and servers, etc) before we started the project. A video tutorial is not enough; If you have questions about it you don't know if anybody found it difficult and don't want to seem stupid. If class time is spent on it then its clear that maybe some people don't know this and its okay not to.

-

~ QUESTIONS AND DETAILS ~	~ ANSWER MATRICES ~																																																						
	<p>How to do the project</p> <p>I wish we covered Git and version control a lot more.</p> <p>I would have liked to see some more design patterns in more detail.</p>																																																						
<p>17. To what degree do you agree with this statement: the team size from the project was appropriate (please elaborate in your class comments).</p> <p>~ Question Type: Likert ~ <i>contributed by Sherriff, Mark (mss2x)</i></p>	<table border="1"> <thead> <tr> <th colspan="9">Results for CS-3240-001, Sherriff, Mark</th> </tr> <tr> <th>Total</th> <th>Mean</th> <th>Std Dev</th> <th>Strongly Agree (5)</th> <th>Agree (4)</th> <th>Neutral (3)</th> <th>Disagree (2)</th> <th>Strongly Disagree (1)</th> <th></th> </tr> </thead> <tbody> <tr> <td>73</td> <td>4.18</td> <td>0.87</td> <td>30 (41.10%)</td> <td>31 (42.47%)</td> <td>7 (9.59%)</td> <td>5 (6.85%)</td> <td>0 (0.00%)</td> <td></td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th colspan="9">Results for SEAS, 3000-level courses</th> </tr> <tr> <th>Total</th> <th>Mean</th> <th>Std Dev</th> <th>Strongly Agree (5)</th> <th>Agree (4)</th> <th>Neutral (3)</th> <th>Disagree (2)</th> <th>Strongly Disagree (1)</th> <th></th> </tr> </thead> <tbody> <tr> <td>73</td> <td>4.18</td> <td>0.87</td> <td>30 (41.10%)</td> <td>31 (42.47%)</td> <td>7 (9.59%)</td> <td>5 (6.85%)</td> <td>0 (0.00%)</td> <td></td> </tr> </tbody> </table>	Results for CS-3240-001, Sherriff, Mark									Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)		73	4.18	0.87	30 (41.10%)	31 (42.47%)	7 (9.59%)	5 (6.85%)	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)		73	4.18	0.87	30 (41.10%)	31 (42.47%)	7 (9.59%)	5 (6.85%)	0 (0.00%)	
	Results for CS-3240-001, Sherriff, Mark																																																						
Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)																																																
73	4.18	0.87	30 (41.10%)	31 (42.47%)	7 (9.59%)	5 (6.85%)	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)																																																
73	4.18	0.87	30 (41.10%)	31 (42.47%)	7 (9.59%)	5 (6.85%)	0 (0.00%)																																																
<p>18. The course addressed technically rigorous subject matter consistent with the course objectives.</p> <p>~ Question Type: Likert ~ <i>contributed by Dean of the School of Engineering and Applied Science</i></p>	<table border="1"> <thead> <tr> <th colspan="9">Results for CS-3240-001</th> </tr> <tr> <th>Total</th> <th>Mean</th> <th>Std Dev</th> <th>Strongly Agree (5)</th> <th>Agree (4)</th> <th>Neutral (3)</th> <th>Disagree (2)</th> <th>Strongly Disagree (1)</th> <th>Not Applicable (NA)</th> </tr> </thead> <tbody> <tr> <td>73</td> <td>4.11</td> <td>0.94</td> <td>28 (38.36%)</td> <td>31 (42.47%)</td> <td>10 (13.70%)</td> <td>2 (2.74%)</td> <td>2 (2.74%)</td> <td>0 (0.00%)</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th colspan="9">Results for SEAS, 3000-level courses</th> </tr> <tr> <th>Total</th> <th>Mean</th> <th>Std Dev</th> <th>Strongly Agree (5)</th> <th>Agree (4)</th> <th>Neutral (3)</th> <th>Disagree (2)</th> <th>Strongly Disagree (1)</th> <th>Not Applicable (NA)</th> </tr> </thead> <tbody> <tr> <td>2498</td> <td>4.42</td> <td>0.75</td> <td>1305 (52.24%)</td> <td>998 (39.95%)</td> <td>111 (4.44%)</td> <td>39 (1.56%)</td> <td>28 (1.12%)</td> <td>17 (0.68%)</td> </tr> </tbody> </table>	Results for CS-3240-001									Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)	73	4.11	0.94	28 (38.36%)	31 (42.47%)	10 (13.70%)	2 (2.74%)	2 (2.74%)	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)	2498	4.42	0.75	1305 (52.24%)	998 (39.95%)	111 (4.44%)	39 (1.56%)	28 (1.12%)	17 (0.68%)
	Results for CS-3240-001																																																						
Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)																																															
73	4.11	0.94	28 (38.36%)	31 (42.47%)	10 (13.70%)	2 (2.74%)	2 (2.74%)	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)																																															
2498	4.42	0.75	1305 (52.24%)	998 (39.95%)	111 (4.44%)	39 (1.56%)	28 (1.12%)	17 (0.68%)																																															
<p>19. The instructor used methods other than/in addition to traditional lectures (for example, active learning, in-class problems, collaborative learning, in-class discussion) effectively in this course.</p> <p>~ Question Type: Likert ~ <i>contributed by Dean of the School of Engineering and Applied Science</i></p>	<table border="1"> <thead> <tr> <th colspan="9">Results for CS-3240-001, Sherriff, Mark</th> </tr> <tr> <th>Total</th> <th>Mean</th> <th>Std Dev</th> <th>Strongly Agree (5)</th> <th>Agree (4)</th> <th>Neutral (3)</th> <th>Disagree (2)</th> <th>Strongly Disagree (1)</th> <th>Not Applicable (NA)</th> </tr> </thead> <tbody> <tr> <td>73</td> <td>4.34</td> <td>0.79</td> <td>38 (52.05%)</td> <td>23 (31.51%)</td> <td>11 (15.07%)</td> <td>1 (1.37%)</td> <td>0 (0.00%)</td> <td>0 (0.00%)</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th colspan="9">Results for SEAS, 3000-level courses</th> </tr> <tr> <th>Total</th> <th>Mean</th> <th>Std Dev</th> <th>Strongly Agree (5)</th> <th>Agree (4)</th> <th>Neutral (3)</th> <th>Disagree (2)</th> <th>Strongly Disagree (1)</th> <th>Not Applicable (NA)</th> </tr> </thead> <tbody> <tr> <td>2830</td> <td>4.06</td> <td>1.05</td> <td>1167 (41.24%)</td> <td>968 (34.20%)</td> <td>354 (12.51%)</td> <td>200 (7.07%)</td> <td>81 (2.86%)</td> <td>60 (2.12%)</td> </tr> </tbody> </table>	Results for CS-3240-001, Sherriff, Mark									Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)	73	4.34	0.79	38 (52.05%)	23 (31.51%)	11 (15.07%)	1 (1.37%)	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)	2830	4.06	1.05	1167 (41.24%)	968 (34.20%)	354 (12.51%)	200 (7.07%)	81 (2.86%)	60 (2.12%)
	Results for CS-3240-001, Sherriff, Mark																																																						
Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)																																															
73	4.34	0.79	38 (52.05%)	23 (31.51%)	11 (15.07%)	1 (1.37%)	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)																																															
2830	4.06	1.05	1167 (41.24%)	968 (34.20%)	354 (12.51%)	200 (7.07%)	81 (2.86%)	60 (2.12%)																																															
<p>20. There was a reasonable level of effort expected for the credit hours received.</p> <p>~ Question Type: Likert ~ <i>contributed by Dean of the School of Engineering and Applied Science</i></p>	<table border="1"> <thead> <tr> <th colspan="9">Results for CS-3240-001</th> </tr> <tr> <th>Total</th> <th>Mean</th> <th>Std Dev</th> <th>Strongly Agree (5)</th> <th>Agree (4)</th> <th>Neutral (3)</th> <th>Disagree (2)</th> <th>Strongly Disagree (1)</th> <th>Not Applicable (NA)</th> </tr> </thead> <tbody> <tr> <td>73</td> <td>4.19</td> <td>0.98</td> <td>33 (45.21%)</td> <td>28 (38.36%)</td> <td>8 (10.96%)</td> <td>1 (1.37%)</td> <td>3 (4.11%)</td> <td>0 (0.00%)</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th colspan="9">Results for SEAS, 3000-level courses</th> </tr> <tr> <th>Total</th> <th>Mean</th> <th>Std Dev</th> <th>Strongly Agree (5)</th> <th>Agree (4)</th> <th>Neutral (3)</th> <th>Disagree (2)</th> <th>Strongly Disagree (1)</th> <th>Not Applicable (NA)</th> </tr> </thead> <tbody> <tr> <td>2499</td> <td>4.12</td> <td>1.01</td> <td>1025 (41.02%)</td> <td>1048 (41.94%)</td> <td>178 (7.12%)</td> <td>152 (6.08%)</td> <td>85 (3.40%)</td> <td>11 (0.44%)</td> </tr> </tbody> </table>	Results for CS-3240-001									Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)	73	4.19	0.98	33 (45.21%)	28 (38.36%)	8 (10.96%)	1 (1.37%)	3 (4.11%)	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)	2499	4.12	1.01	1025 (41.02%)	1048 (41.94%)	178 (7.12%)	152 (6.08%)	85 (3.40%)	11 (0.44%)
	Results for CS-3240-001																																																						
Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)																																															
73	4.19	0.98	33 (45.21%)	28 (38.36%)	8 (10.96%)	1 (1.37%)	3 (4.11%)	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)																																															
2499	4.12	1.01	1025 (41.02%)	1048 (41.94%)	178 (7.12%)	152 (6.08%)	85 (3.40%)	11 (0.44%)																																															

~ QUESTIONS AND DETAILS ~

~ ANSWER MATRICES ~

21. 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-3240-001								
Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
72	3.90	1.08	22 (30.56%)	19 (26.39%)	14 (19.44%)	4 (5.56%)	2 (2.78%)	11 (15.28%)

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)
2499	4.19	0.93	1038 (41.54%)	950 (38.02%)	217 (8.68%)	113 (4.52%)	47 (1.88%)	134 (5.36%)

22. 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-3240-001								
Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
72	3.21	1.02	4 (5.56%)	3 (4.17%)	11 (15.28%)	6 (8.33%)	0 (0.00%)	48 (66.67%)

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)
2497	3.59	1.22	526 (21.07%)	560 (22.43%)	439 (17.58%)	217 (8.69%)	143 (5.73%)	612 (24.51%)

23. 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-3240-001, Sherriff, Mark								
Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
73	4.19	0.95	33 (45.21%)	28 (38.36%)	6 (8.22%)	5 (6.85%)	1 (1.37%)	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)
2829	4.13	1.02	1231 (43.51%)	1052 (37.19%)	262 (9.26%)	177 (6.26%)	82 (2.90%)	25 (0.88%)

24. 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-3240-001, Sherriff, Mark								
Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
72	4.43	0.85	44 (61.11%)	18 (25.00%)	8 (11.11%)	1 (1.39%)	1 (1.39%)	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)
2826	4.55	0.72	1811 (64.08%)	783 (27.71%)	134 (4.74%)	39 (1.38%)	22 (0.78%)	37 (1.31%)

25. The instructor was well prepared for class.

Question Type: Likert

contributed by Dean of the School of Engineering and Applied Science

Results for CS-3240-001, Sherriff, Mark								
Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
73	4.36	0.93	42 (57.53%)	21 (28.77%)	5 (6.85%)	4 (5.48%)	1 (1.37%)	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)
2824	4.41	0.81	1572 (55.67%)	941 (33.32%)	179 (6.34%)	66 (2.34%)	35 (1.24%)	31 (1.10%)

~ QUESTIONS AND DETAILS ~

~ ANSWER MATRICES ~

26. I received adequate preparation from the prior courses in the curriculum to be successful in this course.

Question Type: Likert

contributed by Dean of the School of Engineering and Applied Science

Results for CS-3240-001								
Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
73	3.77	1.17	24 (32.88%)	23 (31.51%)	15 (20.55%)	7 (9.59%)	4 (5.48%)	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)
2499	3.92	1.05	782 (31.29%)	942 (37.70%)	367 (14.69%)	170 (6.80%)	88 (3.52%)	150 (6.00%)

27. The grading policy was fair.

Question Type: Likert

contributed by Dean of the School of Engineering and Applied Science

Results for CS-3240-001, Sherriff, Mark								
Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
73	3.62	1.29	22 (30.14%)	24 (32.88%)	10 (13.70%)	11 (15.07%)	6 (8.22%)	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)
2825	4.05	1.04	1113 (39.40%)	1085 (38.41%)	309 (10.94%)	188 (6.65%)	93 (3.29%)	37 (1.31%)

28. 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-3240-001, Sherriff, Mark								
Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
72	4.51	0.60	41 (56.94%)	27 (37.50%)	4 (5.56%)	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)
2819	4.34	0.85	1433 (50.83%)	1011 (35.86%)	212 (7.52%)	78 (2.77%)	41 (1.45%)	44 (1.56%)

29. The instructor effectively used technology in support of the learning goals for this course.

Question Type: Likert

contributed by Dean of the School of Engineering and Applied Science

Results for CS-3240-001, Sherriff, Mark								
Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
71	4.42	0.79	39 (54.93%)	26 (36.62%)	4 (5.63%)	1 (1.41%)	1 (1.41%)	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)
2815	4.14	0.97	1169 (41.53%)	1073 (38.12%)	292 (10.37%)	135 (4.80%)	72 (2.56%)	74 (2.63%)

30. 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-3240-001					
Total	Less than 1 (NA)	1 - 3 (NA)	4 - 6 (NA)	7 - 9 (NA)	10 or more (NA)
72	7 (9.72%)	32 (44.44%)	26 (36.11%)	4 (5.56%)	3 (4.17%)

Results for SEAS, 3000-level courses					
Total	Less than 1 (NA)	1 - 3 (NA)	4 - 6 (NA)	7 - 9 (NA)	10 or more (NA)
2500	139 (5.56%)	733 (29.32%)	970 (38.80%)	443 (17.72%)	215 (8.60%)

~ QUESTIONS AND DETAILS ~	~ ANSWER MATRICES ~																
<p>31. I learned a great deal in this course.</p> <p>~ Question Type: Likert ~ contributed by Office of the Provost</p>	Results for CS-3240-001																
	<table border="1"> <thead> <tr> <th>Total</th> <th>Mean</th> <th>Std Dev</th> <th>Strongly Agree (5)</th> <th>Agree (4)</th> <th>Neutral (3)</th> <th>Disagree (2)</th> <th>Strongly Disagree (1)</th> </tr> </thead> <tbody> <tr> <td>73</td> <td>3.88</td> <td>1.01</td> <td>21 (28.77%)</td> <td>32 (43.84%)</td> <td>12 (16.44%)</td> <td>6 (8.22%)</td> <td>2 (2.74%)</td> </tr> </tbody> </table>	Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	73	3.88	1.01	21 (28.77%)	32 (43.84%)	12 (16.44%)	6 (8.22%)	2 (2.74%)
	Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)									
73	3.88	1.01	21 (28.77%)	32 (43.84%)	12 (16.44%)	6 (8.22%)	2 (2.74%)										
Results for SEAS, 3000-level courses																	
	<table border="1"> <thead> <tr> <th>Total</th> <th>Mean</th> <th>Std Dev</th> <th>Strongly Agree (5)</th> <th>Agree (4)</th> <th>Neutral (3)</th> <th>Disagree (2)</th> <th>Strongly Disagree (1)</th> </tr> </thead> <tbody> <tr> <td>2496</td> <td>4.13</td> <td>0.94</td> <td>1017 (40.75%)</td> <td>1013 (40.58%)</td> <td>292 (11.70%)</td> <td>125 (5.01%)</td> <td>49 (1.96%)</td> </tr> </tbody> </table>	Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	2496	4.13	0.94	1017 (40.75%)	1013 (40.58%)	292 (11.70%)	125 (5.01%)	49 (1.96%)
Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)										
2496	4.13	0.94	1017 (40.75%)	1013 (40.58%)	292 (11.70%)	125 (5.01%)	49 (1.96%)										
<p>32. Overall, this was a worthwhile course.</p> <p>~ Question Type: Likert ~ contributed by Office of the Provost</p>	Results for CS-3240-001																
	<table border="1"> <thead> <tr> <th>Total</th> <th>Mean</th> <th>Std Dev</th> <th>Strongly Agree (5)</th> <th>Agree (4)</th> <th>Neutral (3)</th> <th>Disagree (2)</th> <th>Strongly Disagree (1)</th> </tr> </thead> <tbody> <tr> <td>72</td> <td>3.92</td> <td>0.99</td> <td>23 (31.94%)</td> <td>28 (38.89%)</td> <td>14 (19.44%)</td> <td>6 (8.33%)</td> <td>1 (1.39%)</td> </tr> </tbody> </table>	Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	72	3.92	0.99	23 (31.94%)	28 (38.89%)	14 (19.44%)	6 (8.33%)	1 (1.39%)
	Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)									
72	3.92	0.99	23 (31.94%)	28 (38.89%)	14 (19.44%)	6 (8.33%)	1 (1.39%)										
Results for SEAS, 3000-level courses																	
	<table border="1"> <thead> <tr> <th>Total</th> <th>Mean</th> <th>Std Dev</th> <th>Strongly Agree (5)</th> <th>Agree (4)</th> <th>Neutral (3)</th> <th>Disagree (2)</th> <th>Strongly Disagree (1)</th> </tr> </thead> <tbody> <tr> <td>2494</td> <td>3.99</td> <td>1.07</td> <td>954 (38.25%)</td> <td>932 (37.37%)</td> <td>352 (14.11%)</td> <td>153 (6.13%)</td> <td>103 (4.13%)</td> </tr> </tbody> </table>	Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	2494	3.99	1.07	954 (38.25%)	932 (37.37%)	352 (14.11%)	153 (6.13%)	103 (4.13%)
Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)										
2494	3.99	1.07	954 (38.25%)	932 (37.37%)	352 (14.11%)	153 (6.13%)	103 (4.13%)										
<p>33. The course's goals and requirements were defined and adhered to by the instructor.</p> <p>~ Question Type: Likert ~ contributed by Office of the Provost</p>	Results for CS-3240-001, Sherriff, Mark																
	<table border="1"> <thead> <tr> <th>Total</th> <th>Mean</th> <th>Std Dev</th> <th>Strongly Agree (5)</th> <th>Agree (4)</th> <th>Neutral (3)</th> <th>Disagree (2)</th> <th>Strongly Disagree (1)</th> </tr> </thead> <tbody> <tr> <td>72</td> <td>4.39</td> <td>0.72</td> <td>35 (48.61%)</td> <td>32 (44.44%)</td> <td>4 (5.56%)</td> <td>0 (0.00%)</td> <td>1 (1.39%)</td> </tr> </tbody> </table>	Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	72	4.39	0.72	35 (48.61%)	32 (44.44%)	4 (5.56%)	0 (0.00%)	1 (1.39%)
	Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)									
72	4.39	0.72	35 (48.61%)	32 (44.44%)	4 (5.56%)	0 (0.00%)	1 (1.39%)										
Results for SEAS, 3000-level courses																	
	<table border="1"> <thead> <tr> <th>Total</th> <th>Mean</th> <th>Std Dev</th> <th>Strongly Agree (5)</th> <th>Agree (4)</th> <th>Neutral (3)</th> <th>Disagree (2)</th> <th>Strongly Disagree (1)</th> </tr> </thead> <tbody> <tr> <td>2815</td> <td>4.35</td> <td>0.76</td> <td>1341 (47.64%)</td> <td>1218 (43.27%)</td> <td>169 (6.00%)</td> <td>62 (2.20%)</td> <td>25 (0.89%)</td> </tr> </tbody> </table>	Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	2815	4.35	0.76	1341 (47.64%)	1218 (43.27%)	169 (6.00%)	62 (2.20%)	25 (0.89%)
Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)										
2815	4.35	0.76	1341 (47.64%)	1218 (43.27%)	169 (6.00%)	62 (2.20%)	25 (0.89%)										
<p>34. The instructor was approachable and made himself/herself available to students outside the classroom.</p> <p>~ Question Type: Likert ~ contributed by Office of the Provost</p>	Results for CS-3240-001, Sherriff, Mark																
	<table border="1"> <thead> <tr> <th>Total</th> <th>Mean</th> <th>Std Dev</th> <th>Strongly Agree (5)</th> <th>Agree (4)</th> <th>Neutral (3)</th> <th>Disagree (2)</th> <th>Strongly Disagree (1)</th> </tr> </thead> <tbody> <tr> <td>71</td> <td>4.20</td> <td>0.87</td> <td>32 (45.07%)</td> <td>24 (33.80%)</td> <td>12 (16.90%)</td> <td>3 (4.23%)</td> <td>0 (0.00%)</td> </tr> </tbody> </table>	Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	71	4.20	0.87	32 (45.07%)	24 (33.80%)	12 (16.90%)	3 (4.23%)	0 (0.00%)
	Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)									
71	4.20	0.87	32 (45.07%)	24 (33.80%)	12 (16.90%)	3 (4.23%)	0 (0.00%)										
Results for SEAS, 3000-level courses																	
	<table border="1"> <thead> <tr> <th>Total</th> <th>Mean</th> <th>Std Dev</th> <th>Strongly Agree (5)</th> <th>Agree (4)</th> <th>Neutral (3)</th> <th>Disagree (2)</th> <th>Strongly Disagree (1)</th> </tr> </thead> <tbody> <tr> <td>2827</td> <td>4.32</td> <td>0.87</td> <td>1467 (51.89%)</td> <td>963 (34.06%)</td> <td>267 (9.44%)</td> <td>92 (3.25%)</td> <td>38 (1.34%)</td> </tr> </tbody> </table>	Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	2827	4.32	0.87	1467 (51.89%)	963 (34.06%)	267 (9.44%)	92 (3.25%)	38 (1.34%)
Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)										
2827	4.32	0.87	1467 (51.89%)	963 (34.06%)	267 (9.44%)	92 (3.25%)	38 (1.34%)										
<p>35. Overall, the instructor was an effective teacher.</p> <p>~ Question Type: Likert ~ contributed by Office of the Provost</p>	Results for CS-3240-001, Sherriff, Mark																
	<table border="1"> <thead> <tr> <th>Total</th> <th>Mean</th> <th>Std Dev</th> <th>Strongly Agree (5)</th> <th>Agree (4)</th> <th>Neutral (3)</th> <th>Disagree (2)</th> <th>Strongly Disagree (1)</th> </tr> </thead> <tbody> <tr> <td>72</td> <td>4.31</td> <td>0.83</td> <td>35 (48.61%)</td> <td>27 (37.50%)</td> <td>8 (11.11%)</td> <td>1 (1.39%)</td> <td>1 (1.39%)</td> </tr> </tbody> </table>	Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	72	4.31	0.83	35 (48.61%)	27 (37.50%)	8 (11.11%)	1 (1.39%)	1 (1.39%)
	Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)									
72	4.31	0.83	35 (48.61%)	27 (37.50%)	8 (11.11%)	1 (1.39%)	1 (1.39%)										
Results for SEAS, 3000-level courses																	
	<table border="1"> <thead> <tr> <th>Total</th> <th>Mean</th> <th>Std Dev</th> <th>Strongly Agree (5)</th> <th>Agree (4)</th> <th>Neutral (3)</th> <th>Disagree (2)</th> <th>Strongly Disagree (1)</th> </tr> </thead> <tbody> <tr> <td>2829</td> <td>4.20</td> <td>0.97</td> <td>1306 (46.16%)</td> <td>1047 (37.01%)</td> <td>268 (9.47%)</td> <td>138 (4.88%)</td> <td>70 (2.47%)</td> </tr> </tbody> </table>	Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	2829	4.20	0.97	1306 (46.16%)	1047 (37.01%)	268 (9.47%)	138 (4.88%)	70 (2.47%)
Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)										
2829	4.20	0.97	1306 (46.16%)	1047 (37.01%)	268 (9.47%)	138 (4.88%)	70 (2.47%)										

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

~
Question Type: Short Answer

~
contributed by Office of the Provost

Results for CS-3240-001	
Total	Individual Answers
44	See below for Individual Results

I like the project portion of the course. The lectures are useful for concept learning.

I think the group size was good. It was pretty easy to divide weekly tasks into even amounts for every team member.

N/A.

While the course material is extremely practical and useful, I feel like it could all be condensed down into a month or two. Take out some of the fluff and speed up slightly and it is half a semester of material. That's why I think the course should be focused more around the project and learning the strategies of software engineering as you go. Besides, there are only 2 courses for people to learn web development (excluding this course) and 1 is in Django and the other is in PHP.

I like the spec grading

Push grades out quicker.

This was a terribly-structured course, that no one learns any actual material in.

Professor Sherriff is one of the best professors I have ever had. I wish we had better TAs but the TA selection process doesn't make sense to me to begin with

* I dedicated so much time to the project because I got passionate about it and enjoyed working on it
* Regarding team size: I really feel like the projects worked on in this class could be published and made into something big. The textbook exchange, assignment calendar or resource finder apps could all be extremely useful utilities for students. If the team sizes were larger, then each team would be able to do so much more with their projects in the given time frame, and would be better able to develop the app into something that UVA students could actually be able to use. It might also be more representative of a real world software development project if it were done this way... but the potential for students in this class to develop legitimately useful apps for UVA should not be neglected. *
Otherwise, I did learn a lot from this project. Being dropped into it with no knowledge of Django, Python, HTML, CSS, or Javascript, and having to pick it all up on my own with minimal guidance, actually proved to be a really worthwhile learning experience. I also learned a lot about the software development process, and overall I'd rank the lab/project for this course as one of the most worthwhile courses I'll take here at UVA.

Good course overall. I think there should be more of an emphasis on good project management techniques (for example, branches, code review, testing, etc). I think the grading policy was very fair. If you're modifying it, please do keep the two tries per exam!

Sherriff was great

I dislike random assignment of groups, but I understand its purpose.

Sherriff does a good job making the lecture engaging

The class was informative and fun, but suffered from a poor grading scheme. The Specification Grading is a decent idea, but leave too much information up in the air for a student to evaluate their current standing in the class (until the very end). Some metrics of grading felt arbitrary, and so felt unfair/unclear what was expected

Spec grading is honestly the worst thing. If you miss one question on one KA you drop down a letter grade, then you have no motivation to get well-reviewed on the project so there's no point in doing well there. I don't know why spec grading was used in the first place, as I don't think it accurately shows how much a student knows about software development. HAVE A DJANGO CRASH COURSE OR SOMETHING. Two people in my group knew web development, me and the scrum master, and the scrum master didn't seem to be motivated so I ended up completely carrying my team because none of my other teammates knew web dev. It is a bit unfair to just throw us into the deep end with this because all the work falls on one person.

I am not a fan of the specification grading as I feel that components of a class that usually may not be as highly weighted could potentially be enough to sink your grade even if you are near perfect in other areas that may better reflect your mastery of the course subject matter. Other than my issues with the grading system, I found this to be a great class and I learned a lot.

none

Group size could be one less for more work distribution (not necessary, but will make work even more distributed). Also, place more emphasis on front-end design since project is graded strictly on both front-end and back-end.

None

Sherriff. You're probably the most entertaining and witty lecturer I've had. And, there's no doubt that you know your stuff. However, I really don't like the setup of the KA part of the specification grading scheme. I think that people end up putting off the KA assessments until the end of the semester because there's no requirement to take them around the same time that you learn the corresponding material. I think there needs to be some requirement in place to ensure that people take them early. In addition, the all or nothing grading requirement for KAs is very stressful, especially when you need perfect scores on all of the KA assessments in order to earn an A in the class. Instead, I'd propose that you aggregate KA scores into a single score. So for example, if there are 6 KAs and they're worth 5 points each, then the total amount of KA points a student could get would be 30. Then, their score out of 30 becomes their overall "KA" grade, and the specification grading chart could have a column that maps different point ranges to grades. This way, a student could get a 4/5 on one or two KA assessments and still earn an A or A- in the class.

The specification grading guidelines were meant to be a way to reduce stress on students while still ensuring that students know the course content. Although the intentions were good, they really increased the stress of most students. Instead of picking and choosing what I should focus my efforts on, I had to focus on everything, and make sure that everything was up to par. The fact that I can't get an A in the class if I miss 1 question on 1 KA is unfair. Also, even though Sherriff said that attendance would not be required, it definitely was during guided practices. He should make the expectations more clear.

Mark Sherriff is by far one of my most favorite professor here at UVA. He makes boring topics into fun and engaging lectures. I don't think I would have learned as much if I had any other professors.

I think that the spec grading could be improved. I am not sure what the best way to do this is, but I feel that missing one question on two assignments should not drop a student by two letter grades. This could potentially be improved by lowering the number of questions required per assignment for a pass, or perhaps by requiring a certain number of questions that must be answered accurately in the KA's. For example, if there are 6 KA's with 5 questions each, then say that a student has to answer a total of like 26 KA questions correctly regardless of which KA's those 26 come from.

I spent this entire course wondering where the magical course we were supposed to take to understand how a website works was. Have gone through an entire project in django and still am not entirely sure how anything works.

As I already said in my survey on Spec. Grading, it's not a system that fits this course very well. I believe Sherriff already made it clear that it won't be used in the future, so that pretty much takes care of my only glaring issue. I would definitely recommend Sherriff as a professor.

Teams were kind of big. 5 Seems likes a good number. No more than 7. Also felt that some projects were more straightforward to implement than others. The study on Grounds was the hardest one or the Assignment. Textbook seemed the easiest one. Maybe pick ideas from this semester and change them up, so they are equally as hard?

huh group size was fine, but my only issue was that i was left in the dust by a group of people who knew the content way more than me. I had never used django, python or any kind of web development before in my life

Mark Sherriff was a fantastic, dedicated instructor and I would love to take more courses taught by him.

This might have been the first time working in such a large group. It was overall fine. My main comment is that there were a lot of members (7) but not enough stuff that needed to be done or everything that needed to be done was very related (so whoever did one thing basically had to do the other stuff). What ended up happening was we would have groups of 2 or 3 who would consistently work together and 1 or 2 people who worked alone. This usually left 2 people (usually the same people every sprint) doing nothing or doing very minimal things. I don't know how you would solve this as making the project bigger would probably just mean that the people doing the work would have to do more and the people not doing any work would just chill. Other than that, the project was fun to work on: got to meet and make friends with new people, got to learn new things that I probably wouldn't have sought out to learn on my own, and got to work/problem solve with very nice people. I found specification grading to be a double-edged sword, especially for the two attempt policy. It's good because on the first attempt there is no pressure, but then on the second attempt there is a lot of pressure especially since you already didn't pass once. Sherriff was, however, very reasonable about it (giving similar exams twice, regrading things, making sure people lost points only if they deserved it). If it was someone else who was a hard**s, I don't think this experiment would have worked out as well. Final Comment: I came into this class with the expected attitude of expecting to get nothing out of the class and thinking this class was useless. However, Sherriff did a really good job of showing us why these topics are important to learn. I think the big group sizes really accomplished their purpose of teaching us how to work in teams and how to deal with "office problems". Overall, I think the class will be very helpful in my career (learning github, learning how to deal with people over a long period of time in a sustainable way, etc...). Prof. Sherriff had to work really hard to make this class work. Although there were some issues (like the spec grading stuff), I think overall it was alright/good and Sherriff's attitude towards teaching made it work. Thanks Sherriff.

Great class to learn a software. Teamwork on the project is not as organized as expected, but we have a good outcome.

n/a

Sorry to future students for saying this, but make this class harder. I feel like now I know what design patterns are, but I didn't learn how to implement any in this course. That's kind of my take-away from a lot of the KA areas. Go into more detail on topics and force students to (or at least reward them for) learn how to *apply* those topics. Things I liked in this course: KA assessments. It was easy to study for these because I knew what to expect, and it also was clear to see what I had learned/was being assessed on. Course was really well organized which was good.

Should not be a requirement.

I enjoyed learning about software development but I'm not sure how I felt about the project. On one hand, it was nice to experience the software process as we learned in class, but on the other it relied on us learning the actual process of making a website on our own.

Sherriff is a phenomenal professor. One of the best I have ever had at UVA. That being said, Specification Grading is terrible. Taking the lowest score seems strange since a perfect student who fails one KA is pushed to a B+. The incentive structure seems wrong. People stop trying or get really nervous that their grade is not secure.

Professor Sherriff was truly passionate about the course material and he did his level best to ensure that everyone learned and our grades in the class reflected our level of understanding of the material. He gave us ample chances to improve our grades, and was overall a very fair professor.

I think the spec grading is a good idea. I've heard some of my classmates complain about it, but I think its very fair.

I believe the group size was appropriate as we were able to finish the project fairly easily without wasting any time. There were only five people in my group but all of them contributed significantly (which unfortunately doesn't seem to be the case for some other groups).

I liked the project and KA aspect of the course. However, for the project, I had limited knowledge of programming languages outside of what we have learned thus far (C++, Java, Python) so it was frustrating when my teammates knew CSS, HTML, and Javascript and I felt behind.

I think the course should be more focused on the project. I.e. we shouldn't have to write essays.

I think this course has a lot of thought behind it and I appreciate the transparency from the instructor for explaining why we do things a certain way. I would say the specification grading was fair because we knew what was expected of us from the beginning, and I appreciate that upfront communication. There were very little surprises in this course. That said, the only thing I am disappointed about spec grading is that if one part is off, while the others are perfect, the grade drops (unless i misunderstand spec grading). Overall, I learned a lot through the project and through the diff software concepts. And I appreciated the attitude of the instructor.

Keep specification grading! Perhaps implement it with more courses in CS. I found this type of grading extremely useful Perhaps teach more CS3240 concepts in CS2110

The team size doesn't matter much as only some of the students will actually work. I essentially had a team of me and two others as three people ghosted the group and did negligible work. It would have been incredible to have 6 people collaborate on the project but due to the way the class is run there is very little accountability and laziness is encouraged.

Not the biggest fan of specification grading, but I appreciate Professor Sherriff's hard work in trying it out!