C S 201-0001 **Software Development Methods** - Fall 2008

School Of Engineering And Applied Science (301rn)

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

Respondents: 65 / Enrollment: 88

Summary: C S 201-0001 Software Development Methods - Fall 2008 (301rn)

Overall Course Rating

C S-201-0001 Mean 3.78
C S-201-0001 Std Dev 1.03
C S-201-0001 Response Count 323

Overall Instructor Rating

INSTRUCTOR: Sherriff, Mark
Mean 4.48
Std Dev 0.67
Response Count 450

~ QUESTIONS AND DETAILS ~

~ ANSWER MATRICES ~

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

~

Question Type: Short Answer

~

contributed by Sherriff, Mark (mss2x)

Results for C S-201-0001, Sherriff, Mark

Total	Individual Answers
61	*See below for Individual Results*

SQL Injection, Complexity

The unit on Swing is my favorite because it starts to get into things that are more related to practical software that is designed for the end user, as opposed to the more basic background functions.

I enjoyed the topic on internet security because it is a very relevant topic for everyday users. The section on sql injection and hacking was especially entertaining.

No opinion.

The process of creating a program and applying it to a project

Operating Systems, because that's what first got me interested in computer science.

I liked the project because it was fun to work with people and create such a large scale project.

linux, sweet

I liked the hashing lecture because it was interactive.

project. working with teams

Threads - I feel as though this is a very important subject when it comes to more complex programs and the explanation relating to operating systems was very interesting.

Binary search trees were my favorite topic, because I nailed the question on binary search trees on the test.

Internet security because we got to hack a server.

Inheritance

I enjoyed the lecture on internet security.

I thought pair programming was a good idea on paper but in actuality it slowed down the programming process.

My favourite was the stuff we did on GUIs, although I'd have liked to do a bit more practice on it before we started the project.

I enjoyed the lecture on SQL Injection attacks because it was very interesting. I felt like if we knew a bit more it could be something I would enjoy messing around with very much because it is like a puzzle.

The collections and iterators were my favorite part because they were the most helpful topic that we covered as far as being able to write code more effectively in Java.

arraylist, maps, GUI

SQL Injection hacking stuff

Operating systems... it was a refreshing break from the heavier programming stuff that we had done most of the semester.

Networks and internet security. Of everything we covered, I see this as being the most applicable. (The OOP and stuff was handy, too, but it's the kind of thing I already picked up just playing around with Java after taking 101.)

~ QUESTIONS AND DETAILS ~	~ ANSWER MATRICES ~
	Special topics. They were fun, short and shallow in depth, and well suited to ADD.
	internet security.
	SQL Injection and Hacking. It was the most fun.
	Security and networking, because they seemed most useful for my future.
	the extra topics at the end of the year the random internet security and operating systems
	SQL Injection
	I don't really have a favorite.
	No particular topic really stood out to me that much
	Threads, I thought it was the most interesting
	The last few lectures on internet security. The SQL injection attacks we did in class was some of the most fun I've had during a class all semester.
	interesting things like SQL injection, GUI stuff, RSS feeds and JavaDoc - These things are really applicable and more than just book examples.
	I feel like the answers I put for question 2 (subclasses and superclass) is kind of is my favorite, because I like learning useful tools, but if I were to pick something intrinsically interesting in of itself, I would actually say hash code. When I learned about information theory in ECE 200, I got the impression that efficiency in sorting required as much distinguishing information as possible, so it was interesting to see a that hash code basically works in the opposite way.
	The Security lecture was informative and fun at the same time. It was interesting to see the hack attempt and success of the blurb sent out before the class where we tried to hack into Krondor.
	Operation Systems Threads Internet Security Because they were all new topics that I had never covered before in my other CS classes.
	the j-unit testing. it was a good supplement to learn how to test our own code
	inheritance, I don't fully understand it yet but it seems useful
	Just object orientation in generalq
	I enjoyed the last few weeks of class, where we covered topics voted on in class.
	I really enjoyed using and designing swing applications. While I have not had any experience with interfaces outside of swing (which I hear is a pain compared to other toolkits/languages) it seems to me that this is still a very useful skill and there is a certain sense of accomplishment that goes with being able to see a physical product of your code.
	Recursive algorithms were my favorite because I found them very applicable to real life problems and just the general idea of recursion is interesting to me.
	Swing, because having a user interface is awesome.
	the im client
	Almost all topics were interesting. I enjoyed OS/Linux more because I already knew about it and could easily connect with it.
	The brief introductions to the topics after Exam 2 were all interesting since they were selected by the class itself.
	data structures
	Internet security. It's something I can play around with while bored and it should really increase my knowledge of basic programming.
	internet security fun and interesting
	Internet security, while not the most useful for my future academic career in CS, was the most interesting single topic, especially because of the hands-on learning in class.
	sql injection...i thought it was fun to see basics of how hackers hack
	Java framework/collections because it was the most practical.
	Guis and event driven programming. These topics allowed me to produce programs that I've been meaning to write but haven't had the instruction for.

~ QUESTIONS AND DETAILS ~	~ ANSWER MATRICES ~
	Not sure, but probably testing because it is very important and we learned JUnit and different types of test cases, which will help in the future. Operating systems.. it was enlightening to know how they worked, and fun hating on microsoft internet security, threads (but I think it needs to have some coding assignment/practice to help learn it), and operating systems, networking SQL Injection attacks - it was exciting and useful for those who might move on to work on secure data bases. Recursion, I find it quite interesting topics describing computer engineering because they showed another, not seen as much side of computer science. Not just coding but how to develop software and the processes used. Design Approaches, Can't think of other things
2. 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 C S-201-0001, Sherriff, Mark Total: 61 Individual Answers: *See below for Individual Results*
	The most useful topic for the future will probably be inheritance because I see myself having to do a lot of superclass and interface work in the future. Phases of development No opinion. I liked learning about subclass and superclasses, interfaces and abstract classes, and polymorphism the best. I felt like learning how to implement those things were of great use to my tools as a programmer, and really helpful for both negotiating with library with already written class, and creating new classes for new frameworks. In both of those ways I got to use these tools and understand how they were helpful and easy. Also, I know that ArrayLists and iterators will make my work a lot easier in the future. Software Development, I feel like I can apply many aspects of it to my systems major The process of creating a program and applying it to a project Understanding the software development cycle and the components of each step. Iteration or something? Not sure I'm not really sure, but I feel like the project encompassed a lot of broader software development concepts The lecture on operating systems could be of use. I've never heard much about Linex until that class and I learned a lot about the type of build there are for Linex, Windows, and Mac. Extreme programming. This will help me work better at a group member. swing data structure The introduction and use of subversion was the thing I found that will be the most useful in the future. Inheritance, casting, and other Java-related topics are going to be extremely useful. Inheritance, and ADTs Polymorphisim, Gui coding and perhaps threading. Again, all topics were useful and relevant to my major I can't think of anything in particular that stands out; all the material I learned, I think, will be useful. Specification, usually the hardest part of project. Everything else is useful as well. Software testing, because it will help out a lot when programs are not necessarily working the way they may need to.

~ QUESTIONS AND DETAILS ~	~ ANSWER MATRICES ~
	I found the topics dealing with internal computer processes and the idea of computer programming being a social event as very important. It helped a lot of people work better.
	Probably the internet security topic.
	Software Development Life Cycle & Testing
	I think that discussing teamwork and the stages of software development will be most important. Anyone can code, but planning out the work in advance and being able to work well in teams is really what helps get things done.
	Networking and internet security. That's more or less the field I want to pursue.
	software testing/ inheritance
	The different ways of developing software. (Phases of development)
	Internet security..as mentioned before it's relevant topic for everyday users and whether I decide to pursue a career in CS or not, this topic will provide insight as to how to better protect my computer.
	I think that input/output streams will probably be the most useful thing I learned. Often times when I'm programming, reading or printing to a file is important.
	The lectures on the non-technical aspects software engineering were probably the most useful. Instruction on phases of development, software design, and agile development were among the topics that would seem very useful for developing larger programs in group environments.
	Everything about inheritance and different types of objects, because I think that is the key to starting to program actual software.
	inheritance and the general knowledge of programming that i can carry on into 216
	The software development method will be useful in the future because it is applicable to other areas besides computer science and is a good problem solving method.
	See #1. (That is, "Networks and internet security. Of everything we covered, I see this as being the most applicable. (The OOP and stuff was handy, too, but it's the kind of thing I already picked up just playing around with Java after taking 101.)")
	not sure
	not sure
	See #1.
	ADTs
	Pair programming will probably be helpful.
	the j-unit testing
	software development: agile & plan driven development
	I liked the topics on testing your code. It was something i never really thought about before this course, and didnt think about writing code to test code. As I build programs outside of class, i have been testing them, which has proved useful and detected at lease a couple bugs so far.
	inheritance
	actually i think all of them will be very useful
	Compairtors
	Im not really sure. Maybe the one on the software life cycle? or on the important of specifications and requirements
	Threads, Object oriented programming, recursive programming, uml class diagrams, etc (too many to name)
	internet security, frameworks, adts, iterators
	the software development process and testing techniques
	Trees & Recursion
	Probably learning about the design process.
	Probably the stuff on GUIs and scanners and the For-each loop.
	Appropriate uses for data structures in complex applications.

~ QUESTIONS AND DETAILS ~	~ ANSWER MATRICES ~
	Internet secuity and 5 levels of development The general structure of the SW development process. security and networking Event Driven Programming - almost all consumer applications have a gui. Probably recursive functions and data structures. A well written recursive function can simplify parts of a program by a lot, and there is a certain trick to understanding them that I feel I now understand. SQL injection could be useful to me as I do have a strong interest in developing web-based applications that I will have to protect for misuse. not very much
3. 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 C S-201-0001, Sherriff, Mark Total: 53 Individual Answers: *See below for Individual Results*
	Nothing in particular Cannot think of any No opinion. They all seemed like important topics. recursive algorithms Not enough time/lab spent on Swing (since I wasn't the GUI person in the project, I didn't really get any hands-on experience with GUIs). Threads and networking seemed to be covered in too cursory of a fashion to be practically useful in the future (though the general concepts are helpful to know). Linux uml diagrams were confusing - i know they'll be useful though. The J unit testing we put in our codes was really pretty worthless. The code either worked or didn't work. This type of test at this level didn't show much about the functionality or correctness of a program. If I had to say one that "didn't work," I would say threads. We studied them, but I didn't feel real confident about how to implement them, and though we talked about some instances in which they are used, really would have liked to have tried something with threads, so I understand their function on a basic level in terms ways to solve cs problems. pair programming.....test document... Threading, while an important concept for students to be aware of, probably wasn't worth an entire class period. Testing (Test cases, testing documents, and JUnit testing) was the main topic that did not work for me. The purpose and types of test cases were well explained, but even with this knowledge coming up with large amounts of acceptable test cases was still very difficult. I particularly did not enjoy the section on recursion...it may be useful in coding, but the material was dry to me. all the uml topics. if it's used in the field then i guess it's necessary, but i felt like it was a topic to power through and we never really used it much. I am not sure about that because I haven't been to the long run yet sorting. i hate it. also recursion. but i guess we need those things. I absolutely did not like the UML lecture, and I have never been the type of person to plan things out before I do them, so I doubt I will find being able to do class diagrams very helpful or useful in the long run. I can't remember anything off the top of my head. At this moment, I don't know

~ QUESTIONS AND DETAILS ~	~ ANSWER MATRICES ~
	I don't believe lectures on hacking were especially useful for the future, however, they were among the most entertaining
	The process of software development series of lectures seems useless to me. I imagine that we will be taught the exact same things in following semesters; it will get old really fast.
	Every lecture/topic was useful.
	almost everything
	I don't think any topics did not work but I would have liked to have spent more time on iterators and the playlist project. I felt like the library class for that project was unbelievable difficult and maybe more time on learning the url class would have helped.
	None
	None
	Most of the theoretical things about software development, or the stages of software development. Then again, I'm not a CS major.
	THREADS, not useful to me as an electrical engineer, the general concept was understood, but i could not wrap my head around the details
	I struggled with recursion, but I think that's just a personal issue.
	I don't think that the networking topics were as useful to the class as some of the other topics.
	While the collections unit seemed interesting, I am not necessarily sure what exactly sure how that might help me in the future.
	N/A
	The discussion of threads was a bit rushed, and overall I think XP is pretty useless. I mean, it's been shown that pair programming only is beneficial when both of the programmers are inexperienced (Jui and Chan, "Pair programming productivity: Novice-novice vs. expert-expert", International Journal of Human-Computer Studies), so it would have made sense to take this into account during the lab section and make pairs of the lesser-skilled programmers while letting the experienced coders work solo.
	I can't think of any topics that "didn't work."
	Sequence diagrams didn't seem to be covered well. I feel like I went away with little more understanding of the topic than I had going into it.
	Runtime complexity seemed like it was just a review of what we learned in 101.
	I didn't like some of the 'big -O' material because it just became confusing and didn't seem to have a real purpose besides defining one function in terms of the other.
	Recursion
	I don't know if pair programming was very useful. I felt that most of this was either common sense or common courtesy. There are always going to be people that you work well in a team or pair with and people that you don't work as well with. You will be much more inclined to naturally do some sort of pair programming with the former and, if that is the case, you may even be more productive because of it. But everyone will have their own style and I feel like it is not something that should be taught a class but something that should be addressed in on a situation-by-situation basis
	nothing really
	The time we spent on different operating systems. I thought there wasn't a lot interesting in there and what there was could have been integrated elsewhere.
	Threads
	I felt that threading was not covered well enough to be useful.
	All lectures/topics had practical uses in my opinion.
	I really disliked trees and recursion, mainly because I found them frustrating. Looks like I'm in for a fun time next semester.
	Threading was too confusing that I didn't feel like I learned much.
	We spent too much time going over the differences between comparators and comparable. It doesn't seem all that useful
	java
	Perhaps a little more emphasis on the good things on object orientation

~ QUESTIONS AND DETAILS ~	~ ANSWER MATRICES ~
	I did not want to learn Swing, because I see it as an aging framework that should be abandoned in favor of a native GUI framework like SWT. Although I lost points for my team on a project assignment, I do not regret learning how to use SWT instead of Swing. complexity I know it is important but seriously it sucks!! Smack

4. How accurate is this statement for you: Pair Programming helped me learn the material better.

~

Question Type: Likert

~

contributed by Sherriff, Mark (mss2x)

Results for C S-201-0001, Sherriff, Mark

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
65	3.02	1.05	4 (6.15%)	19 (29.23%)	21 (32.31%)	16 (24.62%)	5 (7.69%)

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

~

Question Type: Likert

~

contributed by Sherriff, Mark (mss2x)

Results for C S-201-0001, Sherriff, Mark

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
65	4.06	0.66	14 (21.54%)	43 (66.15%)	6 (9.23%)	2 (3.08%)	0 (0.00%)

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

~

Question Type: Likert

~

contributed by Sherriff, Mark (mss2x)

Results for C S-201-0001, Sherriff, Mark

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
65	3.89	0.89	14 (21.54%)	36 (55.38%)	11 (16.92%)	2 (3.08%)	2 (3.08%)

7. 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 C S-201-0001, Sherriff, Mark

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
64	4.06	0.94	24 (37.50%)	26 (40.62%)	8 (12.50%)	6 (9.38%)	0 (0.00%)

8. How accurate is this statement for you: After taking this class, I am more likely to major or minor in CS.

~

Question Type: Likert

~

contributed by Sherriff, Mark (mss2x)

Results for C S-201-0001, Sherriff, Mark

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
65	3.78	1.10	20 (30.77%)	22 (33.85%)	14 (21.54%)	7 (10.77%)	2 (3.08%)

9. How accurate is this statement for you: After taking this class, I have a better appreciation for Computer Science.

~

Question Type: Likert

~

contributed by Sherriff, Mark (mss2x)

Results for C S-201-0001, Sherriff, Mark

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
65	4.12	0.80	22 (33.85%)	32 (49.23%)	8 (12.31%)	3 (4.62%)	0 (0.00%)

10. How accurate is this statement for you: After taking this class, I personally have a better understanding of fundamental concepts in Computer Science.

~

Question Type: Likert

~

contributed by Sherriff, Mark (mss2x)

Results for C S-201-0001, Sherriff, Mark

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
65	4.31	0.68	26 (40.00%)	35 (53.85%)	2 (3.08%)	2 (3.08%)	0 (0.00%)

11. The subject matter was challenging.

~

Question Type: Likert

~

contributed by Dean of the School of Engineering and Applied Science

Results for C S-201-0001

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
65	3.80	0.86	10 (15.38%)	38 (58.46%)	10 (15.38%)	5 (7.69%)	1 (1.54%)	1 (1.54%)

~ QUESTIONS AND DETAILS ~	~ ANSWER MATRICES ~

12. 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 C S-201-0001

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
65	4.23	0.63	21 (32.31%)	39 (60.00%)	4 (6.15%)	1 (1.54%)	0 (0.00%)	0 (0.00%)

13. 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 C S-201-0001

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
64	4.00	0.96	17 (26.56%)	39 (60.94%)	2 (3.12%)	3 (4.69%)	3 (4.69%)	0 (0.00%)

14. 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 C S-201-0001

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
65	4.12	0.74	18 (27.69%)	40 (61.54%)	5 (7.69%)	1 (1.54%)	1 (1.54%)	0 (0.00%)

15. The textbook increased my understanding of the material.

~

Question Type: Likert

~

contributed by Dean of the School of Engineering and Applied Science

Results for C S-201-0001

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
64	2.69	1.11	4 (6.25%)	10 (15.62%)	20 (31.25%)	19 (29.69%)	9 (14.06%)	2 (3.12%)

16. The course material was well organized and developed.

~

Question Type: Likert

~

contributed by Dean of the School of Engineering and Applied Science

Results for C S-201-0001, Sherriff, Mark

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
65	4.48	0.56	33 (50.77%)	30 (46.15%)	2 (3.08%)	0 (0.00%)	0 (0.00%)	0 (0.00%)

17. The instructor was knowledgeable about the subject matter.

~

Question Type: Likert

~

contributed by Dean of the School of Engineering and Applied Science

Results for C S-201-0001, Sherriff, Mark

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
64	4.75	0.47	49 (76.56%)	14 (21.88%)	1 (1.56%)	0 (0.00%)	0 (0.00%)	0 (0.00%)

18. The instructor was well prepared for class.

~

Question Type: Likert

~

contributed by Dean of the School of Engineering and Applied Science

Results for C S-201-0001, Sherriff, Mark

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
65	4.63	0.52	42 (64.62%)	22 (33.85%)	1 (1.54%)	0 (0.00%)	0 (0.00%)	0 (0.00%)

19. 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 C S-201-0001, Sherriff, Mark

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
64	4.30	0.85	32 (50.00%)	20 (31.25%)	10 (15.62%)	0 (0.00%)	1 (1.56%)	1 (1.56%)

20. The grading policy was fair.

~

Question Type: Likert

~

contributed by Dean of the School of Engineering and Applied Science

Results for C S-201-0001, Sherriff, Mark

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
64	4.25	0.80	26 (40.62%)	31 (48.44%)	5 (7.81%)	1 (1.56%)	1 (1.56%)	0 (0.00%)

~ QUESTIONS AND DETAILS ~	~ ANSWER MATRICES ~

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

~

Question Type: Likert

~

contributed by Dean of the School of Engineering and Applied Science

Results for C S-201-0001, Sherriff, Mark

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
64	4.61	0.52	40 (62.50%)	23 (35.94%)	1 (1.56%)	0 (0.00%)	0 (0.00%)	0 (0.00%)

22. 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 C S-201-0001, Sherriff, Mark

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)	Not Applicable (NA)
64	4.32	0.74	30 (46.88%)	23 (35.94%)	10 (15.62%)	0 (0.00%)	0 (0.00%)	1 (1.56%)

23. 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 C S-201-0001

Total	Less than 1 (NA)	1 - 3 (NA)	4 - 6 (NA)	7 - 9 (NA)	10 or more (NA)
64	1 (1.56%)	26 (40.62%)	32 (50.00%)	2 (3.12%)	3 (4.69%)

24. I learned a great deal in this course.

~

Question Type: Likert

~

contributed by Office of the Provost

Results for C S-201-0001

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
63	4.14	0.76	19 (30.16%)	37 (58.73%)	5 (7.94%)	1 (1.59%)	1 (1.59%)

25. Overall, this was a worthwhile course.

~

Question Type: Likert

~

contributed by Office of the Provost

Results for C S-201-0001

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
64	4.25	0.73	24 (37.50%)	34 (53.12%)	5 (7.81%)	0 (0.00%)	1 (1.56%)

26. 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 C S-201-0001, Sherriff, Mark

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
63	4.49	0.54	32 (50.79%)	30 (47.62%)	1 (1.59%)	0 (0.00%)	0 (0.00%)

27. 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 C S-201-0001, Sherriff, Mark

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
61	4.30	0.92	31 (50.82%)	21 (34.43%)	7 (11.48%)	0 (0.00%)	2 (3.28%)

28. Overall, the instructor was an effective teacher.

~

Question Type: Likert

~

contributed by Office of the Provost

Results for C S-201-0001, Sherriff, Mark

Total	Mean	Std Dev	Strongly Agree (5)	Agree (4)	Neutral (3)	Disagree (2)	Strongly Disagree (1)
65	4.52	0.62	37 (56.92%)	26 (40.00%)	1 (1.54%)	1 (1.54%)	0 (0.00%)

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

~

Question Type: Short Answer

~

contributed by Office of the Provost

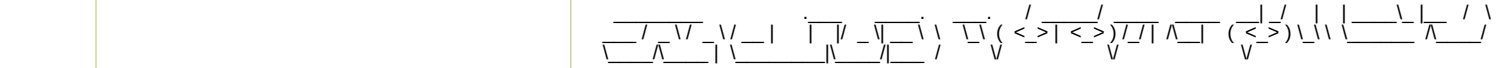
Results for C S-201-0001

Total	Individual Answers
35	*See below for Individual Results*

Great Teacher, Great Course

I lot of the little tips, like how to auto-format your code, generate getters and setters, and export java programs, that sort of stuff deserves in place alongside libraries, programs concepts, and team-working strategy, and it was really need to have those as well.

I attended every lecture I possibly could for this course, however, often times it is useful to go back and be reviewed on old materials. The fact that Mark Sherriff recorded his lectures made this very easy. I found the recorded lectures incredibly useful.

~ QUESTIONS AND DETAILS ~	~ ANSWER MATRICES ~
	It was so refreshing to have Prof. Sherriff for CS201 this semester. It is clear that he cares about his students, and that he has the ability and desire to actually teach the material, as opposed to just lecturing, like some of the professors in this department do. Prof. Sherriff is funny and approachable, and I really enjoyed this class. Prof. Sherriff, if you're reading this, thanks for a great semester and keep up the good work!
	The course was more difficult than CS 101E in my opinion, although to be quite honest, I didn't have as much time this semester to prepare myself and look over the material as much. There was a lot more to learn, and it seems like we were required to learn much more of the coding aspects on our own as compared to 101. Last year, I completely rocked 101, but in 201 this year, I feel like I just barely got by. As for Sherriff, he's a great professor. Most of my problems were simply me not having enough time to study more for the class. But Sherriff always seemed knowledgeable, answered all in-class questions appropriately and in depth, and he was never excruciatingly boring. The labs were painful though. 2 hours of my life wasted. Every week. This was a good class with a good professor; I think I was just a bad student.
	Very fun class and teacher.
	He's pretty nerdy and that's kind of distracting when all of the neckbeards and stuff start laughing at random internet memes, but he's a good teacher and he likes what he does. I think he's awesome, just the class he teaches is full of nerds.
	Sherriff is a BAMF.
	I felt that the tests did not adequately measure my understanding of the subject. I also felt that pair programming really helped the less competent member of the pair by making the other other member do all of the work. If pair programming raises the class average, it's just because the lowest half of the class isn't getting fairly graded on their efforts.
	What I appreciated most about Professor Sherriff was his enjoyable presentation of the material and laid back manner. It's easy to absorb the material in a laid-back environment. One aspect of the course that I found very worthwhile was programming in groups, something I had not done before. I also thought the labs were interesting and challenging.
	Good
	Testing section seems superfluous. Hard to apply concept when testing on small scale projects.
	N/A
	This course was overall an excellent class. I think that Pair Programming should be revised so that students can select their partners, or something to that extent. Think about it: in the real world, you'll be working with other people who understand CS, not people who may or may not be majoring in CS. I think individual software development and allowing students to work with others of their choosing would be a better system.
	Sherriff's 201 course is very well designed. Everything schedule and grading wise is effictively communicated to the students. He is prepared and knowledgable about the topics. His class is also comical and entertaining many times since he is energetic about the material the majority of the time. He is always open to questions and able to take a few jokes which is great. Outstanding professor overall.
	Amazing teacher. Infinitely better than my prior computer science teacher Kevin Sullivan.
	Mark Sherriff has a good sense of humor--I enjoyed his corny jokes.
	This course and its instructor were very excellent compared to other courses I have taken at UVA. Classes were interesting and studying the material covered in class and in review sessions provided adequate preparation for tests. Group homework assignments were rather time consuming, but the amount of work was fair and they helped in learning the course material.
	i kinda want to be a cpe because of this class and sheriff
	One of the best courses I have taken. Also, Professor Sherriff has been one of the best professors I have seen at UVA. Just a concern... some of the topics seemed to be rushed. Also, there was not a good introduction to the topics. I know there was not a lot of time but topics such as Operation Systems were covered very fast and the notes were very vague.
	
	Course is very interesting and the topics are all very helpful for computer science majors. For non-CS majors this class can be helpful in learning things that can be related to other subjects.
	Loved it.
	Professor Sherriff teaches to and designs his class for CS Majors. He realizes that there are non-CS majors taking the course, but neglects to compensate for the fact. Captivated in his "nerdy" CS world, Professor Sherriff explains simple concepts in highly technical terms that few truly understand. He is a great professor, however his techniques are best suited for higher level courses.
	a very good class. I learned a lot

~ QUESTIONS AND DETAILS ~	~ ANSWER MATRICES ~
	No comment Pretty good, entertaining, informative He was somewhat grumpy when I went to see him in office hours, but friendlier in class. Great Course. Very hard for students with weak CS foundation though. I dislike the fact that we had to do a two hour lab and we still only got 3 credit hours for the course. Especially when we were forced to stay in lab during the project. The project was a little tedious, but I learned a lot from it and we did do a little too much at the last minute. Other than that, a very effective course. Also, Mark Sherriff jumble: frisk her farm. While the slides were concise and the lectures were informative, the text book was not as helpful as it should be. Even though the textbook covered material thoroughly, it was very hard to study material from the book because they were too long and at points beyond the material covered in class. Mark Sherriff is one of the best teachers at this school. he knows how to teach, actually cares for his students, and tells you exactly what you need to know for tests and assignments so when you do bad it's probably your own fault. i wish half the teachers were like him but we get stuck with things like the physics department. thanks for the great year. Sheriff was a very good teacher due to his knowledge of the material and his ability to create an interesting and enjoyable. environment in the class. Pretty good course. I think a really important things is runtime. I dont think its ever explained well enough though INCLUDING this year.