

*** PROVISIONAL REPORT ***

UNIVERSITY OF TEXAS AT AUSTIN
Downing, Glenn P C S373 52200
E100 EXPANDED

COURSE-INSTRUCTOR SURVEY
SOFTWARE ENGINEERING

Spring 2017 DEPARTMENT COPY
Enrollment = 51
Surveys Returned = 49

	NUMBER CHOOSING EACH RESPONSE					NO. REPLIES THIS ITEM	AVG.
	Str Disag	Disagree	Neutral	Agree	Str Agree		
1 COURSE OBJECTIVES DEFINED-EXPLAINED	0	0	1	18	30	49	4.6
2 INSTRUCTOR PREPARED	0	0	0	10	39	49	4.8
3 COMMUNICATED INFORMATION EFFECTIVELY	0	0	1	16	31	48	4.6
4 STUDENTS ENCOURAGED-ACTIVE ROLE	0	0	2	7	40	49	4.8
5 INSTRUCTOR AVAILABILITY	0	0	2	14	33	49	4.6
6 COURSE WELL-ORGANIZED	0	1	5	12	31	49	4.5
7 STUDENT FREEDOM OF EXPRESSION	0	1	4	11	33	49	4.6
8 CLASS PARTICIPATION ENCOURAGED	0	0	0	8	41	49	4.8
9 ENGAGING INSTRUCTION	1	1	2	15	30	49	4.5
10 INST. HAD THOROUGH KNOWLEDGE OF SUBJECT	0	0	2	11	36	49	4.7
11 INSTRUCTOR EXPLANATIONS CLEAR	0	0	1	19	29	49	4.6
12 GENUINELY INTERESTED IN TEACHING COURSE	0	0	0	7	42	49	4.9
13 HELPFUL COURSE MATERIALS	1	3	12	12	21	49	4.0
14 ADEQUATE INSTRUCTIONS FOR ASSIGNMENTS	2	2	5	19	21	49	4.1
15 ASSIGNMENTS AND TESTS RETURNED PROMPTLY	0	0	6	20	23	49	4.3
16 ASSIGNMENTS USUALLY WORTHWHILE	1	0	2	20	26	49	4.4
17 STUDENT PERFORMANCE EVALUATED FAIRLY	0	2	4	19	24	49	4.3
18 STUDENT PERCEPTION OF AMOUNT LEARNED	0	2	2	20	25	49	4.4
	Vry Unsat	Unsat	Satisfact	Very Good	Excellent		
19 OVERALL INSTRUCTOR RATING	0	0	5	7	37	49	4.7
20 OVERALL COURSE RATING	0	2	5	20	22	49	4.3
	Excessive	High	Right	Light	Insuff		
21 STUDENT RATING OF COURSE WORKLOAD	1	18	28	1	1	49	
	Less 2.00	2.00-2.49	2.50-2.99	3.00-3.49	3.50-4.00		
22 OVERALL UT GRADE POINT AVERAGE	0	4	5	18	21	48	
	<u>A</u>	<u>B</u>	<u>C</u>	<u>D</u>	<u>F</u>		
23 PROBABLE COURSE GRADE	15	30	3	0	1	49	

For the computation of averages, values were assigned on a 5-point scale so that the most favorable response was assigned a value of 5 and the least favorable response was assigned a value of 1.

COMMENTS:

Total Number of Comments: 29

-
1. One of the best professors in the cs department.

 2. I loved taking SWE, and Downing is the model CS professor. His class is extremely structured and everything is really well outlined. I also really enjoyed the style of his class where he would call on students to participate. The only part of SWE that I did not enjoy were the first two projects, especially Netflix. Considering the second half of the class, the first two projects were a waste of time and made no sense. I would much rather have had the two first projects introduce web dev elements like Flask and React instead of making us learn everything at once for IDB 1. Also, the Netflix project was a joke if you didn't help with the caching repo. Lastly, I wish his Tests weren't like his quizzes and just gotcha questions. 5 5

 3. This class was excellent. The exposure to python, react, sql, docker, and more was a fantastic entry into software engineering in general. The final project was the most realistic thing I have done in a class, with a whole team of 6 and a LOT of moving parts to deal with. Our team ended with more than 600 commits, tons of discussion, and a website I am proud of.

 4. the tools All useful tools, glad we had to learn them. The non-group projects Collatz was a good project to get our environments setup. Netflix felt pretty useless. Because we were sharing caches you either had to race to be useful or wait so as to not waste your time. In the end, a classmate's helpful cache plus two lines of code pretty much solved the whole assignment so we felt kind of underwhelmed. The group project was tough, but very useful. All of the required libraries and tools were cool. Most of the speakers felt like they were just advertising their companies which I wasn't interested in except for Atlassian which had a technical problem to discuss! Made the talk more than worthwhile. More useful tech talks please!

 5. I really enjoyed the teaching and structure of this class, but the in-lecture content felt repetitive and mostly useless. I felt that many of the things we learned in class were way too specific and either out-dated or soon to be out-dated. The projects often involved good topics, but I think learning the ins and outs of a specific language, like Python, is not as useful as learning about more general software development topics I wish more time had been devoted to continuous integration, debugging techniques, good code practices, etc, instead of "how does this one weird quirk work in Python". As stated, I enjoyed the professor's method of teaching, just not the topics covered.

 6. The quizzes are often too abstract and not simply referencing the previous class. This makes them far too difficult to make it fair to be worth 15 of the final grade. The format of the first test completely blindsided many students. Granted, most of the material was covered, but many questions added in some abstractions that made the questions far more complicated than they needed to be to test someone's knowledge of the material. The first two projects were fantastic introductions to Python and good experiences. The website project was a very unpleasant experience. The instructions were incredibly unclear at times and the phase structure wasn't balanced 2 weeks was far too short for phase 2 perhaps move parts of phase 2 into phase 3.

 7. - The tools were all very well laid out and easy to use an integrate. - The non group projects did a good job of getting us accustomed to the tempo of the class. It also helped us get used to using the tools. No complaints for these. - I think that part 2 of the project was a little too difficult because there was so much to do. While the last part of the project there wasn't really much more to do. I think maybe in the future some of the tasks could be passed to part 3 of the project. - The speakers were definitely great.

 8. The group projects were the most valuable part of the course to me. I've never had to work with such large groups in school projects before which was a great learning experience in itself. One thing that I didn't like about the last project was that the project essentially was to create a website for some dataset that was retrieved over a rest api. But a lot of rest apis already have that website component so it seemed like we were just creating a lesser clone of an already existing website. Our group is partly to blame since we chose a data source that already had a great website but I think the project itself can be improved somehow to mitigate this. Anyways, great class I really enjoyed it.

 9. Projects such as 1, 3-5 are imperative for undergrads if they want to get anything worthwhile out of CS degree. I suggest giving more detailed instructions. Half of the course was an in depth study of python. Which I don't really see how it fits with the topic of the course. Less python and more focus in the papers and readings would be better. Additionally having done Project 1, project 2 was unnecessary. The quizzes are an unnecessary source of anxiety. Be more generous with the time. Lastly Help students prepare for the test. For our Test 1 some TA told students what the test was NOT going to be. And that left us those who didn't know anything, at a disadvantage. At least give away basic details of the test. Thanks for the course

 10. ...PROS... workflow taught me good coding practices, learned a lot from the projects learned a lot in terms of software tech and working as a team , readings were interesting and valuable, learned some design patterns, learned from and met people by working on in-class problems. Downing is an enthusiastic professor and fun to listen to. ...CONS... most if not all material taught in class is easily accessible online we will forget it all, then turn to stackoverflow or python docs , a largeportion of our grade is based on how good of a python interpreter we are, not enough writing for a writing flag class. ...SUGGESTED IMPROVEMENTS... less emphasis on details of python language, more in-class problems, make blog posting mandatory.

 11. Virtually everything taught in this class was immediately practical and relevant for modern software development. I don't think I can say that about any other course I've taken here at UT. However, the load was also somewhat high at times. The numerous articles, handful of books, various class notes, and projects throughout the semester proved to demand an often inordinate amount of time. Its workload, all things considered, rivaled 429's workload at times. I think it would be beneficial to cut down on some of the detail, such as certain, rather specific requirements on projects e.g., requiring that we use PostgreSQL rather than a database of our choosing . Some seemed like overkill and would just add confusion with little gain.

 12. The class would be awesome if not for the multiple choice programming tests. That doesn't make sense to me. I understand that it is much less work to evaluate, but it makes the tests more a matter of memorization. The class would be better with no tests at all than with bad ones.

 13. I think it would've been nice to have more choices on how to implement the backend for the last 3 projects kinda like how we were able to choose between angular or react for the frontend.

 14. The first two projects were an incredible waste of time easy and pointless other than quickly learning some of the tools like pylint etc. , especially since we could have used that time to start IDB earlier, which I felt was way too rushed. Being rushed

on IDB ruined the course for me, and I was looking forward to that more than anything. Also, the quizzes are a joke. I feel like they only exist to bring your grade down significantly for no reason, which I disagree with. Overall, you Downing are a great teacher, but I hope you reevaluate the structure of the course to be more focused on the IDB stuff, and less on other random things.

15. I found the course overall worthwhile because the projects were well designed and good learning opportunities. The lectures were good at teaching me Python, but I would have liked more time spent on software engineering practices. The last section covering strategies was more new and useful to me. I think the quiz and test questions felt like unfair evaluations of my knowledge, as they tested minutia, things we didn't adequately cover, or complicated problems that need more than the given time. The speakers were good when they were brought in to talk about their work and company Atlassian's was especially good, but the talk about AWS was not helpful for using AWS nor for getting to know the company.

16. The tools we have to learn are the best part of the class, Downing's classes build resumes like no other class can, this includes the tools one must learn for the group projects. The only really really good speakers this semester disclaimer I did not see Bloomberg were the Atlassian speakers.

17. Overall, I liked the course a lot. The grading criteria for the projects could have been a bit less ambiguous. Sometimes we were not sure if we would get the full points for certain part. Also, the quizzes were fairly difficult to complete in the short time period. I ended up missing a lot of questions because I was rushed.

18. Instructor was very good, however there were times where he would change the grading criterion the day of or day before the project was due, which was not fair to a lot of students.

19. Great class!

20. Thank you for teaching the class. The final project working as a group dynamic was a very good experience. I also enjoyed learning about the details of python. I felt that the refactoring book was a bit too long and had a lot of material in the treatment of the different techniques and was not sure to what extent I should have skimmed it. I also felt that there was a lot more than needed of material on aggregation and subquery of SQL that introduced a lot of confusion. Overall while the quizzes were extremely stressful, I learned many details useful for interviews.

21. I feel that there was a disconnect between the project and the lecture content. The project was a worthwhile experience, and the lectures were informative and well-organized, but they didn't have much to do with each other. It would have been nice to have more lectures relating directly to tools and design philosophies used for the project.

22. Would've been nice to have a little less emphasis on the Refactoring portion of the class.

23. What we were learning in class didn't always line up with what we were doing in the projects - for example reading about javascript way before actually using it so things didn't really 'click' in terms of doing them in practice, or talking about project resources a good deal after the point where a team moving at the proper pace would have wanted to use those resources. We also spent a lot more time on joins than seemed necessary. The group project was a good learning experience, both in terms of quickly picking up programming concepts and tools and in learning how to cooperate effectively as a software development team. A lot of the assigned readings were also really dry. I skimmed most of the refactoring chapters. But good class overall.

24. A couple issues I had with the course where the lack of clarity on the requirements for the final project, the awkward multiple choice format for the exams, and the very drawn out and not terribly useful readings. On the other hand, the lectures were very engaging and the choice of topics were the perfect mix of interesting and relevant. In the end, I learned a lot and got some good experience working with a bigger team on a large project.

25. I am very impressed with the professor's lectures. He is able to explain difficult topics from the bottom-up with astounding clarity and tons of helpful examples. Going into the class, I wasn't fond of being called on at random to answer questions, but I can see how it is helpful to his style of lecture as it allows him to speak as if he was having a 1-on-1 conversation with the students. The projects are okay. I love the exposure to new tools. I also like that the class changes from year to year because it means things are being innovated. But the project specifications are a bit strict and stifle creativity. The final project is a mimic as opposed to an original idea. In the end, our final sites are not useful in the real world.

26. Overall I very much enjoyed the course. The only real issue I came across was the challenge of learning several never before seen tools within a short amount of time. Often times this led to cobbling solutions for the projects together with a basic understanding of the tools that needed to be used. So, while perhaps the the lack of time did not allow me to properly asses a tool for the deadline, it did build my interest in them and so this class has certainly showed me many things I am now interested in looking further into.

27. 1. Autopep8 and pylint were finicky at times, but was useful overall. Coverage was useful for detecting unused lines of code and logic statements. Github, as always, was invaluable for source control. Unittest was extremely useful and is used in industry, and Travis CI was nice for automatically verifying builds. 2. Collatz and Netflix were pretty challenging and fun to do. 3. I would honestly recommend having a group evaluation at the end of IDB. It doesn't even have to affect the grade, but the implication of a peer evaluation would help keep people on track. I feel like some members of my group definitely contributed much less than others. 4. The speakers were informative and engaging Thanks for a great semester, professor!

28. Your course was great. The only suggestion I have is that I would have liked some sort of enforcement on students to do their share of the website projects. We had a group member that did nothing, did not show up to present our website, and will likely pass anyways. Which is extremely unfair. In the real world a group member like that would get fired.

29. Software Engineering with Downing is a challenging course but does an excellent job to bridge the gap between academia and industry. The projects gave me experience with tools and procedures that most students only get from internships. The lectures are pretty intense cold calling keeps students engaged but slightly on edge. The material we learned was useful and relevant. In my opinion, we spent a disproportionate amount of time on database concepts. I enjoyed the guest speakers. Downing is to the point and in my experience doesn't make many exceptions. What you see is what you get. Personable but very professional.