

## \*\*\* PROVISIONAL REPORT \*\*\*

UNIVERSITY OF TEXAS AT AUSTIN  
Downing, Glenn P C S373 51742  
E100 EXPANDED

COURSE-INSTRUCTOR SURVEY  
SOFTWARE ENGINEERING

Fall 2017 DEPARTMENT COPY  
Enrollment = 54  
Surveys Returned = 52

	NUMBER CHOOSING EACH RESPONSE					NO. REPLIES THIS ITEM	AVG.
	Str Disag	Disagree	Neutral	Agree	Str Agree		
1 COURSE OBJECTIVES DEFINED-EXPLAINED	0	1	6	17	28	52	4.4
2 INSTRUCTOR PREPARED	0	0	3	10	39	52	4.7
3 COMMUNICATED INFORMATION EFFECTIVELY	1	0	4	17	30	52	4.4
4 STUDENTS ENCOURAGED-ACTIVE ROLE	0	2	1	17	32	52	4.5
5 INSTRUCTOR AVAILABILITY	0	0	2	19	31	52	4.6
6 COURSE WELL-ORGANIZED	0	3	6	17	25	51	4.3
7 STUDENT FREEDOM OF EXPRESSION	1	0	3	19	29	52	4.4
8 CLASS PARTICIPATION ENCOURAGED	0	0	1	5	46	52	4.9
9 ENGAGING INSTRUCTION	2	2	8	14	26	52	4.2
10 INST. HAD THOROUGH KNOWLEDGE OF SUBJECT	0	1	2	17	32	52	4.5
11 INSTRUCTOR EXPLANATIONS CLEAR	0	0	5	18	29	52	4.5
12 GENUINELY INTERESTED IN TEACHING COURSE	0	0	1	8	43	52	4.8
13 HELPFUL COURSE MATERIALS	1	7	7	14	23	52	4.0
14 ADEQUATE INSTRUCTIONS FOR ASSIGNMENTS	2	9	10	14	17	52	3.7
15 ASSIGNMENTS AND TESTS RETURNED PROMPTLY	0	1	10	19	22	52	4.2
16 ASSIGNMENTS USUALLY WORTHWHILE	2	3	5	14	27	51	4.2
17 STUDENT PERFORMANCE EVALUATED FAIRLY	0	0	10	21	21	52	4.2
18 STUDENT PERCEPTION OF AMOUNT LEARNED	3	2	5	15	27	52	4.2
	Vry Unsat	Unsat	Satisfact	Very Good	Excellent		
19 OVERALL INSTRUCTOR RATING	0	1	4	14	33	52	4.5
20 OVERALL COURSE RATING	1	3	12	13	23	52	4.0
	Excessive	High	Right	Light	Insuff		
21 STUDENT RATING OF COURSE WORKLOAD	5	33	13	1	0	52	
	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	2	8	16	26	52	
	<u>A</u>	<u>B</u>	<u>C</u>	<u>D</u>	<u>F</u>		
23 PROBABLE COURSE GRADE	13	25	13	1	0	52	

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: 17

1. Calling in people in class helped me learn a lot, as well as the daily quizzes (even though they were a lot of pressure). I've heard of many groups have slacking members for their projects -- there should probably some way to deal with that. Other than that, one of the best classes I've taken at UT

2. I like how you engage with us during class and how clear and concise your lectures are. However, it would be nice if you explained the relevance of the readings to us during class. I feel that we read, got quizzed and tested, but never heard your opinion on them. The tools you introduced were useful -- GCP was very difficult though. I thought Collatz was a great segue from lecture to our first project -- however, the workflow was confusing, too many tools for what we were doing. Bootstrap was not the best for front-end work, but the group project requirements were reasonable. Did not think the Google guy said anything useful; Under Armour was just like the readings; Other two speakers were good. Overall, excellent course -- I've enjoyed it.

3. Compared to the professor's object-oriented programming course, SWE has been a hodgepodge of different subjects. The first half was quite intriguing, as it had to do entirely with python. However, towards the second half of the semester, the course sort of falls apart. We touched on a lot of different topics, like regex, SQL, database management and design, design patterns, refactoring, etc. However, none of it seemed connected or full. Most of these subjects warrant their own class (and have their own class), but these little "tastes" just didn't feel like they were adequately fleshed out. I would rather have less topics explained in fuller detail.

4. This class was ok in general. I really lacked work ethic this semester so I can't fairly judge some of the stuff like availability cause I never checked and stuff like that.

5. Enjoyed the class very much.

6. I found the teaching style of calling on students, asking them questions, hand holding them through the answers, and then talking over them when they did have a response extremely frustrating and not conducive to learning at all. Also, having almost 20% of your grade come from the daily quizzes is super unfair to those of us who are interviewing a lot and have to miss quite a lot of classes. I can understand wanting to have class participation and making sure people are engaged but please consider lowering the percentage so that it's not completely crushing for those of us that excel at everything else but those (and oftentimes for circumstances that are outside of our control). Also I didn't learn much because I have worked in industry.

7. Amazing class, Amazing professor, the only class at UT which I genuinely think will help me in the real world and the class I have most enjoyed. Prof. Downing is by far the best professor in the department, and his courses are well thought out and well structured.

8. I really enjoyed this course the only thing I could recommend is to maybe write a little more when you're giving day specific notes as it's difficult to get everything when we're jumping from code to code. Otherwise it was great!

9. The workload was okay, but factoring readings in can be excessive (some were long) on top of the daily quizzes/material/projects. I didn't like how many coding quiz questions required more thinking time than available. I can't think that quickly under pressure; my quiz performance suffered despite paying attention and instantly realizing my mistakes before the grade. There were many times Downing delayed sending clear instructions for projects which was irritating. 10pm deadlines isn't the norm and inhibited our team when we already had difficulty meeting earlier in the day. I also think Downing should be more proactive dealing with team conflict: we had one moocher on our team but Downing made us deal with it and they unfairly got credit

10. The group project was a lot of fun. Getting to build an interesting website with my friends while also learning a bunch of new tools was challenging but so rewarding. Very very glad you let us pick our groups.

11. Good course overall

12. My one complaint is that I didn't really like that the group projects seemed almost completely disconnected from the rest of the class, it didn't seem like we had a lot of guidance in doing them. We also had some problems with GCP, so maybe switching to a different provider like AWS would be good for later semesters.

13. Amazing to see someone who's been teaching for years constantly looks for ways to improve his class each semester. There might be some setbacks through trial and error but it's an upward trend overall!

14. I have taken OOP and SWE from Professor Downing. I thought OOP (I got an A-) was much easier than SWE (likely a C). Before SWE I had never written any JavaScript or SQL and I had barely touched Python. PROS: Downing's teaching and lecturing style is effective. The website building project is more meaningful than other projects in other classes. CONS: I would say quizzes but I think that is more on my part. BOTTOM-LINE: Knowing python beforehand can put you at a great advantage but if you practice you can catch up. If your group doesn't have web development experience, then the projects will be difficult. The amount of work was somewhere in between 314 or OOP and OS. Don't take this class for the writing flag. Downing is great though.

15. I really enjoyed the course, Professor Downing is a brilliant lecturer, and I definitely learned a lot, both in class and working on the projects. I would have liked to have more exams, I feel like I could have done a lot better on the second exam, and it would have been nice if there was one more exam, and each exam was weighted less. The first project, Collatz was a really good introduction to Python, but I spent a lot of time trying to fix things on github and travis, and I didn't use travis after that first project so I felt like it was a bit unnecessary, even though I understand it was to give us experience using continuous integration. It might have been more useful if we had used it for the other projects as well.

16. I loved the class and learning the materials presented. However, I felt as though the first exam was testing us on Python tricks and tips and readings which Downing knew nobody does. Give us some reason to read the readings rather than just for an exam. I enjoyed all of the materials we had to use. PlanItPoker wasn't very useful for the time factor but more so to establish what all we need to finish. Trello was the real deal. Thank you for the course.

17. While I understand the intent behind using HackerRank for examinations, the execution was lacking. Collatz gave a great intro to a set of tools that weren't used at all after Collatz. Postman is awesome, and would render Apiary redundant with some more

stringent requirements. Please use AWS, rather than GCP. The report should be on GitBook from the start. Trello is handy for generally tracking tasks, but as a sort of "Story tracking" program it becomes less useful as more and more stories are added and closed (leading to individually counting the number of issues closed, and having to scroll forever). Plan It Poker is a good intro to Agile-like estimation, but when most groups bs it the value is lost, leaving only extra tedium.

-----