

*** PROVISIONAL REPORT ***

UNIVERSITY OF TEXAS AT AUSTIN
Downing, Glenn P C S373 51625
E100 EXPANDED

COURSE-INSTRUCTOR SURVEY
SOFTWARE ENGINEERING

Spring 2018 DEPARTMENT COPY
Enrollment = 55
Surveys Returned = 53

	NUMBER CHOOSING EACH RESPONSE					NO. REPLIES THIS ITEM	AVG.
	Str Disag	Disagree	Neutral	Agree	Str Agree		
1 COURSE OBJECTIVES DEFINED-EXPLAINED	0	1	3	19	30	53	4.5
2 INSTRUCTOR PREPARED	0	0	1	13	39	53	4.7
3 COMMUNICATED INFORMATION EFFECTIVELY	0	0	2	17	34	53	4.6
4 STUDENTS ENCOURAGED-ACTIVE ROLE	0	0	1	13	39	53	4.7
5 INSTRUCTOR AVAILABILITY	0	0	3	15	35	53	4.6
6 COURSE WELL-ORGANIZED	0	1	5	20	26	52	4.4
7 STUDENT FREEDOM OF EXPRESSION	0	0	8	13	32	53	4.5
8 CLASS PARTICIPATION ENCOURAGED	0	0	1	6	46	53	4.8
9 ENGAGING INSTRUCTION	1	0	3	18	31	53	4.5
10 INST. HAD THOROUGH KNOWLEDGE OF SUBJECT	0	0	5	16	32	53	4.5
11 INSTRUCTOR EXPLANATIONS CLEAR	0	0	5	21	27	53	4.4
12 GENUINELY INTERESTED IN TEACHING COURSE	0	0	0	6	47	53	4.9
13 HELPFUL COURSE MATERIALS	1	3	15	16	18	53	3.9
14 ADEQUATE INSTRUCTIONS FOR ASSIGNMENTS	0	9	13	16	15	53	3.7
15 ASSIGNMENTS AND TESTS RETURNED PROMPTLY	4	16	14	9	10	53	3.1
16 ASSIGNMENTS USUALLY WORTHWHILE	0	0	1	21	31	53	4.6
17 STUDENT PERFORMANCE EVALUATED FAIRLY	0	1	1	27	24	53	4.4
18 STUDENT PERCEPTION OF AMOUNT LEARNED	0	0	2	16	35	53	4.6
	Vry Unsat	Unsat	Satisfact	Very Good	Excellent		
19 OVERALL INSTRUCTOR RATING	0	1	1	13	38	53	4.7
20 OVERALL COURSE RATING	0	1	3	20	29	53	4.5
	Excessive	High	Right	Light	Insuff		
21 STUDENT RATING OF COURSE WORKLOAD	2	30	20	0	1	53	
	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	5	15	31	53	
	<u>A</u>	<u>B</u>	<u>C</u>	<u>D</u>	<u>F</u>		
23 PROBABLE COURSE GRADE	27	20	4	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: 29

-
1. Great professor!

 2. Need a better a better way to evaluate students in this course. Like in our group, I think I do much more work than I should have. Some of the groupmates just wrote a few codes. They do deserve partial credit but not 0 point or full credit.

 3. In my opinion, much of the first half of the course is just review of Python which is great if you are just learning it. However, with mandatory (graded attendance) lectures, it felt like a waste of time to be forced to go to these lectures. Possibly being able to test out of these lectures may be something interesting to try out.

 4. This was overall a very enjoyable class.

 5. Thank you Professor Downing for an extremely well-organized and instructive course. :)

 6. With most projects the instructions are vague. This leads a lot of times to lots of questions on piazza just to clarify what needs to be done. Also, it felt like project requirements would change while the project is in progress causing extra work for groups who weren't already working in that way.

 7. The phases of the IDB project need to be graded quicker. It's difficult to incorporate feedback from the last phase when the grades are released a day before or on the same day the next phase is due. Also, the graders' comments are often vague and unhelpful. The Collatz project was a nice way to get familiar with working in Python but the required workflow felt too strict. Because you don't have to follow the workflow in the rest of the projects, it felt unnecessary in hindsight. IDB was interesting and a good learning experience. It would have been nice to have a little more freedom in choosing what tools to use though.

 8. The class experience was great but one complain was the project grades come back too late. The grades for previous project basically came back during the day that the next one is due so we really don't have much time to fix our previous mistake.

 9. I didn't like how for some of the group projects the instructions were vague on some things. Also it would be helpful if the projects were graded some time before the projects were due and not the day before so we have time to make changes.

 10. I wish the spacing for the last and 2nd to last phase was spaced more. The 2nd to last phase was a crunch time and we were not able to publish or implement feedback until the day the project was due. I wish TAs would grade the project faster so that we can implement the feedback and correct any changes, but I understand that it takes time, so I wish the 2nd to last phase lasted a week longer, since we had an extra week at the end of the semester.

 11. Glenn Downing is an excellent teacher. There are several things wrong with this class though: 1) He needs to update the reading materials. This stuff he's assigning is out of date. Good thing Martin Fowler is coming out with another book. 2) He needs to figure out the formula for the projects. It seems we were guinea pigs for some of his ideas this semester and they weren't very well fleshed out. They could've been better. 3) He needs to expect more out of us. The material in the lecture was pretty boring and easy. This is UT CS, lets go deeper into software engineering.

 12. Excellent course! Out of all the years I've been in CS, I feel that this class was the most helpful in preparing students for interviews/internships.

 13. Grading of the projects took too long making it difficult to apply the feedback we got from the TAs since we received the feedback on the day the next part of the project was due.

 14. I LOVE DOWNING!

 15. -I enjoyed the breadth of knowledge/wide spread of tools Downing encouraged us to learn. -I really appreciated the fact we had one non-group project. I found this project fun to do and I did learn quite a bit. -I felt very mixed about the group project. It took up an enormous amount of time this semester and I found the workload quite heavy. However, other group members probably thought it was a breeze and that it barely took up anytime. Therefor, I would encourage Downing to have group sizes of more like 4 people. It was way too difficult to utilize all 6 people. Grades were often returned very late and we had to argue back for senseless points taken off every time, so maybe better monitoring of TA grading would be good.

 16. The last two phases of the IDB project, the projects were graded the day before the next day was due. It was hard to fix the problems that were wrong in the last phase because there was no time.

 17. I am very fond of the guest speakers and the lecturers that were brought in throughout the semester. All of them are software engineers that have a lot of experience in the field. Listening to them speak about their experiences and their work environment in software engineering was one of the highlights of my semester. I hope the professor continues to bring them in future semesters. One thing I think need to be improved is the class lectures. The lectures tend to focus on details in python or javascript, which I think are not very useful. It would be better if we focus more on design patterns and useful concepts (singleton, factory, dependency injection, databases etc). This would help us prepare for internships/ full-time jobs.

 18. The Collatz project seemed a little out of place. At least regarding the Travis integration, as we never used that later on.

 19. Good course overall, with good emphasis on teaching us various tricks. More time on projects would make it better.

 20. Most of the tools were easy enough to understand once you knew your way around them. Docker was really challenging to get working correctly though so we eschewed it for the most part. Collatz was simple to do, no complaints. As for the group projects, I would appreciate more direction in how to create and work on projects on AWS and using Bootstrap, React, etc. We barely touched on those topics in class yet they were crucial to doing the in-class projects. Something I didn't like was that we were expected to already know JavaScript and Python coming into the projects despite those languages also being taught during the class. It just doesn't make much sense to me. Of the speakers I went to, I found their chats to be really interesting.
-

21. I hate Gitbook

22. I really enjoyed professor Downing, he was interesting to watch and all of the talks I went to were cool. I would like to see more in class coverage of the project material though. Explaining how some aspects of REACT work would've been nice. I didn't do any of the front end work for our project but I know there was a lot of confusing stuff with REACT. I would also recommend not using gitbook, maybe the new client it better and fixed some of the problems I had when editing with it but the legacy was incredibly frustrating to work with. Granted I did use the web editor so take that with a grain of salt, the difficulty was at the very least partially my fault.

23. I enjoyed the projects and how much freedom we were given. I don't think anything specific should be changed about the course.

24. Glenn Downing is an incredible person in the UTCS department. He's involved, knowledgeable, friendly, and fair. His interest in his students endeavors and also in the orgs around the department really stand out to me. The tools that we're encouraged to learn in this course are the most real-world-applicable I've gained from a UT course and I think every CS major should take a Downing course at least once. I only wish I got to know Professor Downing a little bit better as I feel like he'd be a great person to get career advice from.

25. Most of the professor's "lectures" involved calling on the class to teach his own material, which many of us for whatever reason (surprise, confusion, introvertedness, or simply not knowing) didn't respond to immediately or clearly. For this reason, as well as the professor typing lecture notes as he went (often pausing for errors), half of the class periods were near silence. The projects were amazing, I'd recommend or retake the class for those alone. However the lecture topics had very little in common with those. Said topics were often trivial enough to just reference an online tutorial for and learn as I worked. I wish the lectures covered overviews of React, Flask, or other tools we could use on the projects.

26. The project specs were incredibly vague in this course compared to OOP. I know that in general the project is very subjective, but creating a search and having it be "google-like" doesn't leave a lot to go off of. Our team consistently was unsure of what exactly was expected of us and felt that even the piazza post explanations were not very clear about to what requirements we should make each part of the site.

27. As in the Object-Oriented programming course (which I took in the previous semester), this course was very enjoyable and exceeded my expectations. I greatly appreciated the additional HackerRank examples in class. Being able to repeatedly go back and implement concepts (and discuss them with other students) helped me to feel more comfortable with the course content. Finally, the assignments were more interesting than I had expected. Though there were some difficulties here and there (i.e., noticing a bug right before a due date and scrambling to fix it), I found the experience to be a worthwhile and generally positive one. I would highly recommend this course.

28. Professor Downing's lectures were wonderful and engaging, and I think Collatz served well as an introduction to Python. The IDB project did feel disconnected from the rest of the class though. I also think it would've helped to know from the start everything that would be required in the end, especially since getting back grades and feedback took some time. That said, I do think the project was a great one. I thought the tools required of us were useful to learn, perhaps for the exception of Postman. I thought HackerRank served as a great tool for both practice exercises and exams. Overall, I'm definitely glad I took the class and would recommend it to other students under the caveat that they be prepared for a huge time commitment.

29. Not a fan of the hacker rank for tests. Not comfortable with such a buggy interface for submitting answers. Very often no issues in code but hacker rank would make me do everything again to recognize code.
