

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

UNIVERSITY OF TEXAS AT AUSTIN  
 Downing, Glenn P            C S373            85610  
 E100 EXPANDED

COURSE-INSTRUCTOR SURVEY  
 SOFTWARE ENGINEERING

Summer 2019 DEPARTMENT COPY  
 Grade-eligible enrollment = 36  
 Surveys Returned = 33

	NUMBER CHOOSING EACH RESPONSE					NO. REPLIES THIS ITEM	AVG.
	Str Disag	Disagree	Neutral	Agree	Str Agree		
1 COURSE OBJECTIVES DEFINED-EXPLAINED	0	0	2	11	20	33	4.5
2 INSTRUCTOR PREPARED	0	0	0	8	25	33	4.8
3 COMMUNICATED INFORMATION EFFECTIVELY	0	0	1	10	22	33	4.6
4 STUDENTS ENCOURAGED-ACTIVE ROLE	0	0	0	5	28	33	4.8
5 INSTRUCTOR AVAILABILITY	0	0	2	13	18	33	4.5
6 COURSE WELL-ORGANIZED	0	1	6	12	14	33	4.2
7 STUDENT FREEDOM OF EXPRESSION	0	0	1	12	20	33	4.6
8 CLASS PARTICIPATION ENCOURAGED	0	0	0	3	30	33	4.9
9 ENGAGING INSTRUCTION	0	0	5	12	16	33	4.3
10 INST. HAD THOROUGH KNOWLEDGE OF SUBJECT	0	0	0	5	28	33	4.8
11 INSTRUCTOR EXPLANATIONS CLEAR	0	0	1	12	20	33	4.6
12 GENUINELY INTERESTED IN TEACHING COURSE	0	0	0	3	30	33	4.9
13 HELPFUL COURSE MATERIALS	0	3	10	8	12	33	3.9
14 ADEQUATE INSTRUCTIONS FOR ASSIGNMENTS	0	5	9	10	9	33	3.7
15 ASSIGNMENTS AND TESTS RETURNED PROMPTLY	0	0	6	15	12	33	4.2
16 ASSIGNMENTS USUALLY WORTHWHILE	0	0	1	13	19	33	4.5
17 STUDENT PERFORMANCE EVALUATED FAIRLY	1	1	7	12	12	33	4.0
18 STUDENT PERCEPTION OF AMOUNT LEARNED	0	0	1	10	22	33	4.6
	Vry Unsat	Unsat	Satisfact	Very Good	Excellent		
19 OVERALL INSTRUCTOR RATING	0	0	2	8	23	33	4.6
20 OVERALL COURSE RATING	0	1	7	9	16	33	4.2
	Excessive	High	Right	Light	Insuff		
21 STUDENT RATING OF COURSE WORKLOAD	7	17	9	0	0	33	
	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	3	4	15	11	33	
	<u>  A  </u>	<u>  B  </u>	<u>  C  </u>	<u>  D  </u>	<u>  F  </u>		
23 PROBABLE COURSE GRADE	11	13	9	0	0	33	

-----  
 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: 19

---

1. This was not a course designed to be taken over the summer. We only had a week to do most of the projects, and this became nearly a full-time job for the members of my group. We worked upwards of 40 hours per week and still struggled to get everything done in time. We also weren't prepared well for the tests. My biggest complaint with this class, however, was the lack of direction on the projects. I know that we were supposed to figure most of it out on our own, but given that we had such little time to complete the projects, the lack of direction was a real hurdle. I had no prior experience to web development, so it was insanely difficult to learn what I needed to do and then actually do it within the timeframe of the projects.

---

2. This class was very valuable, I definitely need to know the material taught to further my CS career. However, this shouldn't be a summer class in my opinion. The workload for the projects was simply too high in my opinion, there were weeks where some of us were working 12+ hours every day in order to complete one of the phases (which we didn't complete anyway, and, in fact, no one in the class completed on time). The tests/exams/quizzes are fine, however, the projects either need better spaced out due dates or should have fewer requirements to complete than the normal semester. Another option would be to make this class into a whole semester summer class, with more spaced out due dates for the projects.

---

3. The coursework was very heavy

---

4. The project was useful and highly relevant but was also very rushed. The rubrics for the project were also fairly vague. For example, phase 3 was primarily about searching, sorting, and filtering, but the points for these requirements only appeared in the technical report section of the rubric (when they were probably also weighted on other sections of the rubric like the website).

---

5. There's a lot of work for this course over the summer session. For a spring or fall class I think the work would be ok. I dislike how the class material doesn't reflect the projects. There doesn't seem to be a good reason to not teach information that directly affects the projects. Cold calling was kind of annoying but I understand why you do it. Overall the class was alright but taking it in the summer was like os 2.0

---

6. The professor was a great lecturer. I definitely thought the lectures were very unique. I appreciate that he cold calls, because it requires the constant attention of his students. He explains everything step-by-step, which is great for people to learn without being confused. The tests are difficult, and I believe there should be more practice problems or study material for students to prepare in advance. The group projects should be given with more instructions and details on what is expected. A great professor overall who likes teaching.

---

7. The IDB project was one of the most interesting college assignments I've had thus far. I'll look back on this class as being one I enjoyed attending. I never came to like the cold calls, but at least I became more comfortable with them. Honestly, the risk of being cold called kept me focused on every lecture, something I've been very bad about at UT.

---

8. Wonderful lectures where I learned in depth on python, SQL, and Java. Since very little web development was covered in class (understandable due to time constraint of lectures), I would've liked to been able to choose our own groups so that everyone could have similar motivation levels.

---

9. cool class

---

10. :)

---

11. My one complaint about the course has been addressed by Downing many times. That what Downing went over in class and the content of our projects didn't have much to do with one another. I only used Python once in the project and that was the majority of what was taught in class. I would've liked some lectures about React or JavaScript.

---

12. The instructions on course assignments were quite decentralized and sometimes it was difficult to find information concerning some aspect of an assignment/phase. Some requirements weren't requirements on the project (mocha, d3, tests living in postman) and our team only learned about this by word of mouth from other groups after/around time of submission, or from TA comments. We were told on two occasions that we were the only team implementing our solution using the method/tool described as a requirement, which was frustrating when better tools existed. The wording on requirements for some of the phases was somewhat vague and didn't match the rigidity with which we were graded.

---

13. Hands down one of the best courses I have taken at UT. Professor Downing is an incredible teacher, and his method of teaching (socratic) is very engaging, and it forces me to pay attention as well as helps me retain and understand the material better. I honestly wish that more professors followed this style. I gained a lot from this course.

---

14. The lecture/project disconnect was explicitly acknowledged, but still annoying, as it felt like a good chunk of lectures were irrelevant to what was required for the projects (aside from learning Python and Collatz for the Collatz project). Learning SQL would have been more useful earlier on in the semester, so it could be applied to the database construction phase. Also, learning about design patterns and refactoring, which is more relevant to the subject of software engineering, was pretty much at the end while a good chunk of the lectures were just doing things in Python. The projects had a quick turnaround but that's due to the constrained summer schedule, which should be taken into account more (although I think grading was easier).

---

15. Some of the grading criteria wasn't clear. Some of the projects took too long to grade. Group projects were very challenging, mostly due to my specific group and their behavior. Wasn't enough time over the summer for doing the projects. First test could have gone better if function unpacking was explained before the test. Second test was too easy besides the question about factory methods, which was quite difficult.

---

16. The project was really intense, but I came out the other side having learned so much. Also, I'm annoyed that I didn't get credit for a blog post because we had to submit the link on canvas. I guess it's necessary for grading. Overall, a worthwhile class!

---

17. The class was great. I would recommend, if at all possible, use more visuals and diagrams to explain confusing or elaborate topics.

---

18. The course was fast-paced, informative, and kept me on my toes. This meant that I gained quite a bit from the experience, but I imagine that I could have done better if the direction and goals of the course were expressed in more detail near the beginning of the semester. Learning what to do on the projects as they were released left my team disorganized and unable to work effectively on the learning aspect of the projects. As we were trying to just finish the required work, we didn't have the time

to do things that would benefit us as students (such as switching roles periodically). Other than that, I would like to say that the lectures were top-notch, likely the best I've experienced thus far.

-----  
19. This is honestly one of the most practical and useful classes I have taken at UT. I would not change anything with the sole exception that there needs to be MUCH MUCH more practice material for the exams. If you expect exams to be 40% of the class grade, it is only fitting that students be able to reinforce those skills prior to the examination. In general, it would be a good idea to have much more optional practice on HackerRank for each coding exercise. The cheat sheets and process of creating them are irrelevant and useless, even if the intent is that students be able to study by creating them. Even five practice questions without solutions posted on Piazza would be infinitely more useful, especially for refactoring.  
-----