

*** PROVISIONAL REPORT ***

UNIVERSITY OF TEXAS AT AUSTIN
Downing, Glenn P C S373 86360
E100 EXPANDED

COURSE-INSTRUCTOR SURVEY
SOFTWARE ENGINEERING

Summer 2018 DEPARTMENT COPY
Enrollment = 44
Surveys Returned = 42

	NUMBER CHOOSING EACH RESPONSE					NO. REPLIES THIS ITEM	AVG.
	Str Disag	Disagree	Neutral	Agree	Str Agree		
1 COURSE OBJECTIVES DEFINED-EXPLAINED	1	1	2	16	22	42	4.4
2 INSTRUCTOR PREPARED	0	0	1	7	34	42	4.8
3 COMMUNICATED INFORMATION EFFECTIVELY	0	2	1	7	32	42	4.6
4 STUDENTS ENCOURAGED-ACTIVE ROLE	0	0	1	7	34	42	4.8
5 INSTRUCTOR AVAILABILITY	0	0	3	11	27	41	4.6
6 COURSE WELL-ORGANIZED	0	1	6	13	22	42	4.3
7 STUDENT FREEDOM OF EXPRESSION	0	0	4	7	31	42	4.6
8 CLASS PARTICIPATION ENCOURAGED	0	0	1	3	38	42	4.9
9 ENGAGING INSTRUCTION	1	1	2	13	25	42	4.4
10 INST. HAD THOROUGH KNOWLEDGE OF SUBJECT	0	2	1	8	31	42	4.6
11 INSTRUCTOR EXPLANATIONS CLEAR	0	2	1	11	28	42	4.5
12 GENUINELY INTERESTED IN TEACHING COURSE	1	1	1	6	33	42	4.6
13 HELPFUL COURSE MATERIALS	1	1	11	14	15	42	4.0
14 ADEQUATE INSTRUCTIONS FOR ASSIGNMENTS	4	1	5	14	18	42	4.0
15 ASSIGNMENTS AND TESTS RETURNED PROMPTLY	0	0	1	10	31	42	4.7
16 ASSIGNMENTS USUALLY WORTHWHILE	0	1	2	7	32	42	4.7
17 STUDENT PERFORMANCE EVALUATED FAIRLY	1	1	7	19	14	42	4.0
18 STUDENT PERCEPTION OF AMOUNT LEARNED	1	0	2	9	30	42	4.6
	Vry Unsat	Unsat	Satisfact	Very Good	Excellent		
19 OVERALL INSTRUCTOR RATING	0	1	4	4	33	42	4.6
20 OVERALL COURSE RATING	0	2	4	13	23	42	4.4
	Excessive	High	Right	Light	Insuff		
21 STUDENT RATING OF COURSE WORKLOAD	9	17	15	1	0	42	
	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	1	4	7	12	18	42	
	<u>A</u>	<u>B</u>	<u>C</u>	<u>D</u>	<u>F</u>		
23 PROBABLE COURSE GRADE	14	20	8	0	0	42	

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: 25

1. Incredibly well organized and informative, practical and useful in industry. Lectures engaging and informative, but I wonder if learning tiny edge cases is the most practical thing to learn in those languages. Quizzes were challenging but doable. After this class I learned a lot about those languages that I am comfortable using them. Class is divided into two parts with the website project and the lectures. In the website we applied so many tools actually used in industry to create a project with a group, in an experience that feels like a real job. That part was difficult and took lots of time, but rewarding. Individual project prepared us for group project. Enjoyed guest lectures to learn not just about programming, but industry.

2. 1. Although the lecture is strongly disconnected from the project, I think the knowledge from the lecture about Python and JavaScript is very useful when you know basic of React and Flask. Using iterator, lambda, etc make life lot easier. 2. Refactoring is very important in a group project since your teammate will often read your code, and do some modified in it, so a good structure will help a lot. Especially in React, you have a component system. 3. Collatz is a great introduction to git system, definitely keep it. 4. To be honest, I hate react at first glance, but once you know how to use it, it becomes very powerful. 5. I think peer review should somehow actually affect the final grade since some people really doing nothing.

3. Prof. Downing is one of the top three lecturers I've worked with during six years of undergraduate and graduate classes in the Natural Science and Engineering departments at UT. CS373 isn't the course where I learned the most, but Downing is probably the best at conveying information of all the professors I've learned from. It appears that he has put enormous effort into optimizing his lecturing style. However, that is a double-edged sword. His methods are so engaging that his lectures are a bit exhausting. I wouldn't want to take multiple other courses concurrently with this one. Still, I appreciated the hands-on nature of the assignments and I expect to use this material in most of my future work. Overall an outstanding course.

4. CS373 is probably the most worthwhile class I've taken so far at UT. Learned a lot about Python, Javascript, and web development. The Collatz project was a great intro to git and the web project was cool too. Although I might not go into web development as a career, it's nice to have more tools in my toolbox I can use. The class also helped me get rid of my tendency to procrastinate since the project required my group and I to work together 5-6 days a week for 7-10 hours a day.

5. I think the quizzes are a bad way to force attendance. If you're even 5 minutes late, it's a miss. If anything, have it at the END of class so people can still get attendance for late show ups.

6. I loved learning React, I think it's a really cool/intuitive way to build GUIs. I think the Javascript part of lectures felt like an addendum, and didn't add much. I would have preferred a bigger focus on SQL, and I would have preferred if the SQL portion of the lectures came earlier in the class schedule, like around Phase 2 when we were working on getting our database up and running.

I liked the guest lecture by the three speakers a lot and found it informative, but I thought the timing of the guest lecture wasn't great. I remember really wanting further clarification on requirements for Phase I, but office hours were cancelled. I think the guest lecture would have been better suited for after Test 1, when we didn't have a deadline.

7. The class is well taught and I loved the material. However, if you are stuck in a bad group, your grade will suffer since a good portion of your grade is based on the group projects.

8. Loved the class. The only thing I didn't really enjoy was using docker. I've heard that docker only recently became required. I'm not sure why that is, although it is a useful tool, I didn't find it to be particularly important to the project. Your lectures were great. Very engaging, forced me to listen (even when I didn't want to HA) but regardless it's obvious that this class is supposed to be like a prerequisite of what the real world will look like. I assume that you've taught the class during the summer before, but it does seem a little fast paced (it of course has to be). I wish that perhaps we would be warned a little more about how much more work phase 2 would require...? BUT, regardless, thank you.

9. I really enjoyed Mr. Downing's class and found him to be very engaging with his students. I think my biggest gripe is the number of technical errors I encountered during projects and the fact that Mr. Downing did not show us how to use some of the tools for these projects thoroughly. Overall, I found Mr. Downing to be a pleasant individual, who I feel teaches well, although I think I may have liked taking his class in the Fall or Spring, instead of in the summer.

10. I found the group project to be a very worthwhile learning experience. However, I would have liked if there was more instruction during lectures on the tools we had to use. For example, getting Docker set up was extremely confusing, given that we were told to just go out and get it done. I feel that even a basic lecture like the one Taben gave us on Docker would have been extremely useful in the beginning of the course. Being a summer course, I can't help but feel that our group was overwhelmed with the amount of tools we were expected to learn and use for the project in the short amount of time that we had. Besides that, it has been a very enjoyable course and has taught me a lot.

11. The lectures were very engaging. It can be a little confusing when you switch from Python to JavaScript and back to Python. It would also have been nice to have all your files available online. Very inconsistent with their availability. GitLab even went down the night before an exam. I spent a lot of time learning the tools on my own. It would be nice to have some helpful documentation for each tool (how to install, etc.). Collatz was a good project. I spent a lot of time figuring out Docker, but once I had it figured out it was smooth sailing from there. I really enjoyed the group project. I look forward to showing it off to employers and my friends/family. The speakers were very entertaining and inspirational.

12. This course was precise, thorough, and organized. Downing spins life into what otherwise is ordinary lines of code, but he maintains the code simplicity no matter how complicated the subject gets. Downing animates the syntax of any language to explain his subjects in an admirable way. His approach is systematic and helped me understand and appreciate concepts that I was familiar with, but I could never appreciate or barely recall for that matter. Downing's cold calling kept me paying attention to every minute of his class because it is always an interesting conversation. Advices: (1) I would have appreciated a couple of iterations on difficult concepts. (2) Please less questions on the quizzes or more time.

13. While I agree with not teaching any specific tool used for the projects (after all, the tools we used are different from the tools we would have used two years ago which are again different from the ones four years ago and so forth), since being able to choose a framework based on project needs and learn it is both common and important in my work experience, we didn't get to material I expected from this course until the last week of class. The lectures on refactoring and design patterns were ones I would have liked a great deal more of, and fit the "Software Engineering" moniker much better than the js/python basics and

trivia that filled much of the rest.

14. From the course, we can learn a lot of things, like the use of many tools. However, we have to learn them all by ourselves. When problems happen, it's usually hard for us to debug. So some small bug may cost us a lot of time and will lead to a very low score. I think the professor can provide more tutorials. Also, there are some teammates which don't do much in our group project. I hope peer review we write can have its use.

15. great lectures, well explained and articulated with useful and interactive hackerrank exercises. far too little direction given on how to use the tools. lecture had almost nothing at all to do with what we actually had to do for homework. collatz was fine and useful for learning some of the more basic tools. in theory it was fun to work on such an independently driven project, but we were basically expected to figure out and learn everything ourselves, which led to a lot of frustration because some of the tools (e.g. figuring out how to host the site and database) were really complex and hard to find good documentation on. speakers were great, loved the guest lectures

16. The reading is a lot to ask for with everything else we do. Also, I was not a fan of the way the quiz system worked. Other than that, I enjoyed the Socratic method in action and enjoyed the course.

17. Most of the projects in this class were not related to the material taught in class. While I understand the importance of being able to teach ourselves the material, some helpful links as to where to get started or what not to do would have been very helpful since during the summer there isn't a lot of time for trial and error. Even though I've felt overwhelmed in this class, I believe it was a worthwhile experience. Thank you for teaching the class and making us actually try to talk and interact with each other instead of secluding ourselves.

18. This class was my first foray into larger projects/group projects, so it was strenuous at times but a great learning experience. I really appreciate working through examples in the lectures, it makes it very clear what actually is going on.

19. The project IDB is difficult and the professor did not teach anything about it.

20. Lectures were good but boring at times when people couldn't answer questions. The number of tools we used was daunting but extremely useful later on. The non-group project was too easy coding-wise but taught us proper workflow. The group projects were fun because I had a great group, but the time crunch was very real. The speakers were awesome and gave me a great insight on what it's like to work a SWE job at a tech company. My only suggestion is this: I'd like tests to be worth less and projects to be worth more in the future, since I feel that the projects are MUCH more helpful in the long run than the stuff we learn in class. I also feel like grading was too lenient on the projects--many groups got away with Bs for incomplete websites.

21. I liked how interactive this class was because no other CS classes do that. Projects were extremely hard because I had some lazy members in my group. Other than that, you're a great professor! Thank you!

22. I learned a great deal in the course and really enjoyed it. For the IDB project, it would have been nice if you could have provided good resources to check out. I spent a lot of my time spinning my wheels and getting caught on small things. I was somewhat disappointed we did not learn more about general software engineering philosophies. We didn't cover refactoring that much. I'm sure the book on refactoring is a good resource, but I didn't have the time to read it. So it would have been nice to cover software engineering principles more in depth. We covered testing very well though. I liked the Collatz project a lot and the number of tools we were exposed to in this class was fantastic. Overall a great course taught by a great prof!

23. 1. The lectures on Python, JavaScript, Refactoring, and SQL were generally engaging and good. The quizzes were intense because of the time limit. 2. autopep8, Canvas, coverage, git, GitLab, and HackerRank were easy to use. Understanding mypy, pydoc, and pylint working was harder. 3. Collatz was okay. 4. The group project was a new and great experience, even though there wasn't enough time to learn all the tools and working with all the new tools was overwhelming at first. 5. I found what the speakers had to say about the outside world to be interesting.

24. Prof. Downing always showed a great deal of interest in teaching the course and in us students taking something from it for the rest of our careers. He knows the content awfully well, so much so that he can answer almost any question thrown at him at any time. The content he covers in class, although somewhat advanced, is of great interest for those of us who want to become software engineers. The projects can be a great hassle since they are not tied to what is seen in class, but once again, they can be a great learning experience. I would totally recommend Prof. Downing and this class to anyone serious about learning more about real world programming and software engineering. Thanks!

25. The project is seriously densely packed with requirements. If I didn't have experience already working on web development the experience would've been wayyy different. It was already an extremely large workload and we definitely didn't need to spend as much time learning stuff and figuring out tricky new bugs as other groups. My suggestion would be to cut some of the requirements such as rich filtering/searching/sorting (maybe just 1 of each) or requiring embedded images. Also, I felt it was a little unfair to change the 4 quiz drop rule 3 quarters of the way through the course, though I understand the logic behind the change. Otherwise great class.