

*** PROVISIONAL REPORT ***

UNIVERSITY OF TEXAS AT AUSTIN
Downing, Glenn P C S373 51215
E100 EXPANDED

COURSE-INSTRUCTOR SURVEY
SOFTWARE ENGINEERING

Spring 2016 DEPARTMENT COPY
Enrollment = 57
Surveys Returned = 53

	NUMBER CHOOSING EACH RESPONSE					NO. REPLIES THIS ITEM	AVG.
	Str Disag	Disagree	Neutral	Agree	Str Agree		
1 COURSE OBJECTIVES DEFINED-EXPLAINED	1	1	2	20	29	53	4.4
2 INSTRUCTOR PREPARED	1	0	0	9	43	53	4.8
3 COMMUNICATED INFORMATION EFFECTIVELY	1	1	3	15	32	52	4.5
4 STUDENTS ENCOURAGED-ACTIVE ROLE	0	0	2	15	36	53	4.6
5 INSTRUCTOR AVAILABILITY	0	1	2	19	30	52	4.5
6 COURSE WELL-ORGANIZED	1	1	4	18	29	53	4.4
7 STUDENT FREEDOM OF EXPRESSION	0	0	3	18	32	53	4.5
8 CLASS PARTICIPATION ENCOURAGED	0	0	1	6	46	53	4.8
9 ENGAGING INSTRUCTION	1	2	5	17	28	53	4.3
10 INST. HAD THOROUGH KNOWLEDGE OF SUBJECT	1	1	0	19	32	53	4.5
11 INSTRUCTOR EXPLANATIONS CLEAR	0	0	4	17	32	53	4.5
12 GENUINELY INTERESTED IN TEACHING COURSE	0	0	0	9	44	53	4.8
13 HELPFUL COURSE MATERIALS	0	4	10	22	17	53	4.0
14 ADEQUATE INSTRUCTIONS FOR ASSIGNMENTS	3	0	4	26	20	53	4.1
15 ASSIGNMENTS AND TESTS RETURNED PROMPTLY	1	0	1	23	28	53	4.5
16 ASSIGNMENTS USUALLY WORTHWHILE	2	0	2	17	32	53	4.5
17 STUDENT PERFORMANCE EVALUATED FAIRLY	0	1	8	23	21	53	4.2
18 STUDENT PERCEPTION OF AMOUNT LEARNED	0	2	2	21	26	51	4.4
	Vry Unsat	Unsat	Satisfact	Very Good	Excellent		
19 OVERALL INSTRUCTOR RATING	0	2	1	13	37	53	4.6
20 OVERALL COURSE RATING	0	2	6	15	30	53	4.4
	Excessive	High	Right	Light	Insuff		
21 STUDENT RATING OF COURSE WORKLOAD	4	21	28	0	0	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	6	25	20	53	
	<u>A</u>	<u>B</u>	<u>C</u>	<u>D</u>	<u>F</u>		
23 PROBABLE COURSE GRADE	12	31	9	1	0	53	

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: 42

-
1. Professor Downing is a master of his craft, I hope he sticks around. All I would say is get rid of rackspace for the project it's a nightmare to set up. Consider using heroku or Amazon web services heroku first . It's free and is as easy to deploy as typing git push heroku master.
-
2. I learned a great deal in this class, and I'm happy to have taken it before graduating! It was nice to have a course that acknowledged real world skills like version control, teamwork, and learning new frameworks on demand. Your practice of asking everyone a question is great and kept people on their toes I found myself following along, trying to understand what you were showing us before hearing the explanation. The 20 off for making a small mistake is pretty heavy though. I know myself and a few others got 20 off for making a mistake fixed with 1 line. I also think a little more focus placed on language agnostic skills such as design patterns would be appreciated, I loved learning Python but perhaps cut that section a bit shorter.
-
3. I think that we should be given more time on quizzes. 3 minutes is just truly not enough time, especially given that sometimes when we reviewed answers, we would take more than 3 minutes to go over just one question. Also it would be helpful in the future if you would go over your project requirements before posting them. It seems to an extent you just copy and paste the requirements and rubric and there is often confusion when that happens. There is also a lot of vagueness in the project description so it makes it hard to truly know what we are supposed to be actually doing without having to post a hundred times to Piazza asking somewhat "dumb" questions. Other than that, I thoroughly enjoyed this class, and the IDB projects. Thank you!
-
4. The tools were good because CI and version control are necessities. The speakers were good as well but repetitive if a student has taken your OOP class. My only suggestion is to ask you to teach SQL and Javascript before P3 so that students who do not have prior experience with databases and JS get a little bit of knowledge before having to use them.
-
5. I had enjoyed the class and learned a lot of things that I didn't know beforehand. I liked that the class was done in python since I never knew the language beforehand and it was nice that not knowing the language didn't hinder me during the class. I especially liked learning about the topics towards the end of the semester when we focused on different refactoring techniques. I wish that we learned more about these topics and perhaps had an assignment over it that way we had more practice for it. I also liked the last three projects where we had to work with a group to create a web application. Since our group had a lot of problems with Docker, it would be nice if deploying our web application on Docker was optional.
-
6. Docker was a pain to use in the beginning. Even though I wasn't the one in charge of using Docker, I felt like it gave unnecessary problems. AngularJS was somewhat easy to use. There were a lot of tutorials online that helped me in the process of making the website. I think it would be a good idea for the TAs to give some information about each web tool. I kind of felt overwhelmed having to learn a new language on the fly when the deadline was coming up soon.
-
7. I really enjoyed the course, and had a lot more fun doing the projects vs the projects in OOP. I learned a ton in both classes. That being said, I learned a lot in OOP primarily from what you taught in lecture, but in SWE, I learned from my peers and group members about the web dev tools . I like the idea of having TAs come and give lectures about the web stuff. And in those lectures, they wouldn't necessarily have to teach us how to use the tools, but a "best practices" lecture would be good too. I'm definitely putting IDB on my resume, but I'm a bit worried that a software engineer at a company I'm interviewing for will look at the project code and think "They're using AngularJS completely wrong!", or something like that.
-
8. I really learned a lot from the final project, though I think it would be better to have smaller groups, maybe 3 people rather than 5. I really enjoyed using React, and I hadn't used it before. SQLAlchemy was pretty simple and straightforward. MySQLwas fine. However, it seemed like there were more helpful libraries for the search feature that only worked with PostreSQL. We couldn't find any good ones for MySQL so we just ended up using SQLAlchemy queries. Docker Carina was honestly pretty bad.Had it been bug-free, it would have been great. But we spent way too much time dealing with Docker issues. I would have liked to do more of the refactoring lectures than Python lectures, but I overall really enjoyed the class!
-
9. Tools- probably my favorite thing the first project was a good introduction to the tools, but a bit more instruction on the tools would have gone a long way. Non group prj- Collatz was alright, but I think instead of netflix it would have been nice to have a project more related to IDB. I think this would have let us produce even better quality work by the time we turned in IDB v3, and would give us more exposure to some of the cool web frameworks and tools. Group prj- this was awesome. Learned a great deal thanks to teammates with a wide variety of skills. Again, some instruction wouldve been useful. Speakers- excellent, wish more courses had these. Overall, I wish that there was more focus in class on the material in the projects.
-
10. I feel that much of the class talks about the how and what of implementations. How does python implement list comprehensions, generators, etc. I would like the class to at least touch on some of the Why's of these decisions, and what alternatives there are. We talk about many of the design options when we try implementing things in different ways, but we don't compare pros and cons of the implementation much, or about what would make you choose one or the other. In addition, I feel the presentation for the website should have a section for design choices that were made, so that students can compare solutions and see other students reasoning. Overall, a great class, a little slow at times, but very worthwhile and rewarding.
-
11. Order of topics can be improved incorporate required readings into class lectures discuss topics related to IDB such as APIs,front end frameworks,back end frameworks thoroughly check project specification page to prevent confusion in what's due discuss best programming practices and expect us to implement for all projects rather than do one part of agile at the very end
-
12. Fantastic class. My favorite at UT. I wish I would have had a smaller workload so I could have focused more on this class. This was by far the most practical class I've taken to date. I'm not a fan of plan-it poker. I think it's too simplistic to be taken seriously. Would have been nice if we were required to use it for the first phase. Docker is a beast. My team struggled with it until we got docker-compose running. Definitely make sure to emphasize that students should use compose instead of just docker. It made my life a lot easier. Maybe give Asana a try for next semester? It's a glorified todo list with slack integrations, assignable stories and epics. I've used it on several different professional projects and love it.
-
13. I do agree with making us use a scrum agile process from the beginning of Project 3. I think it would be helpful to introduce the tools we'll be using with a brief explanation of how they interact with each other to get us started. I had never used most of these tools before so I struggled with the learning curve. I think this class was fun and hands-on, but I wish the projects were worth more and the quizzes worth less.

-
14. I wish he gave more lectures like the last one he gave before our final project presentations. He gave a lot of advice and helpful pointers in regards to the industry, and it was extremely enlightening. The workload for the last few assignments weresomewhat higher than expected but it was a very rewarding experience. Overall, I would totally recommend this class.
-
15. SQLAlchemy was way more confusing to me than simply using SQL statements and interacting with the database. Other than that, the tools were useful, and I learned a lot of new things about a lot of different subjects!
-
16. the tools Learning Github is a great skill to enforce, as source control is really important. Same goes for Unit tests. I felt like the requirement of using Coverage wasn't justified I don't think it was explained well enough how important code coverage is in Python as opposed to other languages due to lack of compile-time checks the non-group projects It wouldn't be a Downing course without Collatz! Netflix was a really engaging and fun experience the group projects The use of a wide variety tools although was a pain, I feel was justified. the speakers I usually enjoy the speakers, but feel like the use is lost if the presentation isn't a dialogue. Just presenting on the company itself isn't that fun
-
17. I felt that the course projects focused a lot on tools and languages we have no experience on so maybe giving some insight to how some of the languages such as CSS and Javascript work would be extremely meaningful to the project experience.
-
18. I think it would be beneficial to have groups learn use planning poker and agile at IDB1
-
19. I took a course from this instructor before, I will confirm that I had a great experience in both of the classes and learned a lot. The structure of this class was similar to the other class, but it was more reasonable this time. If you can, keep upthe new extra quiz and quiz drop rule for future students. It will help them out a lot on those days where they need to attend interviews or take care of unexpected events. I pretty much only have good things to say about this class. I have learned a lot from just those two classes and I know it will be useful for me in the future. Thank you for another wonderful class.
-
20. This course left me feeling misled. We began with nothing but pure Python projects and lectures. Then suddenly we were forced to use a dozen completely new tools and languages. I understand the point of the project as a job simulation, but I feel the execution was poor. There was not enough support. Some students shined, while others dimmed out. I was in the latter group. This project left me feeling demoralized, defeated, and even useless. I spent hours upon hours trying to learn the material required, with little success. It was simply too much, too fast. I feel this course unfortunately sacrificed the education of some in exchange for the experience of others. I respect Downing as a great lecturer, but this course was ill-formed.
-
21. I very much enjoyed this course. However, the projects toward the end were poorly explained and help was not offered. I understand that a future manager at a company will want you to figure things out on your own, but it's discouraging when you're stuck on something and neither the instructor nor the TAs can help you because they have no idea how to use that specific technology. A few lectures on the technologies we used for the projects would have been helpful, and interesting.
-
22. While learning Python in class was interesting and an asset to overall learning, having weekly practice problems would help greatly in reinforcing our learning of the language and preparing for tests. The projects take up a good amount of time, so maybe optional problems for us to practice things like comprehensions, iters, etc. The problems we do in class are helpful, but I wish we had more to practice with, especially because the projects didn't really utilize complicated problem solving. Our group used React and liked the option. Knowing the entire scope of the IDB project would have helped in picking tools and planning our project, such as MySQL vs PostgreSQL, which has search functionality. TA lectures on web dev could be helpful
-
23. I found this course very engaging and I have learned a lot from this course with many things. Many of the tools that we used were fine such as angularJS and what not. But I found that using Docker was difficult and would suggest moving to some otherhosting like Digital Ocean. As well although I had experience already with web development, I think future students would greatly benefit from getting some lectures by TA's on angular react or simple server things just to get the project started and let the student still stumble a bit with the more difficult portions of the project. I found just by knowing a little bit helped me gain confidence using the new frameworks and helped me understand on my own harder materials.
-
24. I enjoyed this class overall. A few negatives, I dislike the revelation of rankings to students in the class, when you're in the bottom half it doesn't really do me much to know that it only serves to bring me down in my opinion. I also disliked Carina and I feel like web dev should be a prerequisite to take this class. I feel like I wasn't nearly as involved in the project as everyone else because I was starting from absolutely no knowledge of web dev while everyone else knew it well, thus it didn't make sense to give me the difficult tasks to do because it would take me 3 times as long as everyone else to complete them. However, learning python in this class was excellent really feel like that will be of value. Learning github is of value
-
25. Unbelievable disrespect for students. Wasted 20 hours optimizing for Sphere tests that would never have passed with only your instruction. Wasted twice that beta testing for a grade. You interrupt people more than anyone I have ever seen. While true question student- ans.- Interrupt student mid-sentence. Attendance points quizzes are the inexcusable resort of instructors unable to provide enough value that students want to come to class. ZERO feedback on exams?! Sexism has no place in a CS course. Corporatism has no place in public education. Stick to subject matter. Stop selling students to recruiters. Live is better outside. Pedantic coding is for compilers, not people. We pay you, not the other way around. Remember who works for whom
-
26. It was a pleasure to take your class. Your teaching style is extremely effective in enhancing understanding of all the topics. The readings were all high quality and even helped me during interviews. There are a lot of intangibles that your class really does help to instill, some of my favorite are as follows. Punctuality. As much as I despised having to be in class a few minutes early to take those quizzes, punctuality is a crucial habit to develop in the workforce. Rhetoric. Between in class and the readings you helped to embed a rhetoric to speak about the topic that really help to increase our ability to convey our thoughts effectively. Just try. By forcing use to work through in class examples, I felt more empowered and learned more.TY
-
27. It has been a great experience taking your class this semester, Professor Downing. I feel like the class should be more focused on the projects. That's what I enjoyed most about the class actually applying tools in Software Engineering. However, I felt as if the class was almost split in half. Half of the time we're focusing on projects, and the other half we're learning extremely technical aspects that we don't really ever have a chance to utilize. I feel like it would be great to have the class teach the basics of more of the tools that we use for the projects. I understand that's a lot of material to cover, but I felt I gained the most from the class from making the projects. I felt like the quiz system was weighted a bit too high too
-
28. I feel using docker added a lot of headache and I think google cloud would be a better option. Also I think on the javascript front Jquery is still a great library that is widely used in industry. Jquery also has tons of libraries built off it and the twitter bootstrap is even built upon it so I think seeing that used in this class would be beneficial. I also think it would be

alot more interesting if we had only one non group project and then allowed another group project where teams are allowed to make their own enhancements sort of a playground time for ideas to build out cool new features.

29. -Canvas wasn't a good way to give quizzes. Slow to load quiz and each page, with so many people accessing at once. Would have preferred clicker questions or paper quizzes. And would it be possible to do quizzes after five minutes of class? -JavaScript book wasn't very helpful, but I like the Extreme Programming book a lot. -Loved the projects! It's great to be able to use tools that we find online instead of being restricted like in most classes. -Found the lectures very interesting and informative. It might be useful to have a python environment open even through something like repl.it to test out student questions during class.

30. I very much enjoyed taking this course. The most important thing that I found helpful was working on the final project in the class. It exposed me to all these different tools that provides the foundation for a fully functional website and It also taught me how to learn about these tools by going through their documentations and outside resources. On another note, I wish you had given us an option to take the tests on paper rather than on our laptops. I would have preferred this. Also, It mightbe helpful if you could provide a couple of sample questions showing the formatting of the test at least for the first test. This might help people who haven't taken any of your classes before.An additional minute on the quizzes would also be great.

31. While I enjoyed the lectures, it felt like they were a bit disconnected from the group project material. Perhaps some lectures about a few more of the specifically required tools would be nice. For example, JavaScript is very foreign and confusing to me, although I did end up learning some while working. I know the point of the class is to model a sort of disconnected project manager who expects us to use tools that he doesn't himself understand, but perhaps a few of the lecture topics could be swapped out for more tool explanations. Anyways it was a great course, very happy to have taken it.

32. Professor Downing, this is the second course I've taken from you, and I've learned a lot from both of them. Keep on kicking ass. There aren't enough characters here for my full comments, so I'll send them via another channel.

33. Calling randomly on students to answer questions definitely forced me to pay attention throughout every lecture. There were questions that I had the answers to, as well as ones I didn't. This made me feel that Professor Downing should allow encourage answering voluntarily every now and then. Getting called on when I felt kind of lost made me very nervous however, it DID result in me having a tighter grasp on the material if I did not understand it before.

34. I have a few criticisms about the projects. 1. You, or at least the TAs, should give all the major tools a test-run a few weeks before projects are due. That might help avoid the Carina situation we had or other weird things happening, and would make it easier to help students who can't get the hosting platform to work. 2. You probably have, or could make, an automated system that checks for basic project problems, like invalid project.json, after submission. It would be nice if this ran before a project was due so students could correct simple mistakes early. My internship company had scripts that automatically check commits for simple configuration errors So I don't think taking off points for configuration errors is very educational.

35. My chief work in the projects dealt with flask, sqlalchemy, and angularJS. I had little issues with these, though I know that members of my group became quite stressed out working with Docker. The section I found most intimidating about this classwere the exams. However, the class does provide multiple opportunities to practice writing code in class, so in reality they are nothing out of the ordinary.

36. Great class. Throws you into the deep end of the software engineering pool and forces you to sink or swim. Class starts off very similar to OOP except in Python, which worried me at first, but after the first test the class diverged into new material. There are such a wide variety of topics that its difficult to learn them all in detail. I enjoyed the group project, but I wish that it had been more relevant to the in class material, especially when it relates to the XML and SQL topics. Overall this class was another great Downing practical skills class - which in my opinion, the university needs to provide more of these types of classes in the Computer Science department.

37. I have received an A in literally every other CS course that I have taken thus far except in Downings OOP and probably SWE, but I don't mind it a bit because I'm also learning a lot more!

38. The projects, though useful for the knowledge of the tools we used, felt very disconnected from the course material taught in class. I would have been nice to spend much less time learning about python at the first half of the class since most things that were taught could be learned by referencing a guide or manual , and instead diverted some of that time to learning about some tools, even at a very high level Lightning-talk style would even be great . Just the lectures over Python felt likea waste for me personally since I already know python and can read the online guide for what I don't know how to do.

39. This is a great class. Although Carina was a great. I think it would be better to make them create a website using there own knowledge and tools they have gather over computer science and googling. It would probably give a lot of flexibility on creativity and not one project would look the same.Give them some tools they could use but not limit them to only those tools. For example if they found a better tool to render pages and routing then Angular. Over all this class was great though. Workingwith new people I really enjoyed it.

40. Overall the class was a very good introduction to web development. One thing to note in future classes is maybe not using tools that aren't completely fleshed out. Docker to be specific. It would also help a lot if the lectures on XML and JSON were replaced with intro to JavaScript stuff. More design patterns would also be nice because I feel like the average CS student underappreciates the utility provided by them and spending more time on it would definitely help deal with this.

41. This has been one of my favorite classes in UTCS. The lecture style is very engaging forcefully so , and Dr. Downing is excellent at forcing you to understand each individual line of code. Some potential improvements 1. I don't think the projects do enough to force us to use the concepts we learn in class. I.e. the cache on Collatz and the unit tests on IDB are great places for decorators, but it feels like no one is enforcing any kind of guidelines on the code we write, which encourages shoddy practices. 2. Consider having much smaller assignments in the off-weeks, much more like problem sets, and similar to the test questions. Would be useful for studying and for boosting grades a bit. Thanks for a great class!

42. I like the idea of containers, but Docker Carina was a pretty big pain. I also feel like the second exam was a little too recall-heavy. I'm not sure how it's going to be graded, but I think in a SWE class, remembering the exact interface to a function is a silly thing to test. Every engineer googles that stuff.