*** PROVISIONAL REPORT ***

UNIVERSITY OF TEXAS AT AUSTIN   COURSE-INSTRUCTOR SURVEY   Spring 2016 DEPARTMENT COPY
Downing, Glenn P   C S373   51220   SOFTWARE ENGINEERING   Enrollment = 56
E100 EXPANDED   Surveys Returned = 51

	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	28	51	4.5
2 INSTRUCTOR PREPARED	0	0	2	7	42	51	4.8
3 COMMUNICATED INFORMATION EFFECTIVELY	0	1	7	12	31	51	4.4
4 STUDENTS ENCOURAGED-ACTIVE ROLE	0	2	2	12	35	51	4.6
5 INSTRUCTOR AVAILABILITY	0	0	10	15	26	51	4.3
6 COURSE WELL-ORGANIZED	0	2	3	13	33	51	4.5
7 STUDENT FREEDOM OF EXPRESSION	0	3	3	14	31	51	4.4
8 CLASS PARTICIPATION ENCOURAGED	0	1	1	9	39	50	4.7
9 ENGAGING INSTRUCTION	1	2	4	11	33	51	4.4
10 INST. HAD THOROUGH KNOWLEDGE OF SUBJECT	0	0	4	13	34	51	4.6
11 INSTRUCTOR EXPLANATIONS CLEAR	0	1	2	15	33	51	4.6
12 GENUINELY INTERESTED IN TEACHING COURSE	0	0	1	7	43	51	4.8
13 HELPFUL COURSE MATERIALS	3	6	8	15	19	51	3.8
14 ADEQUATE INSTRUCTIONS FOR ASSIGNMENTS	0	7	8	17	19	51	3.9
15 ASSIGNMENTS AND TESTS RETURNED PROMPTLY	0	1	5	19	26	51	4.4
16 ASSIGNMENTS USUALLY WORTHWHILE	1	1	7	13	29	51	4.3
17 STUDENT PERFORMANCE EVALUATED FAIRLY	0	6	9	20	16	51	3.9
18 STUDENT PERCEPTION OF AMOUNT LEARNED	1	0	4	16	30	51	4.5
	Vry Unsat	Unsat	Satisfact	Very Good	Excellent		
19 OVERALL INSTRUCTOR RATING	0	0	4	13	34	51	4.6
20 OVERALL COURSE RATING	1	0	10	19	20	50	4.1
	Excessive	High	Right	Light	Insuff		
21 STUDENT RATING OF COURSE WORKLOAD	4	30	17	0	0	51	
	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	10	25	13	50	
	__A__	__B__	__C__	__D__	__F		
23 PROBABLE COURSE GRADE	9	26	14	1	0	50	

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: 44

1. It isn't directly stated on the project description that we had to use docker-compose and stuff like that for the projects but with carina, it seemed like we HAD to copy it. I learned docker-compose and carina out of necessity not because I cared. If you want use to use the tools, that's fine but there wasn't adequate explanation on it. 3 lectures on what docker is and why we should use it would be infinitely more useful than learning about the various sql joins. The design pattern stuff is a bit lackluster in my opinion. Batory's class on it goes more into detail and is really what I think you should be aiming for. Other than that, I enjoyed the course and still think you're the best profs in the department. Thanks!

2. One complaint I would like to make is about the quizzes. It was said that the quizzes are used to promote class attendance. However, if you happen to miss all the questions on the quiz, then it would be like you didn't attend class. I was told by a TA that if your name was called in class and you're not there more than 5 times, your grade drops a letter. I think if the main purpose of the quiz is to promote attendance, then instead of hurting our grades, I think it would be better to reward those who come to class instead. Having two ways of hurting our grade for attendance is a little harsh in my opinion. Maybe if as long as the student takes the quiz, they will be rewarded at least one point for the quiz or something along those lines.

3. Netflix I did not think the Netflix project was helpful. The Project was honestly just who could move numbers around best. I didn't actually learn a useful way to predict movie ratings. I just learned how to shift some numbers around to get the tests to pass. IDB I think it's strange that we are required to use tools when the instructor doesn't know anything about them. I know he says the reason is for real world, but the reason we are in school is to learn. I was in charge of implementing the front end. I am unsure if how I implemented the front end is actually the correct way, it works, but I don't think it is the correct way it should be done. So it would be nice to have someone on the teach crew that knew how to use these tools.

4. Wish your classes had slides as opposed to just looking into raw code. It would be nice getting some more old and or practice exams. The daily quizzes were awful. Aside from the code questions there were relevant, everything else was just a pain and didnt add much value to my learning at all. I learned a ton in your class but just had a hard time being engaged.

5. I learned more in this class than any I have taken previously. Although a lot of the tools were new they were explained pretty well and I am now familiar with most of them. I enjoyed the individual projects and found them easier to get a good grade in, but I feel I learned more from the group assignments. Having the speakers in was great, it would have been nice if there were a few more.

6. I found the course very engaging and challenging. I liked how the professor encouraged us to be engaged in class by calling our names to walk through various tasks. This is something I have never had in a class and found it encouraged attendance andparticipation and kept me engaged through every class. The professor is very friendly and approachable and is extremely detailed in his explanations. Overall I think the course definitely pushes you quite hard which is a good thing if you're able tocommit to the workload. Although I did find myself giving significantly more time to this class than any other class, which may just reflect my skill level. I don't really have any criticisms of the class, I found it well rounded and interesting.

7. My only criticisms for the course are Rackspace and the course readings. As you know, Carina was a pain to use. I suggest trying AWS for the next semester, everyone can get a free 12 month trial but you have to have a credit card. I was very disappointed that the assigned course readings were never really discussed in class. I expected this class to focus more on the software engineering process, not programming languages.

8. I liked the class, the projects, and everything I've learned. It would have liked it better if more resources were provided for docker and AngularJS, but I guess that is part of the learning process. Other than that is was a great class.

9. I really hope that you can teach some JavaScript in the future just so students can have some basic knowledge on it to start the project. Learning a new language on our own is trivial, but being such a great teacher means you should talk about it atleast for a week. I also wish you can spend more time on "software engineering" as the first half of the course didn't feel like that as it was more geared toward learning Python. Lastly, I kinda hope you can teach another language in this course. Perhaps you can try new languages like Swift or Go, or a web related language like Ruby. Again, the best part of this course is you teaching. You should really use that as your advantage and teach students as many languages as possible. SayNoToDocker

10. There should be a few classes on the web tools being used for the projects.

11. Glenn is the man!

12. Dr. Downing has designed this class to be preparation for getting a job in the real world in ways outside of just building a website. The readings, the extra credit assignments of which my only complaint is that I didn't feel I got enough of a heads up about the blog, by the time I knew about it, it was too late . It is a fantastic course and I look forward to having him in OOP next semester.

13. I believe it is obvious that using corina and docker were mistakes. But I just want to make the statement to help suspend any doubt. The quizzes were a little unfair, I know you drop 5 but not being able to go back really sets us up for failure, plus one mistake and you fail the quiz. You almost take pride when you as a question on a quiz, and the next question reminds us the correct answer on the previous question. I really think either more easy quizzes or have the overall weight of them lower would really put less pressure on us and be an overall better experience.

14. Most of my work on the IDB project was to make our project work with Carina Docker. Although Docker was extremely difficult at first, I believe this tool is very valuable and powerful. The tutorials were extremely helpful in deploying our app, and once I discovered the docker-compose log command, it was easier to identify the issues with deployment. I believe the support was quick and useful through the Carina forums. Overall, I had a frustrating time with Carina, but I believe it is a very user friendly service once you get the hang of it. I felt like I had more issues with Carina than most, but I could be wrong, so many opinions might be different from mine.

15. This was a great course, however I think quizzes should be different. A little more time and being able to go back would have increased my quiz scored tremendously. The reason I say this is an issue is because it is worth a large part 21 of the grade for the class yet it is very pressured. Another aspect of the class that is an issue was Docker. It gave us many problems, yet

there was still penalties even when we had to host the project on another server where it worked perfectly fine.

----

16. Even though I wasn't really a fan of getting called on in class, I found it really helped me grasp the material we were covering better because I had to make sure I paid attention to what you were asking other students so that I wouldn't look like acomplete idiot once you did call on me. I also found the test to be a little on the tough side. That or the information given to us about what might to expect on the test might not have been enough. Overall, great class, wish I had had more time to spend on the material.

----

17. My only problem with the class was the required tools for the final project. At this point it's been made clear that Docker has it's problems, but I feel like we should have been free to use our own host of choice from the beginning - even outside of Rackspace. If the requirement was just to use a cloud host or unmanaged VPS I think even that would have been better. We would still have to how to configure a server   environment, but could at least use something more familiar or with a more active support forum. The rest was enjoyable and a worthwhile learning experience. The other projects were a great warm up. And the other tools are an integral part to development outside college so very beneficial to use and be familiar with too.

----

18. Probably had the most fun working with a group. Not the actual class though. Your lectures became nearly unbearable when we started talking about XML and beyond.

----

19. Making us use a Beta-feature of Rackspace's was a mistake. It's still too volatile and incomplete to be a useful tool to learn. Too much time was spent on python basics -- 2 days were spent saying type list   is list and type type list   is type, now let's repeat for sets, dicts, ints, etc -- , while not spending more time on frameworks such as SQLAlchemy and ReactJS AngularJS which are much more confusing and were required for the homeworks. Projects were more focused on following every rule to the letter rather than actually applying ourselves in a way that would be expected at any real company were autonomy is valued .

----

20. I really enjoyed the IDB project. It gave a real-life example of the development of a project is in a company-like environment. It gave examples of tools clashing against each other, learning languages and tools on the go, researching the best way to implement a feature, and team management as well as how to deal with slackers or lazy teammates. I was lucky enough to have an awesome team but others had a really tough time with their teammates. Although I would encourage more lectures regarding IDB and the tools. SQL lectures helped but AngularJS or React , Bootstrap, Gunicorn, Carina, and other tools were very obscure to most since a lot of students have not heard or learned about these tools and languages. Really enjoyed your class!

----

21. The quizzes were far too difficult and I felt like there was a lack of time and resources available for them. I did not learn as much from the final project as I hoped to.

----

22. Please don't get rid of docker. Yes it was very difficult to learn primarily due to issues associated with Carina , but it was a valuable learning experience. This could be mitigated by providing a class module specifically for containers and deployment. Instead of Carina a more stable container management platform would be ideal. Learning the design patterns earlier in the year would be great so that we could use them during the final project. Overall I really value everything I've learned in this course. Thank you for a wonderful semester!

----

23. Professor Downing has been the best lecturer I've had at UT. His methods of teaching and requiring active participation in class make the students pay attention and use good communication skills. I wasn't familiar with any of the web tools we used this semester so I can't say much besides I thought it was a great experience to be able to build a site from the ground up. I'm not sure I would like a lecture given by a TA since you're method has been truly effective for me. However, I think it would be great if we could have a few more web tool representatives give talks advice insight on their product. Thank you Downing for a great last semester of college and I hope this class can someday be apart of CS degree core requirements.

----

24. --Overall - felt lacking in terms of development theory, but good practically. Nervous about having my grade attached to the performance of others, but I suppose that's life, right? --Tools - Most of the tools were good in theory, but many were moretrouble than they were worth given the scope of the projects. Good tools included GitHub, GitHub Issues, TravisCI. I appreciated Docker. Apiary, RESTful APIs, a JS framework, SQLAlchemy, Flask, and MySQL are all worthwhile things to know, but it felt like I only had time to begin to learn a small subset. I strongly feel the last project was too large having a fixed a data source would have allowed focus on design for phase 1. Extension could be off found data in phase 2. Allow smaller teams.

----

25. I enjoyed the Python and refactoring lectures. I thought the lectures on database concepts - such as relational algebra and joins - were good, but we should have skipped learning SQL, since it wasn't used much in the project - SQLAlchemy did the work for us - and you can learn SQL in Data Management. There was too much disjunction between class and the final project - nearly every tool used needed to be learned on one's own, while I was expecting a lot more in-class guidance. Replacing the SQL lectures with an intro to these tools would be better, I think. This would also give students a better idea of what they would like to master further when contributing to the team project. Otherwise, please give this class an Independent Inquiry flag.

----

26. I see that you were trying to replicate industry in an academic setting, and for the most part I think you did a good job. For the most part I thought the topics covered were good. However I think you should cut out the SQL stuff. That time would bemuch better spent talking about HTTP requests headers. I also did not like how the test had short answers on it. That just encourages people to copy from their cheat sheet, not learn the material. Tests should be all coding. Finally I know a lot of people want lectures of Angular React. To accommodate this request I think you should let people do the code academy for Angular React and if they prove they did it, you will give them a 7 7 on a quiz.

----

27. Professor Downing was very knowledgeable and communicated the course material very efficiently. Software Engineering was the most useful class I've taken at UT in preparing me for the industry. The only complaint I have is that we had to use Docker and Carina. My team spent about half of the time trying to figure these tools out when we could have been developing. Also, I wish that Professor Downing covered some front-end skills in his lecture, such as JavaScript since it is in our readings .

----

28. -No more Docker. -Too many lectures on python, sql lectures felt rushed, design patterns were great. -Felt like no front-end techniques were covered in the lectures. Same with RESTful APIs. Please include. -Guest Speakers felt like a waste of time.The Rackspace one was nice because it helped us with Docker. -Some quizzes needed longer than 3 minutes to complete -Google Doc was a great reference to use to study.

----

29. The two non-group projects were helpful for introductory applications of Python. The IDB group projects were harder for people who did not have any experience with web development tools and frameworks. It would have been helpful if this had been communicated early on in the semester. It was certainly doable but knowing about it beforehand certainly would have made the process less stressful. Docker and Carina made us sad. I enjoyed the teaching style and the "forced" participation and quizzes -helps students keep up with the material. Test was fair. Sometimes it's hard to keep up with the readings. Not sure how much of

the readings others did. Overall, a very valuable experience.

---

30. Probably learned more in this course than any other CS course. I wouldn't recommend using the Docker and Carina pairing together, it was quite a pain to work through all their issues which retracted from my time actually working on our web application.

---

31. My biggest concern with this class was the 3-part project which took up the entire second half of the class. I feel like if that much of the class was about that particular project which is fair, as it is Software Engineering , we need to be taughtabout some of the tools we were supposed to use. I wanted to learn about Angular, Docker, and MySQL, and I had the opportunity to do that, however because of the distribution of experience of our group, I did not. I think if the scale of the projectwas smaller and the groups were smaller, we could be responsible for all parts of it, and actually learn how to make a good website from start to finish. Right now, I don't know how to do that at all.

---

32. RESTful APIs AngularJS   I worked frontend and angular was AWESOME! I had a hard time getting things up and running for project structure, but once that was done using angular was very intuitive. Flask I wish we were using Django since more of industry actually utilizes it. PostgreSQL My group used MySQL which worked well for us. I am not sure why you'd only allow one of those. Rackspace Docker   Docker really wasn't that bad. I mean if you just followed the same setup as the presenter gaveus everything should have worked. More importantly, if you had an issue or question and posted it to their forums everyone was EXTREMELY helpful and timely with their response. I just don't feel like some of the groups problem solved appropriately.

---

33. Apiary-Extremely useful when using the other group's API. Plus we need to know how to document APIs we create. Good skill to practice. Keep using...RESTful APIs-Keep requiring this because designing a good RESTful API is a valuable skill that every engineer needs some experience in...AngularJS or React-You should allow the choice of both. I used React for my group and I personally felt like it was much easier to get started with, as well as more applicable to the industry because you can drop it into existing codebases...SQLAlchemy-No opinion...Flask-Didn't use this for routing, but worked great for API backend. I'd add Node.js as an option too...MySQL or PostgreSQL-no opinion either way really...Rackspace and Docker-Docker keep, Carina no

---

34. I thought the class was great, the only problem for me is that I some times get lost when studying the code that we went through in class. There are no notes for what Professor Downing says about the code, only the examples that he places in the google doc. Maybe more comments in the code would help.

---

35. The way we did unit tests were different than the way we're used to with coverage and other method checking properties but the other tools I used in my internship except for using Jenkins instead of TravisCI. The non-group projects were fine and encouraged knowing people to get into a good group haha. However, the group projects were terrible because there were no sense of forced meetings. I mean we checked in on each other on Slack but we barely met which I hope would be a requirement instead of a suggestion. The speakers were great and gave great insight into the real world industry. Overall a great class and I'm sorry for not attending as much.

---

36. I really like this class and I've learned a lot from it, especially all the tools we used for the group project. I like the idea of working in groups because we get to solve problems together and learn things from each other. Although it was a little bit hard to keep everyone in the same boat because not all of us know the same things, but we learned how to communicate and work together better. Overall, I think it is a great class to take!

---

37. Use docker but don't use carina. It's good to learn docker but carina is unstable.

---

38. The projects and course material itself provided quite a general database of knowledge that a lot of students should know, and I learned the purpose of many tools, slack and git extra well. One of my issues with group projects which extends across ALL courses that exist in an educational setting is at least one of these things occurs   1 Someone overextends their amount of work, 2 Someone has trouble getting concrete work into the project, 3 More than one person conceptually doesn't understand the entire project, or parts of the project that are inherently useful. 3 is important in that it kind of sours the educational experience, and the first two require a group synergy that doesn't always occurs. Speakers were informative!!

---

39. I had no issues with the tools being used in the non group projects. Carina was obviously a pain. My vote is for AWS, it would certainly be helpful to learn it. Flask was perfect for beginners and Bootstrap is bootstrap.   The project was great andthe only problems I had was with Carina. Even using docker containers was simple enough and great to learn. Hosting and deploying it to Carina was the only problem.

---

40. 1.   I enjoyed learning git and GitHub because of how usesful and systemic it 2.   Both were good projects, but I think that our code should be more closely evaluated because we used globals as a last ditch effort and I felt I shouldve been penalized for it.   3.   Can't comment on React but we used the rest. I think that we should have more freedom choosing a front-end I've used vuejs which I absolutely love and makes angular look like it was made by a buncha monkeys. 4.No comment

---

41. I put probable grade as A, but I am being optimistic about the test! Anyways, Downing is a great teacher. I feel that OOP is an easier course, and the assignments are useful study materials for the test. In SWE, there is no correlation with the lastthree assignments and the tests, so it almost feels like two different classes. Also, Docker is evil, but somewhat necessary. I still think 20 off the project grade for forgetting a GIT LOG is kind of absurd since that should not have any impact ongrading the project. Most of the class mirrors industry, but that part is really really really bad. In class, we should spend more time on materials that can help with the big project since we really do not currently.

---

42. Excellent course. Lecturing style was very effective in teaching. Daily quizzes were alright, but would be better to provide a more time on quizzes that required some time to process a chunk of given information ex. SQL questions. Assigned article readings were interesting. Group Projects - Huge learning curve at the beginning, mostly because Docker Carina were difficult to learn and keep working. Flask and Bootstrap were easy to learn and use. Slack was really convenient and a great tool for collaboration. Would be nice to have some lectures on the tools for IDB or general web development. Speakers - gave good glimpse into industry, but maybe better to go in depth on a useful tool or good engineering practice.

---

43. Good class, sometimes the projects were a little too open ended. The tests were harder than in OOP, I think. The schema question on the exam being worth 55 220 points feels weird, though.

---

44. Many requirements for the projects were unclear. Didn't know the powerpoint and critique were due on presentation day. While I think Docker Especially , Carina, Flask, and SQLAlchemy were good learning experiences that will help me later in my career. I wasn't a total fan of SQLAlchemy though because of the way it models databases tables. I think bootstrap was a great way to get people started, but maybe it shouldn't be an exact requirement. I wish there was more discussion on the Netflix project and on the Docker and SQLAlchemy tools we'd be using in class. I think the speakers were great. I would make projects count waaay more

for grades. Great class but could do some tweaking.
