

*** PROVISIONAL REPORT ***

UNIVERSITY OF TEXAS AT AUSTIN
Downing, Glenn P C S373 88661
E100 EXPANDED

COURSE-INSTRUCTOR SURVEY
SOFTWARE ENGINEERING

Summer 2015 DEPARTMENT COPY
Enrollment = 63
Surveys Returned = 58

	NUMBER CHOOSING EACH RESPONSE					NO. REPLIES THIS ITEM	AVG.
	Str Disag	Disagree	Neutral	Agree	Str Agree		
1 COURSE OBJECTIVES DEFINED-EXPLAINED	1	0	0	16	41	58	4.7
2 INSTRUCTOR PREPARED	1	0	0	3	54	58	4.9
3 COMMUNICATED INFORMATION EFFECTIVELY	1	0	1	9	47	58	4.7
4 STUDENTS ENCOURAGED-ACTIVE ROLE	0	0	0	3	55	58	4.9
5 INSTRUCTOR AVAILABILITY	0	0	1	12	45	58	4.8
6 COURSE WELL-ORGANIZED	0	0	0	15	43	58	4.7
7 STUDENT FREEDOM OF EXPRESSION	0	0	4	5	49	58	4.8
8 CLASS PARTICIPATION ENCOURAGED	0	0	0	3	55	58	4.9
9 ENGAGING INSTRUCTION	0	0	0	4	54	58	4.9
10 INST. HAD THOROUGH KNOWLEDGE OF SUBJECT	0	0	2	6	50	58	4.8
11 INSTRUCTOR EXPLANATIONS CLEAR	0	0	0	11	47	58	4.8
12 GENUINELY INTERESTED IN TEACHING COURSE	0	0	0	5	53	58	4.9
13 HELPFUL COURSE MATERIALS	0	0	9	16	33	58	4.4
14 ADEQUATE INSTRUCTIONS FOR ASSIGNMENTS	0	2	5	22	29	58	4.3
15 ASSIGNMENTS AND TESTS RETURNED PROMPTLY	0	5	16	20	16	57	3.8
16 ASSIGNMENTS USUALLY WORTHWHILE	0	0	0	9	49	58	4.8
17 STUDENT PERFORMANCE EVALUATED FAIRLY	0	1	8	20	29	58	4.3
18 STUDENT PERCEPTION OF AMOUNT LEARNED	0	0	0	6	52	58	4.9
	Vry Unsat	Unsat	Satisfact	Very Good	Excellent		
19 OVERALL INSTRUCTOR RATING	0	0	0	10	48	58	4.8
20 OVERALL COURSE RATING	0	0	2	10	46	58	4.8
	Excessive	High	Right	Light	Insuff		
21 STUDENT RATING OF COURSE WORKLOAD	14	26	16	0	2	58	
	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	4	22	28	4	58	
	<u>A</u>	<u>B</u>	<u>C</u>	<u>D</u>	<u>F</u>		
23 PROBABLE COURSE GRADE	13	32	13	0	0	58	

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: 38

1. Very good teaching style

2. - Top Hat Glitchy. Super glitchy and I'm still bitter about it. - Slack Slack's integration with Github did absolutely nothing for us. We used it as a real-time messaging service. Eh. - Bootstrap I don't think that students should be able to use pre-made bootstrap templates themes. because it takes away from what we can learn. That's like letting us use pre-populated databases that we found online to pull data from. With 5 members in the group, one person can definitely make the front-end from scratch and they'll end up learning a tremendous amount. - Rackspace speaker was NECESSARY. Would've been lost without him. - Overall Downing sometimes makes you feel stupid but in the best way possible. Best trolllllll ever. Love it.

3. The assignments for the last project could have been graded a more quickly to allow setting up the website for the next due date however, this is an accelerated summer course, so grading is naturally slower. I think if Downing could ask the students at the end what played nicely and then recommend SETS of tools at the beginning of the big project would help a lot. For example, our group used SQLAlchemy for the database, and haystack for searching. We later found a GitHub post that said that haystack would be tricky to use with SQLAlchemy and we ended up reimplementing the database in the Django ORM anyway. If we had information from past semesters about what worked well with what, then we would have known to use Django from the start.

4. i really enjoyed this class, and can say with honesty that Downing is the most organized professor i've ever had. thoroughness was encouraged when it's not explicitly required. forcing us to use industry tools outside of our experience brought us closer to real world programming-- the practicality of the class has been second to none. i learned more about git in the last few weeks than in the last year, came to respect python, and made something interesting with peers. i truly cannot think of a negative thing to say about the class. downing forced me to eat my vegetables, and i am grateful for it.

5. pydoc-could never get it to work happily with Flask, recommend finding something else Travis CI-helpful as long as we updated .travis.yml Top Hat-cool, but mobile app has problems displaying different kinds of questions answers Collatz-a good introduction to the project style Netflix-fascinating problem to solve! wish I'd had more time to really sink my teeth into it Group-Bootstrap, Rackspace, and Slack were excellent and useful, while SQLAlchemy seemed finicky Rackspace-The best talk by a mile. The instructions were helpful and his advice was genuine and useful Atlassian-Not impressed. Seemed mostly sales pitch. MutualMobile-Decent, interesting topics. Best-seeing our site and realizing I understand how it all works Worst-pydoc Thanks!

6. No doubt that Dr Downing had explained materials crystal bloody clear for the materials. Python and Django had attached me after the class. My dev. exp had risen from nothing to something at least, and be able to learn on my own in the future. Git gives me confident to develop Github gives me social to share Unittest gives me desire needed to coverage the accuracy Pydoc gives me slightly briefly view on the each py? Travis CI helps to continuously integrate codes to build Top Hat gives me and Downing the ability of tidying quizzes Collatz teaches how Competitive Programming work See you on Friday on Fall Netflix teaches me how to teach my partner how Git branch work Apiary RESTful APIs teaches me how to API Rackspace is babe Thx

7. The course was probably one of the most useful as far learning, but I do believe the workload was quite high. Also, I thought it was very ridiculous how many insignificant things can "break the script" on an assignment causing us to automatically lose 2 letter grades on the assignment. I could spend 30 hours on a program and get 80 instantly all because I accidentally attached the wrong travis CI link to the Google form, for instance. If the class was worth more hours, or the workload was slightly less, and there was more leniency on the "breaking the script," then this class would be so much more worthwhile.

8. Saying I learned a lot in this class is an understatement. Despite the late nights spent on the group projects, I would not change a thing.

9. Great class. Really enjoyed the speakers because they talked about tools they used and what it was like to be a developer. Also liked how I could relate what I was learning to work. Not sure how I feel about top hat, but it was better than writing answers down on paper. Forcing me to learn Git is always a plus as well as learning the importance of coverage and unit tests. I didn't think Travis CI was necessary though. Collatz and Netflix projects were fine, just wished they better prepared us for the bigger project. Like maybe scratched the surface of databases or front end dev. Overall I learned a lot and now going through the interview process this is the class that I always talk about. So grateful to have taken this class.

10. The Tools They felt excessive at first but I got used to some of them, and others, not so much. The documentation and tests were sometimes harder to get working than the actual projects. Non Group Projects A good intro to Python The Group Projects Django was by far the best part of the class for me. I had wanted to learn it and the project forced me to learn Django, get better at using Bootstrap, and utilize a Postgres database that is hosted on Heroku. I will now write all my future web apps using the Django framework. I can't comment on the other tools because I did not touch them. Speakers It was ok, a nice change of pace, but I did not glean anything particularly useful from these talks. The speakers in OOP were better.

11. Loved the class. I enjoy how Dr. Downing calls on people and makes everyone participate. Consider using BitBucket? There are unlimited free private repos for students. I don't understand why the first test was a test on the Python language. I feel like this could be accomplished with the 1-hour python course that UT probably offers. I enjoyed all of the tools. Travis CI was kind of lame, but I'm not using a huge code base, so I guess it works better in a real SE environment. I felt like in the group project, everyone just kind of melted into the roles that they were most familiar with. I learned some new things from the project, but not a lot. I enjoyed projects 1 and 2. I don't care about the guest speakers.

12. I found this class to be extremely useful. I think what would of been beneficial to us was to learn or at least have more exposure to SQL and JS before beginning the 3-phase project. I also think the Rackspace demo done in class wasn't sufficient enough, and that we could of all participated in the act of starting up our own server when he was in the room. My final comment is that I thought the MutualMobile lecture was unnecessary.

13. Very much enjoyed the course, appreciate the patience and knowledge gained. The course projects were fun and team projects even more so. Thank you!

14. A good course, great lecturer. The lectures over python were great and valuable. The tools used in class are great, however I think since the instructor does not go into great depth of using the tools, the policy of a zero grade should not be in effect, particularly in the group project. The non group projects are solid and I think that they were a fair assessment. Yet with the

group projects containing an much more challenging task and the matter of working with a team , the grading structure should change. I do enjoy the method of calling on class members to participate. You feel more involved. The speakers should not have been allotted the whole class period to speak. With guest lecturers its can be very challenging to stay focused.

15. Great professor! learned a great deal of python and full stack development! Will definitely help me in my job search! Thanks again!

16. - Github, unittest, coverage, pydoc and Travis CI are great!. They are really useful tools for general programming. Top Hat is nice that we can review the quiz and see the grade right after we took it. The quizzes are challenging in the problem itself and timing constraint. They required us to think on our feet. - Collatz and Netflix are great warm-up using Python and the above tools. They prepare us for the group project. -The group projects tools help us learn different aspects of the web application. I think they reflected real project for jobs. Maybe in the future class, should there be an option to use tools from Atlassian? - I like how we have several speakers from the industry.

17. Fantastic course. I am very happy I accidently had to take it this summer right before my graduation. Just fantastic. The team environment, the necessity of taking initiative to learn the tools, and focused and progressive goal that the website provides, all of it blends for a very real software development experience. Of course this serves as a microcosm of actual development processes, it very much reflects my real world experience so far in the nature that nothing is provided for you, you have to work together and figure it out, you have to identify the tools, learn them, and use them to construct your product. As for tools Slack is awesome, love the emphasis on git, Travis CI is great as well. Top Hat was nice. SQLAlchemy was great.

18. In IDB3 ,I think it should have been disallowed for groups to make changes to their API the day the project is due. There should be hard deadline for this, and if the group must make changes then they should email a TA first or talk to the professor to assess whether the change would significantly affect the group exercising their API's implementation. Other than that, I really enjoyed this class,and I like that you called on us during class because that definitely kept me awake and paying attention. Thanks.

19. Admin Top Hat was finicky with quiz submissions, I have not witnessed a live interactive quiz system that works perfectly yet. But until there is one, I would strongly encourage that the quiz grade not be weighted so high Projects having to learnall the tools has been empowering THANK YOU Tests I was unprepared for the tests material tested on was a mystery right up until a few days before. Either put more emphasis on the material that will be covered on test, or make projects worth moreand tests less Most Important this class is awesome, and no material should be stripped from it and I know this was a summer session, however, there is way too much in it to be a 3 credit class. It should be a 4 or 6 credit, bare minimum a 4

20. I loved this class. Quizzes were really tough though.

21. I am a huge fan of Downing's classes. He's extremely knowledgeable, and very good at explaining ideas. In all 3 of the Downing classes I have taken including cs373 he has made me think about programming at a level that I have previously ignored. He has taught me to understand the intricacies and subtleties that determine the difference between an ordinary programmer and an excellent programmer. For this class I wish he took at least one lecture to talk about some of the tools relevant to our websites. We were required to use Javascript in our website, but we spent very little time talking about actual uses of javascript. Instead we talked about theoretical stuff in javascript. Overall I highly recommend this class and this instructor.

22. Things that made the course worthwhile was learning python. I never learned python before, and this course helped me learn to code in python effectively. I did enjoy how Dr. Downing kept the entire class on their toes by calling on individuals in class. This created a friendly learning environment and helped me notice I'm not the only one confused sometimes. Each project had a ton of information and seems overwhelming at first, but in the end, they helped me learn more about efficiently coding with Github and TravisCI. Only negatives I have from the course is when the CS Labs computers locked down for over 3 hours on the day the project was due and we were unable to reach Dr. Downing for a response. That isn't too big of a deal though.

23. Professor Downing's knowledge is impressive. The only thing I would change about this class is maybe adding a lecture on some techniques on how to debug the last project more efficiently.

24. This class had helped me learn how to use github more effectively. Tophat was not all that great, especially when we had to pay for it. Collatz and Netflix were good starting projects. The group projects was the best part about this class, it made us go out of our comfort zones in term of social aspects and ability to learn new tools in a short amount of time. The Guest speakers was helpful, my particular favorite was MutualMobile.

25. I liked the course. My problem with the course is the self-dubbed "draconian rules". The grade in the class is very unforgiving to issues that I would imagine to be common, such as unfamiliarity with a tool required for the assignment. This lends a huge advantage to students already familiar with these tools, widening the gap between students who are ahead of the curve and those behind it. I would like to see better resources for learning tools required for assignments, such as TA sessions designed to give basic introductions to tools like Github, or required TA familiarity with all of the tools required for the final project. I felt forced to learn a lot on my own, and lost points in the class for struggling with that.

26. I enjoyed using most of the tools. Some of the project requirements, especially for the last project, I felt were unhelpful. TravisCI in particular was just a box to tick on the IDB project, since the Django unittest suite doesn't work well outside Django. I was also a little unhappy with having to put the tests on our actual site. We had to rewrite some of them to be less useful because of the way the database gets loaded--basically if you have a test that executes a query on the database, and the test is called outside Django's test suite, it runs the query on the entire database instead of the test environment. Also, High Noon is a better movie than Shane.

27. I was hoping you would be teaching Drupal in this semester. Knowing Django is good too I guess, even though I'm not planning to work with it.

28. This is an amazing class to take if you're interested in any part of web development. It's also a great class if you really want to pick up Python. The first half of the class covers Python basics, and what makes Python what it is. For the second half you get put in teams of five to make a website from the ground up, including getting a blank web server from Rackspace and go from there. Other than requiring frontend to be Bootstrap with either AngularJS or React and backend to be Django or Flask, Downing gives a lot of freedom for how to go about doing it. Unfortunately that means he doesn't cover anything regarding the frameworks and you're left to either figure it out yourself, or post on Piazza and ask your classmates for help.

29. The tools- TopHat worked relatively well, especially with the answer submissions via laptop, phone app, or text,or on paper as a

last resort. Git and GitHub were great, but I would have liked a greater emphasis on feature branches and pull requests, maybe incorporating Git Flow. Collatz and Netflix were great projects, no real complaints. Possible changes for later semesters- implementing another group's API - possibly create an earlier deadline for the API or have all groups call the same API. Talk a bit about the importance of sticking to the docs, tutorials, and expected use cases for MVC frameworks like Django. Recommend spinning up a prod server and a dev server? Great, extremely useful course overall.

30. Some of the content of lectures in this class was unproductive and useless, and the quizzes concerning that content were gimmicky and unfair. It makes little difference if I know, for example, where to place an asterisk in order to unpack variables in python. That's the kind of thing you learn as you code and debug, not that have to memorize and be quizzed on immediately. Nor does it fall under the scope of what "software engineering" should mean. If I was interested in that kind of material I would lobby for there to be a class called "nuances of python and javascript." Much more class time should have been dedicated to teaching us how to set up and use the tools we needed for the projects. Good course, but that aspect was, frankly, bad.

31. Great class! The presenters were helpful. The projects were interesting and rewarding. Forcing us to use a bunch of tools has greatly improved my resume which I am thankful for. I like that you have a focus on the student's future by encouraging internships etc. Also, the webpage was a fantastic project. It has everything a real world problem would such as working with a team, having a deadline, learning new tools, and presenting. More professors need to force presentations because public speaking is not something we get to practice often. One downside is I felt that the javascript stuff wasn't that relevant and the in class lectures while working on the website didn't feel as engaging since I was more focused on the website

32. The overall course was excellent. I really learned a lot from the tools required for the class. If anything give a little more time per project but since it was in the summer it makes sense that everything was so crunched together.

33. This was probably the most useful upper level computer science class I've taken. Your forced class participation and group projects helped a lot. The tools were all great, but if you're going to use Top Hat again please try to request a classroom where there is cell service, i.e. not a basement. The speakers were all great.

34. The group project was a lot of work for inexperienced students. My group spent around 60 hours a week in the lab, sometimes staying up until 7am, just to meet the minimum requirements for each phase and not even earn full credit. The speakers were very good. They made me realize that I could probably do software engineering as a profession.

35. . the tools git, GitHub, unittest, coverage, pydoc, Travis CI, and Top Hat . the non-group projects Collatz and Netflix great for getting used to git and travis CI and make and what not. . the group projects the use of Apiary, RESTful APIs, Rackspace, Django, Flask, SQLAlchemy, Bootstrap, and Slack. More time...for sure... . the speakers Rackspace, Atlassian, and MutualMobile

36. I believe the last project was a very worthwhile experience. It was great to actually get an opportunity to apply what we learned. I feel like I learned more in this 9-week course than I have in any other class I have ever taken because we were forced to learn by doing, and to do it very quickly. I don't know if it was just because it was summer, but I think students would benefit from extending the IDB project to 5 phases. Ease them into the process the first phase or two, then add on more expectations. It was difficult to keep track of all the new expectations during each phase and there were things my group wanted to go back and redo, but didn't have the time. My time would have been better spent on that than on Netflix or Collatz.

37. Really could have used half lecture on version control git and half lecture on web frameworks. shot ourselves in foot w git many times. There's almost too much in the project specs, hard to tell what's important. Talking to TAs piazza helps though. Was cool learning about Travis, pydoc, unittest. Useful stuff I had never tried. Speakers were interesting, not too many few. Fav was Rackspace because we actually learned something needed for project. I would like to have learned from a pro how to iterate over angular html design, what tools, workflow, etc? Quizzes, ugh, would rather do on paper with all questions given at same time. No scary timer. probably 25 higher grade for me that way talking exercises in class great! lectures great!

38. . the tools git, GitHub, unittest, coverage, pydoc, Travis CI, and Top Hat TopHat was super cool to use because it kept the grades and attendance on the app. All tools were great to learn and I will be using them in the future for personal projec . the non-group projects Collatz and Netflix Projects were fair since they were used mainly to practice using the tools. Netflix project is doable solo. . the group projects the use of Apiary, RESTful APIs, Rackspace, Django, Flask, SQLAlchemy, Bootstrap, and Slack. Slack was incredibly helpful for team communication. Some problems with django and pydoc but it was resolved. . the speakers Rackspace, Atlassian, and MutualMobile Speakers were generally well spoken and provided nice insight