

*** PROVISIONAL REPORT ***

UNIVERSITY OF TEXAS AT AUSTIN
Downing, Glenn P C S373 52201
E100 EXPANDED

COURSE-INSTRUCTOR SURVEY
SOFTWARE ENGINEERING

Spring 2017 DEPARTMENT COPY
Enrollment = 57
Surveys Returned = 56

	NUMBER CHOOSING EACH RESPONSE					NO. REPLIES THIS ITEM	AVG.
	Str Disag	Disagree	Neutral	Agree	Str Agree		
1 COURSE OBJECTIVES DEFINED-EXPLAINED	1	1	3	16	35	56	4.5
2 INSTRUCTOR PREPARED	0	0	0	5	51	56	4.9
3 COMMUNICATED INFORMATION EFFECTIVELY	0	1	4	9	42	56	4.6
4 STUDENTS ENCOURAGED-ACTIVE ROLE	0	0	2	9	45	56	4.8
5 INSTRUCTOR AVAILABILITY	0	1	1	9	45	56	4.8
6 COURSE WELL-ORGANIZED	0	2	6	15	33	56	4.4
7 STUDENT FREEDOM OF EXPRESSION	0	0	1	11	44	56	4.8
8 CLASS PARTICIPATION ENCOURAGED	0	1	2	5	48	56	4.8
9 ENGAGING INSTRUCTION	2	0	3	11	40	56	4.6
10 INST. HAD THOROUGH KNOWLEDGE OF SUBJECT	0	0	0	14	42	56	4.8
11 INSTRUCTOR EXPLANATIONS CLEAR	0	0	1	16	39	56	4.7
12 GENUINELY INTERESTED IN TEACHING COURSE	0	0	0	3	53	56	4.9
13 HELPFUL COURSE MATERIALS	3	1	12	21	19	56	3.9
14 ADEQUATE INSTRUCTIONS FOR ASSIGNMENTS	1	4	10	20	21	56	4.0
15 ASSIGNMENTS AND TESTS RETURNED PROMPTLY	1	2	4	16	33	56	4.4
16 ASSIGNMENTS USUALLY WORTHWHILE	2	0	2	19	33	56	4.4
17 STUDENT PERFORMANCE EVALUATED FAIRLY	1	2	6	20	27	56	4.3
18 STUDENT PERCEPTION OF AMOUNT LEARNED	1	0	4	16	35	56	4.5
	Vry Unsat	Unsat	Satisfact	Very Good	Excellent		
19 OVERALL INSTRUCTOR RATING	0	0	5	7	44	56	4.7
20 OVERALL COURSE RATING	1	2	7	14	32	56	4.3
	Excessive	High	Right	Light	Insuff		
21 STUDENT RATING OF COURSE WORKLOAD	2	30	23	0	1	56	
	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	29	19	56	
	<u>A</u>	<u>B</u>	<u>C</u>	<u>D</u>	<u>F</u>		
23 PROBABLE COURSE GRADE	19	26	10	1	0	56	

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: 30

1. I feel that the Netflix project gave no value to the course and should be removed. It should be replaced by some project that trains people how to use flask react in a basic level before the big project.

2. I lost like half a letter grade on two questions on the midterm which colored my enjoyment of the class

3. Misleading course title. Expected more software engineering and less python.

4. Professor Downing is one of the most captivating professors at UT. He has one of the most resourceful, interactive, engaging experiences to offer with a UT upper division CS elective.

5. This was easily the most instrumental course in cementing my interest in computer science, as a work field. Before this class, I was unsure of how I could transition into the software engineering industry, and now, I feel much more confident and excited in my future endeavors. While some of the IDB instructions could have been clearer, it was one of the most enjoyable projects I have ever taken part in. On top of that, Professor Downing is easily one of the best lecturers in the department, and was able to properly engage the entire class each and every day. By far one of the best courses I've taken.

6. Coverage was awkward to set up for me, but once it was working it was very helpful. pylint is opinionated, but useful for maintaining code standards. Travis is easy to setup and use, used in industry as well. Collatz was a good starter project, but I think was too intensive because of HackerRank requirements. Netflix was an interesting project, and I like that we got to use a real world database. SQLAlchemy Worked well for us but also had a decent learning curve. I'd keep using it unless something simpler comes out. The Red Hat talk was bad, it went far too quickly and was not helpful. Consider not asking them to return, Atlassian Loved their talk, was funny and insightful

7. Downing is an incredible professor, perhaps the best I've had at UT. I cannot recommend this course enough to anyone willing to put in the effort to learn.

8. In regards to tools, the only complaints I have are the hassle of autopep8 and pylint. I would run autopep8, but it didn't do much to my code. Most of it I had to manually fix myself and running my code through an online pep8. With pylint, a lot of its complaints were irrelevant. For example, with SQLAlchemy code, I ended up muting most of pylint's complaints because it didn't apply for database work. The alone and group projects were good. There is a big jump, however, in workload from Projects 1-2 to IDB that I did not see coming. Finally, most of the talks were great. The only one I found less relevant was the GCP talk because I learned nothing about how to set up or use GCP. That talk was more about boasting about GCP and Google.

9. Professor Downing is a great lecturer and knows how to engage an audience, which is something that I haven't seen from other professors. I learned a great deal in this course, and each project was a valuable experience. I'm especially glad that I had an opportunity to learn and use tools that are being used in industry. As far as the grading, many of the quiz questions were borderline unreasonable, but the extra credit balances things out. While I found many of the papers interesting, about 70% of the assigned reading was frankly too much and there is no way anybody read all of it. The class as a whole is great, but the assigned readings need adjustment remove the JS book !

10. Be sure to warn students that Google Cloud Platform is not compatible with PostgreSQL, which put my team behind in the 2nd phase of the project.

11. The projects sometimes had ambiguous details or requirements that caused some confusion especially for IDB projects . Sometimes requirements were changed close to the deadline which could lead to frustration for some people, however, these changes were fairly minor and far in between so in the end it wasn't too big of a deal. The lectures were fantastic and definitely ended up teaching me a great deal about software engineering and Python Java. The tools we ended up learning were invaluable and will help a great deal in the real world. Can't really think of much to improve in this category.

12. I thoroughly enjoyed the class. The enthusiasm and knowledge displayed really made me get into it. Autopep8 and pylint directly helped me at work. The course overall taught me some great practices in coding that I needed to learn. In regards to changes or additions Personally, I feel that squashing commits on a master branch for the first two assignments where we submit pull requests would be a worthy addition to the course. Working in the industry, one cannot simply make a pull request with piles of commits muddling it up and its relatively easy to do with a bit of research aka git rebase -i . I am a senior and will be graduating soon, but I will never forget OOP nor SWE and will keep recommending them for years to come.

13. 1 I enjoyed using the tools in this course as I did not have a lot of experience with them previously. I can see now how they are utilized in the context of software engineering. 2 Collatz was a great warm-up to the rest of the projects as it allowed me to get familiar with some of the tools that we would need to use for the rest of the semester. I found the Netflix project to be very interesting. 3 The group projects allowed me to work on my teamwork skills and I had not built a website from scratch before so this was good experience for me. I got really good exposure to the tools that were incorporated in the projects so I am grateful for that. 4 The speakers gave me valuable insight into industry. I appreciate these opportunities.

14. I really really did not like the format of the exam. Before the test, I was confident in my knowledge of the material but felt that multiple choice questions cannot aptly test knowledge of Python.

15. I wish the exams did not have such a high percentage in a project based intensive class. The exam or class material are in no way related to the project assignments. We had to use Flask, React, AngularJS, python and html. We were only taught python in the course and I felt it was not even relevant to the project. While Flask, React, javascript, AngularJS and Html were barely touched at all. Nobody in my group knew React or Angular so we did the whole website in html with just a small part in it using React. I feel the class should be more about these topics and even the tests. The professor just expects the students to self-learn these concepts and work on the project which I could have done without the class .

16. This was a fantastic and challenging course. The major con was the exams. There were vague instructions on how the exams would be formatted and what exactly would be asked. Consequently, I bombed the first exam and will do my best on the second. I do fear that no matter how well or differently I study, that I will receive a poor grade due to the difficulty and specifics of the exam questions. Besides the exams, this course has been one of the most challenging and fulfilling courses I have ever taken. Prof.

Downing is an absolute joy in lecture and an extremely fair and understanding instructor. I am certain without a doubt that I will use the skills I learned in this course for the rest of my life. Thank you.

17. downing is like the dad i never had

18. Professor Downing has a very trenchant teaching style. He hits home on every concept, and clearly elucidates every topic in terms we can understand. With every college course I've taken, there is some intellectual gap between professor and student, and it is up to the student to close the gap. Professor Downing is the first professor I've had that does the work of closing the gap himself, often saving each of us hours of time Googling. He is a very smart man, but he explains concepts in terms of how we understand, not in terms of how he understands. I emulate his teaching style when I am teaching others how to code.

19. I really enjoyed how enthusiastic Professor Downing was about teaching the class. Also, I liked that he didn't only care about students doing well in the class but also in their career paths. He consistently informed us about internship and job opportunities and showed a legitimate interest in helping us to with our futures.

20. This has probably been the most useful class I have ever taken at UT, and I don't doubt it will stay that way for my last two semesters. The tools were all pretty useful, I really think learning how to use unit tests and Travis CI will be helpful, as I see a lot of companies looking to expand their self-testing capabilities. I wish I could have learned more about Docker. It seemed like a cool tool, but I had trouble figuring it out at first and then had to stop messing with it so I could finish the project in time instead. Netflix project was great and fun, Collatz project was a little dry, and seemed like a lot of people got around it using a full Meta-cache. The group project was fantastic, some of the talks were too late at night and long

21. GCP was a strange platform to work on. The pricing is confusing to understand and may be the reason why we burned the 100 of credit within the project time frame. Flask was an amazing package to work with. There is a huge community around flask that allowed us to find flask extensions for every problem that was thrown at us. While the projects were fun and great learning experiences, the class seemed a little broad. I couldn't really determine what I was to get out of the class other than learning some new languages. However, I really enjoyed the class and Prof. Downing and hope to take another class of Downing's. You don't seem excited? High point of the goddamn survey!

22. The planning poker activity requirement being included only on the last phase of the project, I felt was kind of pointless and not really that useful. It would have been more useful to include something like this as a requirement in the beginning Phase 1 of the project. This way we would have all become more accustomed to it and would have been able to use it better.

23. Tools - I thought the tools were great. Maybe dive a little deeper with the docker stuff though because there is a lot of power with understanding docker. Non-group projects - Also good. The way the online judge judged collatz only allowed for one implementation though. Group Projects - I personally loved the project and learned a lot. One thing to maybe consider for improving the experience is creating teams based upon grade level 2SR, 2Jr, 2Soph because if I was a sophomore trying to do that project I would've been overwhelmed with the material. Obviously that is not real life though so whichever way works. Maybe force pull requests for the repo as well. Speakers were excellent.

24. The way you explain the material and organize the course is very well done. I do not agree with your policy on missing quizzes. I believe that the zeros for either missing quizzes or earning zero points on quizzes should be its own punishment. I do not believe it is necessary to drop a letter grade for having 10 zeros in quizzes. I enjoyed your class very much and found the material extremely interesting and useful for any project. Thank you for teaching this class.

25. One thing that can be improved is teaching us about the technologies used to do the final 3 projects- we only covered Python in class, and it was difficult to do the assignments without learning it from somewhere.

26. Very worthwhile lessons, but many of the tools suggested or required for projects had very little or no explanation. After class office hours help with this but seems unfair to students who are not able to make it to those.

27. I think the content of this course could be modified. I think the refactoring has more to do with software engineering than some of the finer points of python, and I think it would have been interesting to discuss the refactoring strategies earlier, and possibly have us apply them to our own code we developed for the class. I did find the emphasis on tool usage to be interesting, and I learned quite a bit about what it means to use libraries other people had written. Previously, I had done pretty much everything myself, so having to deal with oddities and bugs in code which I really had no direct influence over was a good experience.

28. Too much minutiae of Python! Felt out of theme with the rest of the course's focus on exposure to tools

29. Great course, learned a ton. At times, too many tools were required, so the projects could get bogged down with trying to shoehorn in a particular required tool. But, I understand the premise that even a little exposure to the huge number of available "supplemental products" out there is a good thing.

30. I strongly feel that the material learned in this course was not helpful to me at all. I took this course with the hope that I would be exposed to the best web technologies and full-stack software practices and also gain a great deal of full stack web development experience. While the last three projects were clearly intended to full fill most of this, it would have been extremely worth while to have the class material revolve around these topics as well rather than learning python, xml, sql from scratch. It was awesome that we were required to use certain tools such as autopep8, travis CI, pydoc, etc. The speakers were a good touch to the course to expose us to real world tech companies.
