

*** PROVISIONAL REPORT ***

UNIVERSITY OF TEXAS AT AUSTIN
Downing, Glenn P C S371P 50885
E100 EXPANDED

COURSE-INSTRUCTOR SURVEY
OBJECT-ORIENTED PROGRAMMING

Fall 2015 DEPARTMENT COPY
Enrollment = 73
Surveys Returned = 65

	NUMBER CHOOSING EACH RESPONSE					NO. REPLIES THIS ITEM	AVG.
	Str Disag	Disagree	Neutral	Agree	Str Agree		
1 COURSE OBJECTIVES DEFINED-EXPLAINED	0	2	2	20	41	65	4.5
2 INSTRUCTOR PREPARED	0	0	0	9	56	65	4.9
3 COMMUNICATED INFORMATION EFFECTIVELY	0	0	4	19	42	65	4.6
4 STUDENTS ENCOURAGED-ACTIVE ROLE	0	0	5	20	40	65	4.5
5 INSTRUCTOR AVAILABILITY	1	1	5	21	37	65	4.4
6 COURSE WELL-ORGANIZED	0	2	1	16	46	65	4.6
7 STUDENT FREEDOM OF EXPRESSION	1	1	6	21	36	65	4.4
8 CLASS PARTICIPATION ENCOURAGED	0	2	1	5	57	65	4.8
9 ENGAGING INSTRUCTION	0	1	4	21	39	65	4.5
10 INST. HAD THOROUGH KNOWLEDGE OF SUBJECT	0	0	1	5	59	65	4.9
11 INSTRUCTOR EXPLANATIONS CLEAR	0	0	2	22	41	65	4.6
12 GENUINELY INTERESTED IN TEACHING COURSE	0	0	3	14	48	65	4.7
13 HELPFUL COURSE MATERIALS	2	2	13	17	31	65	4.1
14 ADEQUATE INSTRUCTIONS FOR ASSIGNMENTS	0	2	4	28	31	65	4.4
15 ASSIGNMENTS AND TESTS RETURNED PROMPTLY	0	2	11	34	18	65	4.0
16 ASSIGNMENTS USUALLY WORTHWHILE	2	1	3	19	40	65	4.4
17 STUDENT PERFORMANCE EVALUATED FAIRLY	0	3	14	23	25	65	4.1
18 STUDENT PERCEPTION OF AMOUNT LEARNED	0	1	4	18	41	64	4.5
	Vry Unsat	Unsat	Satisfact	Very Good	Excellent		
19 OVERALL INSTRUCTOR RATING	0	0	5	17	43	65	4.6
20 OVERALL COURSE RATING	0	2	11	17	35	65	4.3
	Excessive	High	Right	Light	Insuff		
21 STUDENT RATING OF COURSE WORKLOAD	4	30	30	1	0	65	
	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	17	21	25	65	
	<u>A</u>	<u>B</u>	<u>C</u>	<u>D</u>	<u>F</u>		
23 PROBABLE COURSE GRADE	14	26	22	1	2	65	

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: 40

-
1. This course was amazing! Downing really knew how to organize each assignment and lecture so that we got the most concise and worthwhile volume of knowledge. The projects were manageable and just challenging enough that I still learned what I needed to without spending 40 hours week in the lab and feeling overwhelmed. My only complaint is that the TA excessively cancelled his office hours, so we really never had a TA when we needed them. I would definitely take another course from Downing again.
-
2. Would prefer a sixth and seventh project instead of the two exams. I love the projects!
-
3. Great Professor, alot of the grading is up the TA's discretion but his class is amazing.
-
4. Fantastic course. The tools required were very useful and I'm sure the experience I had with them in this class will be helpful in my career. I would say that providing the makefile in project 1 was not conducive to helping the students on how to learn how to write them. Most everyone I know in the class, myself included, just copied the makefile from one project to the next. Not sure if you can require them to write it for the first project and then encourage them to reuse it, but I feel I could have learned more about how to create them had I done it on my own. Also, Canvas is better than Tophat. It's free and easy. Overall though, I really liked the course and have recommended it to many friends who have not taken it yet.
-
5. I thought it was worthwhile to learn how to use git. I only used it once and I didn't really understand what to do with it, but after this semester, I know git better. Unit test making was honestly very tedious, but I know the necessity in it. I thought Travis CI was kind of useless or pointless. All of the projects were pretty fun to do and straight forward. I learned a great deal from them and appreciated that we were required to have a consistent style or semantics for our code. All of the speakers we had were pretty cool. I learned more about the work world for computer science majors and about the recruitment process for jobs.
-
6. Great class. Expectations are high and assignments are challenging, but at the same time, are manageable as long as work is put in. Really felt like I was learning while doing assignments rather than writing as much code as I could as quickly as I could just to keep up. Instructor is always amazingly prepared for lectures and knows where he is going with everything he presents. I have not been in a better organized Computer Science course to date. I wish my other professors could prepare their lectures half as well as Downing does.
-
7. The way that projects used industry tools was misleading. e.g. submitting Doxygen documentation by committing the finished HTML rather than the Doxyfile. Also, the requirement to have a set number of issues in the GitHub issue tracker led to filing bogus issues, and we got no experience with how an issue tracker actually gets used. Also, considering that tests are essentially just small projects that you have to do in a short time span, I felt that tests have too much importance in the grading scheme relative to projects.
-
8. I learned quite a lot while taking CS371p I learned about gov, valgrind and travis ci. Those tools are used by software developers to make sure the software they build continues to compile and function the way it should. git and Github were taught to me in my current internship, but I do believe that we are quite expected to know how to push code into github and work as teams, which is great. Projects required quite a bit of time, but once they were working my partner and I always had the feeling of satisfaction. I really enjoy sitting down in Downing's lectures and hearing talk about all the different things that could happen in C . Most of the time I am unaware of all the possible scenarios but Downing always opens up my mind.
-
9. Great course. It's a difficult class but very rewarding. Professor Downing is the best!
-
10. I understand that calling on people in class helps make people pay attention, but for me, it made me uncomfortable even when I wasn't the one being called on. I prefer lectures in which you are free to ask questions but don't feel on the spot or nervous, it kind of made me not want to come to class. I didn't find the textbook helpful at all, it was difficult to follow and didn't seem to be worth the time to read. I also wish that the test was more varied than just writing straight code, I think it would've been more fair to cover a more varied portion of the material T F, short answer, fill in the blank, etc. Other than that, I really appreciate the enthusiasm and breadth of knowledge that you shared with the class.
-
11. Great course! CS students all say Downing is a great professor, and I think his reputation is well-deserved. He was responsive to questions on Piazza--important since the assignments could be open-ended--and well-prepared for lecture. My one complaint--also noted by others--is that the daily quizzes were a big percentage of our grade and we couldn't make up quizzes even for excused absences like out-of-state job interviews. As a graduating senior with a 4.0 I had a lot of absences for interviews, which tanked my quiz grade despite near-perfect scores on all the tests and assignments. And I just can't decline an onsite interview with Google to take a quiz. In the future, maybe you could drop the 5 lowest quizzes like you used to?
-
12. I really enjoyed the lectures, but I felt like we barely talked about Object Oriented Concepts. Sure, we had inheritance and polymorphism, but there was very little discussion about specific designs and going about the process. I learned a great deal in class, but I feel like the title should be called Intro to C than Object Oriented Programming. There were only a handful of time that languages beside C were talked about. I still loved the class, and I am very glad that I took it. I just believe that the title of the course can be misleading.
-
13. I thoroughly enjoyed this course and found the integration of the primary tools to be effective in the context of the course. I thought that a little more emphasis could have been placed on the correct usage of Doxygen. I felt that all the projects were educational and fair, but found Darwin to be slightly excessive in terms of workload, especially when compared to Life which was one of the "shortest" projects.
-
14. First downside of this course... TAs that are in charge of grading, and yet can't speak English. Second downside of this course... the course is titled "Object-Oriented Programming," but should be titled "Intro to C "... This class is about knowing basic C and using its STL. There's not much covered in the way of OO design. If you are not okay with pencil-to-paper coding tests, then that is a third downside but I don't mind that format for testing. Other than all that, the course is solid. Downing's a great teacher, the projects are reasonably challenging and usually not too time-consuming, and the class fully prepares you for the tests. But if you already know C , you probably won't get much of anything out of this class.
-
15. I wish that there was a bit more emphasis on why or in what use cases you would be likely to do the object-oriented practices that we covered later in the semester and not just what is technically possible. I also would have preferred that laptops be

allowed to take notes during class. The provided note files are useful, but with no explanations it is hard to remember what they are trying to show sometimes. I simply cannot write fast enough by hand to record all of the material covered in class, so laptop use would have increased my comprehension significantly. Aside from these two complaints, this class was extremely valuable to me and definitely ranks as one of the most useful I've taken so far.

16. I feel like the quiz is hard, even I prepared carefully, I still cannot get it right. Then I don't really want to spent lots of time preparing for quiz.

17. I feel like I didn't really know how to use gcov valgrind, it was just something that magically got outputted when I ran tests. Everything else was great.

18. Having the tests consist only of code is a poor choice in my opinion. I, for one, have trouble coming up with fantastic ideas on the spot, with little to no bugs. It made the tests rely upon coming up with code quickly, which I think is a poor way to evaluate experience.

19. loved all the tools, projects, and speakers. However, I felt that the course could elaborate on the materials in the book because it is as if we never used it.

20. I liked how you called on people and really got everyone involved in discussion and understanding what is really going on. This has been by far the best experience I've had in a UT class, CS or otherwise.

21. This course was a great intro to C . However, I took it expecting a course on Object Oriented Programming. Unfortunately since I already knew C I didn't learn as much as I had hoped. Furthermore, I specifically did not like the projects Allocator and Darwin, since I've implemented the same project to allocator in 429H and a more complex and robust allocator in 439H. And in 314H I implemented a project very similar to Darwin, called Critter. Therefore I would strongly discourage honors students from taking this class, as it consisted of very similar ideas to things I have learned before.

22. no

23. The class was good but not about OOP. I learned huge amounts about Cpp, the STL, and industry tools, but we only focused on OOP towards the end of the semester and in the readings. Prof Downing is clearly very knowledgeable I thought I already knew Cpp well before, but that was nothing compared to what I learned. Having attendance checks and calling on people in class does increase participation however, I prefer more self-driven classes, which give me freedom to explore topics from the class that strike my interest. The assignments were OK, but the turn-in system and requirements are very cumbersome. I usually spent 10 percent of the time doing the actual assignment, and the rest fulfilling the assignment reqs. Overall, a great class.

24. Having quizzes every day sum up to losing quite a bit of points when I had to go out of town for 5-6 separate days for job interviews. I know classes are important but getting a job after college is more important to me.

25. The course was very helpful and I believe that it was worth while taking. The assignments were well defined, although I do not like how we changed the way we would submit the assignments halfway through the semester. Using Canvas for turning in our assignments I felt was a bit better than the others as I did not have to out of my way to look at my grade and see the rubric. As well i liked using Canvas for the quizzes, as again it kept everything in a central location and can be easily looked up.

26. This course was alright, although I found the supplemental reading to not be incredibly beneficial, and as such I didn't do it a lot of the time due to it's lack of guidance in the course. I was not a fan of the way exams were handled basing this off of exam 1 and probably exam 2 as well , because being forced to memorize syntax ! a good programmer. On the first exam I got marked heavily due to my inability to coherently express my ideas in code. I think part of the coding experience is messing up first and fixing your issue afterwards, and that includes syntax. When you only have one shot to solve a problem from scratch, it puts a lot of pressure on memorization and loses sight of the purpose of the exam knowledge retention.

27. Thank you for teaching this class. You're a good lecturer and knowledgeable on c . First of all, I think this class should be called C . The amount of things I learned about C far outweigh the things I learned about OOP. I appreciate testing out Canvas for the quizzes, but I don't think it was the right choice to make. When we were using TopHat, we could use our phones. With Canvas, the mobile app doesn't work forcing people to bring their laptops. Some quizzes were not clear. For example, the quiz about the output of the code while having "Nothing" as an answer option. Someone asked if you meant output as in output and I remember you said yes. "nothing" seemed as the logical answer as no cout was used, but it wasn't.

28. Many times Professor Downing would explain some intricacy of the assignment in class, but not have it explicitly stated on the assignment page, so if you missed class you would not necessarily know about the requirement until you happened to see something about it on Piazza which is hard to comb through or a friend told you. Also, Downing's quiz policy is a bit hypocritical in my opinion. He says, "Do whatever you can to get an internship this summer" and stresses it A LOT . But, if you miss class for an interview you cannot make up the quiz, so it's really, "Do whatever you can to get an internship this summer, but don't miss my class." This is obviously difficult when flying out because the class is 3 days a week.

29. Mr. Downing did an exceptional job at articulating concepts and engaging the students. However, the policy regarding late project submissions and missed quizzes is too demanding. Most classes have a few allowable skip days students may use without penalty throughout the semester on larger assignments, which provide a little bit of leeway on late submissions. A few missed quizzes should be allowable as well. Students would greatly benefit from these skip days, and may actually do better on projects. Also, it'd be great if Mr. Downing awarded students for correct responses when he calls on them to answer questions during class.

30. I liked projects using git, however, I did not like having projects hosted on github. I only have 5 private repos, and for someone who works alone, they are forced to either delete old projects. while i see that github provides easy access to issue tracking and continuous integration, i feel like finding an alternative would be nice also.

31. I think that it would be more beneficial to make attendance worth one point, and the quiz to be worth 6 points, rather than having attendance be a potential loss of points and quizzes being worth so much. It would encourage us to come to lecture, and give us some lenience on the quizzes.

32. GREAT COURSE REALLY ENJOY YOU SHOULD ALLOW TO GO BACK TO THE OTHER QUIZ QUESTIONS. MAKES IT ALOT EASIER TO THINK

33. Next time, make the test on paper. Canvas text box is the worst place to write code.

34. This is the most helpful course for software engineering I ever took. Also Dr Glenn provided us detailed explanation with his passion to let us get the knowledge. But I think the questions on exams are not effective enough to evaluate our learning outcome.

35. i was tempted to penalize the score for vague assignment instructions, but i suspect this is to some degree on purpose. strong emphasis was put on piazza communication and discussion, and questions posted there were answered by official or unofficial channels - a crowd-sourced socratic method. this seems to be the class's overall message use the tools available to you to their maximum effect, in all facets of development.

36. Really great professor. Did not like that the quizzes were what kept me from getting an A, but I guess I should have studied more daily. Other than that, really enjoyable class.

37. As far as not being engaging is concerned, I feel that partially relies on me because I do not particularly care for lectures and prefer the feel of sub 25 groups where every one tends to know each other and discussion is a must. That would not be a good way of doing this course though. When it came to grades...feedback and grades were slow from the third assignment and after. It was often difficult to know what exactly was needed as what was accepted and not accepted tended to change through out the last week of an assignment usually for the better though . In general I absolutely loved Downing. By far one of the best lecturers profs I've had at UT. I do wish that tests also had a small multiple choice section that was akin to the quizzes.

38. this class was not about object oriented programming. call it 'intro to C ' if that's what it's going to be.

39. The material was great, projects not so much. Didn't really learn much Object Oriented Design from the projects but learnt C which is not why I took this class. The instructions for the projects could have been more. The quizzes also weigh way too much. If you are trying to teach don't penalize us so heavily for not knowing something every single day of class. And some of the quiz questions are ambiguous too. The people from other companies talked and it was great to see how things work at a company. Good class but I wish the grading was fair.

40. Def improved since last year. Please try to avoid one word variables a lil more when you can.