

*** PROVISIONAL REPORT ***

UNIVERSITY OF TEXAS AT AUSTIN
Downing, Glenn P C S371P 51730
E100 EXPANDED

COURSE-INSTRUCTOR SURVEY
OBJECT-ORIENTED PROGRAMMING

Fall 2017 DEPARTMENT COPY
Enrollment = 53
Surveys Returned = 52

	NUMBER CHOOSING EACH RESPONSE					NO. REPLIES THIS ITEM	AVG.
	Str Disag	Disagree	Neutral	Agree	Str Agree		
1 COURSE OBJECTIVES DEFINED-EXPLAINED	0	0	1	9	42	52	4.8
2 INSTRUCTOR PREPARED	0	0	0	4	48	52	4.9
3 COMMUNICATED INFORMATION EFFECTIVELY	0	0	1	5	46	52	4.9
4 STUDENTS ENCOURAGED-ACTIVE ROLE	0	0	0	14	38	52	4.7
5 INSTRUCTOR AVAILABILITY	0	0	3	14	35	52	4.6
6 COURSE WELL-ORGANIZED	0	0	1	13	38	52	4.7
7 STUDENT FREEDOM OF EXPRESSION	0	0	3	10	39	52	4.7
8 CLASS PARTICIPATION ENCOURAGED	0	0	0	4	47	51	4.9
9 ENGAGING INSTRUCTION	1	0	0	14	37	52	4.7
10 INST. HAD THOROUGH KNOWLEDGE OF SUBJECT	0	0	0	4	48	52	4.9
11 INSTRUCTOR EXPLANATIONS CLEAR	0	0	0	12	40	52	4.8
12 GENUINELY INTERESTED IN TEACHING COURSE	0	0	1	4	47	52	4.9
13 HELPFUL COURSE MATERIALS	0	3	12	15	22	52	4.1
14 ADEQUATE INSTRUCTIONS FOR ASSIGNMENTS	0	1	7	23	21	52	4.2
15 ASSIGNMENTS AND TESTS RETURNED PROMPTLY	0	0	2	13	37	52	4.7
16 ASSIGNMENTS USUALLY WORTHWHILE	0	0	0	17	35	52	4.7
17 STUDENT PERFORMANCE EVALUATED FAIRLY	1	2	5	15	29	52	4.3
18 STUDENT PERCEPTION OF AMOUNT LEARNED	0	0	1	13	38	52	4.7
	Vry Unsat	Unsat	Satisfact	Very Good	Excellent		
19 OVERALL INSTRUCTOR RATING	0	0	0	8	44	52	4.8
20 OVERALL COURSE RATING	0	0	3	15	34	52	4.6
	Excessive	High	Right	Light	Insuff		
21 STUDENT RATING OF COURSE WORKLOAD	1	12	36	1	2	52	
	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	3	7	15	27	52	
	<u>A</u>	<u>B</u>	<u>C</u>	<u>D</u>	<u>F</u>		
23 PROBABLE COURSE GRADE	24	18	8	2	0	52	

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: 27

-
1. You're an amazing professor! My only issue was the workflow for projects. I wish we had more insight on what each of the tools did. For me, a lot of these tools I feel like I never ended up learning; I just feel like I typed the same line each project without knowing what I was actually typing (doxygen for example).
-
2. Great professor!
-
3. The tools that we used in class were definitely beneficial. I think the only caveat is that some like doxygen, gcov, and valgrind were somewhat skipped over. I did think that Docker was extremely useful, especially for being able to work on projects from my computer.
-
4. Fantastic instructor! Among the best lecturers at UT.
-
5. My only suggestion is to be more cognizant of the dynamic between students and yourself, particularly when calling on individuals in class. Often times people will attempt to answer a question and you will cut them off. While your explanations are typically better than the students, it'd be much better to let the student speak, and then fill in any gaps that they missed.
-
6. The Tools: For the most part, I feel like the tools have been helpful, but I also feel like I didn't take advantage of them fully. I'm still not sure about all the functionality of things like Docker and Valgrind for example. The Projects: The projects have been challenging (especially Darwin and Life), but very rewarding and great learning experiences. I feel like I've learned a lot about OOD and C++ as a whole. The Speakers: I went to all the speakers except 1, and they were hit or miss. Often times, they talk about things that are either too advanced for me or not interesting to me personally. However, I do think they provided some good insight into what it's like working at these companies.
-
7. I think I learned a lot in this course, which is almost an understatement. I learned a good deal about the mechanics of C++, I felt that they were conveyed pretty clearly. I think those quizzes and the fact that you called on people kept everyone's attention. I think overall this course is worthwhile for sure, it is very invaluable. I wouldn't be spending the time to write up comments if that weren't the case. I think that the tools like doxygen and clang were kinda meh, didn't really find them useful. gcov is interesting for sure. I felt that the projects were alright, just right. The use of github is pretty invaluable, and I think everyone should learn how to use git. Thanks for teaching this course, learned a whole lot.
-
8. The only complaint I have was the lack of something like slip days. Please consider replacing the 20% off each day policy with slip days instead
-
9. I wish he spent more time for polymorphism instead of spending a lot of time for the basic stuff at the beginning of this semester
-
10. Even though I enjoyed this class a lot and learned quite a big deal about how C++ works I feel that this course focused more on providing details about how C++ works rather than explaining the fundamentals of OOP. I still think this was a very valuable experience since now I feel much more confident when I write code in C++ and after taking this class I now prefer C++ over Java.
-
11. Readings did not provide much value because they were mainly things that we had learned in previous classes. (basic OOP concepts) Some of the papers that presented new ideas were not emphasized in class so I don't feel that I have really learned them. The class was well structured but the notes were not. They flowed well in class but were hard to review out of the context of the class.
-
12. In my opinion, by far, the best professor at UT. Downing motivated me to write more about what I'm passionate about as well as providing tools to help me get better at what I want to do as a career. The talks he organized for us were also a great opportunity, I, unfortunately, did not have a chance to attend them but he was thoroughly engaged in preparing each and every one of his students. Glad to have the opportunity to learn from him.
-
13. I thoroughly enjoyed this course. I enjoyed learning about CI and testing since these are things used more in an actual job that, before this class, had never been covered for me. I have three gripes with this course, however: first, the quizzes sometimes felt like gotchas, second the textbook is just kind of thrown in and never used, and third the OOP nature of the course feels like it's second place behind learning about C++. The quizzes always felt fast and they were always hard, my average on the quizzes is failing I'm sure. I'm not too sure why we have the textbook. It didn't come up on the first test and nearly never during the quizzes.
-
14. The tools: all the tools given to us (Docker, GitHub, etc.) were actually very useful. I feel like it's a good idea to be taught how to use these tools, and I'm glad that I know how to now. Docker was initially very confusing for me, so I had to ask a friend how to use it, and I would suggest going more in depth with Docker. The projects: I found the projects as fun ways to put what was taught in class to use. I appreciated the hints/instructions that you gave in class regarding the projects. The speakers: The speakers were somewhat interesting, and I like the bonus points we get from attending their talks.
-
15. I truly enjoy professor Downing. Absolutely took this course just for how much I enjoyed having him as a professor for SWE. High point of the goddamn semester!
-
16. Tooling: I think it would be helpful to spend another day or two on tooling if possible. In particular makefiles, reasons for project requirements (committing best practices and issues and stuff), etc. These may not be directly relevant to course material but would be very relevant to projects, given the heavy software engineering component. Darwin and Life were frustrating given that the duplicate HackerRank input added a lot of time (on the order of several hours) debugging. While asking students to explain material generally cleared up confusion quickly, students who thoroughly understood it and were called on tended to speed up the pace of explanation considerably. Pausing for general questions occasionally would help a lot.
-
17. I really enjoyed the projects and I regret not working hard to do well on the first project and test, but I am happy with my performance the rest of the semester.
-
18. I thoroughly enjoyed the lecture format and assignments in this class. In terms of the tools we used on the projects, I

personally didn't end up using Docker although I made a brief attempt at setting it up and wasn't able to. I did, of course, use everything else (most of it through the makefile provided). The projects were all engaging, and I especially enjoyed Darwin and Life.

19. I had no issues using any of the tools. They were all useful and very straightforward. The double github repository was a little tedious but not a big deal. The projects were a useful experience and not too difficult. The only real mistake I made was starting too late on all of them. The first couple I could do in one sitting, the last two I could not. I think they could be a little more challenging, as they felt significantly easier than the quiz and test questions. I did not go to see any of the speakers, which I regret.

20. For test 1 I felt that the Hackerrank portion did not have nearly enough time. Coming into the exam I knew to practice iterators and typed up our example 3 times. Despite being prepared I was still unable to complete the 5 HackerRank questions simply because the time limit was so short. I understand that we are expected to type quickly but I felt that my programming knowledge was not fairly measured; instead my typing abilities were tested. Projects: HackerRank is an excellent tool for projects. However the HackerRank system is NOT designed for accurate runtime testing. Runtime must be done in consistent environments (exact same machine). I frequently received different runtimes with exactly the same code on HR. Consider alternatives.

21. I think that the lecture style is great and I very much liked the hackerrank tests this semester as opposed to canvas tests in the previous semester when I took SWE.

22. I learned a lot in this course. I really like the fact that the projects required us to learn and put to use a lot of new tools used in the real-world, professional environment. The class was also fantastic! Prof. Downing puts in a lot of energy and enthusiasm that makes the class fun, interactive and really useful. He gives code examples as he explains concepts, which I think helps learn the concepts a lot better. My only complaint would be the exams. I think they don't evaluate fairly what is taught in the class, especially the HackerRank exam. Being able to solve a couple of coding problems is not fair. The questions should be brief and to the point, each one testing a concept. One mistake and your whole program (and exam) breaks.

23. While the overall course was great, I feel that both exams, which overall were the determining factor in your final grade, were poorly executed. The first midterm was fine save for a bit of a time nuisance regarding the separating into two parts, resulting in many students, myself included, sitting around for 10 minutes of exam time unable to do anything. This extra time would have saved my first exam grade. The second exam suffered from a noticeable lack of prior testing. While the intent was to allow for partial credit, the result was a mess of being unable to develop and debug a solution due to severe restriction in the testing harness and a lack of adequate time due to the complexity of the problems and not being given a final's 3 hours.

24. The second (and final) exam was probably one of the worst I've ever encountered in my entire life. This is not a sentiment that only I hold - I'd wager about 80 percent of the class failed it simply because of how poorly it was planned, how different it was compared to the first exam, and how badly these changes and expectations were communicated to the class. The way it worked meant that I didn't receive credit for a single question. I had a B in the class prior to it - I only ever missed one day of class and did very well on the projects, and I can honestly say that I learned a LOT in this class. However, the professor felt the need to do something very experimental, and I really don't appreciate being a guinea pig when my GPA is at stake.

25. There was something of a kerfuffle regarding the last exam in this class, which at the time of writing is still unresolved. For the most part, this class has been excellent. Downing is a great lecturer, and the projects are worthwhile, though there is a strong disconnect between the theoretical concepts we learn and the code we write. Except for the last exam, I feel all the assessments in this class have been fair. However, I feel that Downing did not totally understand what he was asking for with this last test. Downing's a good guy, but he can be a bit of a mad scientist, and I think this experiment exploded in his face.

26. I completed all of the readings, all of the assignments, and did well on the quizzes. I thought I understood what I was supposed to know until the last test.

27. Almost everything about the course is great. It would be nice to have short daily (optional?) homework just like the quizzes so we could practice and learn concepts without tanking our strongly weighted quiz grade. The textbook was good, but so few questions were asked about it that there was very little incentive to read it. The second midterm was very bad. It was impossible to finish in the allotted time and very hard to earn even partial credit. The first exam's format was very fair. Keep it like that. There should be some multiple choice with daily quiz-like questions. It should not be all free response, especially if you do it like the second test. Nor is it fair to tell us the format is so different only three days before.