

*** PROVISIONAL REPORT ***

UNIVERSITY OF TEXAS AT AUSTIN
 Downing, Glenn P C S371P 53120
 B100 BASIC

COURSE-INSTRUCTOR SURVEY
 OBJECT-ORIENTED PROGRAMMING

Fall 2014 DEPARTMENT COPY
 Enrollment = 78
 Surveys Returned = 63

	NUMBER CHOOSING EACH RESPONSE					NO. REPLIES THIS ITEM	AVG.
	Str	Disag	Disagree	Neutral	Agree	Str Agree	
1 COURSE OBJECTIVES DEFINED-EXPLAINED	0		1	1	11	50	63 4.7
2 INSTRUCTOR PREPARED	0		0	0	9	54	63 4.9
3 COMMUNICATED INFORMATION EFFECTIVELY	0		0	0	10	53	63 4.8
4 STUDENTS ENCOURAGED-ACTIVE ROLE	0		1	1	15	46	63 4.7
5 INSTRUCTOR AVAILABILITY	0		1	0	18	44	63 4.7
6 COURSE WELL-ORGANIZED	0		0	1	13	49	63 4.8
7 STUDENT FREEDOM OF EXPRESSION	1		2	2	17	41	63 4.5
8 HELPFUL COURSE MATERIALS	0		3	9	21	30	63 4.2
9 STUDENT PERCEPTION OF AMOUNT LEARNED	0		1	4	6	52	63 4.7
	Vry	Unsat	Unsat	Satisfact	Very Good	Excellent	
10 OVERALL INSTRUCTOR RATING	0		0	4	6	53	63 4.8
11 OVERALL COURSE RATING	0		3	5	15	40	63 4.5
	Excessive	High	Right	Light	Insuff		
12 STUDENT RATING OF COURSE WORKLOAD	3	24	36	0	0	63	
	Less 2.00	2.00-2.49	2.50-2.99	3.00-3.49	3.50-4.00		
13 OVERALL UT GRADE POINT AVERAGE	0	2	8	29	23	62	
	A	B	C	D	F		
14 PROBABLE COURSE GRADE	21	28	10	1	2	62	

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: 35

-
1. Downing is a fantastic professor and I would definitely recommend taking one of his classes. I've enjoyed his teaching style as he keeps you on your feet by randomly calling on students during the lecture to answer a series of questions about the code we're looking at. Downing really cares about seeing you succeed and I feel like you don't always get that vibe from many professors. My only complaint about the class were the quizzes. A lot of them were difficult to prepare for and at times I felt like we did not receive enough time to complete them for the coding quizzes, that is . Other than that, it was a fantastic class and I would highly recommend taking it.
-
2. Being called on in class was a bit scary in the beginning since I had little to no background in c and really little understanding of how Java and C actually worked. However, I found that being called on really forced me to think on the spot and pull from previous material we had covered in class. I enjoy the repetition of material in class. I found it really beneficial for ensuring that I understood the material.
-
3. There was a lack of instructions on programming assignments. Especially for the allocator and life assignments, I was forced to re-write code because some aspect of the project was not clarified.
-
4. Downing is probably the most useful teacher to any CS student at UT in my opinion. One of his priorities is to make you employable and he does it really well. This was one my favorite classes because he taught us C by comparing it to Java, which he knows we know, and forced us to do all our projects in C . My only request for him would be to come to class and have most of his example classes structs typed out, so he doesn't have to spend time typing it out. Overall very satisfied. Thank you Professor Downing.
-
5. Downing's Awesome. Class is Awesome. Kinda wish we had lighter penalties for the small stuff, but that's reasonable. It'd be nice if he gave us less material to go over during a given week. In order to correctly do this class, you'd have to read and understand 2 chapters, read the articles and pick out quizzable information, and try to pick out details of the lectures which might be quizzable, schedule programming with a partner, read and or comment on Piazza posts. I wish I would've knowthat before taking the class, cause it's heavy. Class is great though. Wish this was required directly after 429 before 439. I'm failing it cause I screwed up on re-submitting asgl , did mediocre on the 1st test, and screwed around on another asg
-
6. Honestly one of the best CS courses I've taken, both here at UT and when I was in school at UC Boulder. I did feel it would have been useful to relate some of the more concrete minutiae that we covered to the larger common design patterns that are present in software development, but perhaps that is better suited for SWE. More than anything however, I felt that the exposure to github, the forced project management tactics estimations, testing, issue tracking while a bit contrived for the context were fantastic tools for preparing for larger scale projects we use mecurial at work, along with CodeCollab and Quest ugh... for much of our VCS and review . Overall great class, really wish I could get a slot in SWE next semester. Alas.
-
7. Downing is a great professor. Make sure to take him.
-
8. I really enjoyed the analysis of the STL functions and the idea behind I know you don't have as much control over it, but the size of the class was rather off putting. Although I can't really complain considering the increase in size allowed me to sign up for this course. I feel that the nature of the conversation style lecture you have benefits off smaller classes because you end up calling people more often I was only called 3 or 4 times making them more engaged. I'm willing to bet you feel the same and have your hands tied in the situation ALSO! Please have a Software Engineering course in the Fall 2015 Semester! I wanted to take it this next semester but it was closed before I even registered DAY 2 of registration!
-
9. I enjoyed most of the projects, and although I spent a lot of time being frustrated with the professor and the style of the programs, I eventually came to see the benefits of the specified implementations. The grading in the course is a little excessive giving an automatic 0 on a project that almost works doesn't seem right to me. I also did not agree with the excused absence policy. I was bedridden for five days but was only able to get one of the days excused with my doctor's note, which excused me for all five of the days. The policies of the class should definitely be revisited for fairness however, I gained a lot of knowledge about C , which was beneficial. It wasn't the course I had hoped for, but it was a decent class.
-
10. It was great to be able to take this class as I did learn a whole lot about OOP. Overall, I know this class will be valuable to me and I will be able to apply a lot of what I learned and use it to expand my knowledge even further. However, the only thing I wish we had was a bit more time for the quizzes. There are many times when I rushed to get down what I wanted to write but ran out of time.
-
11. Best CS class that I have had so far. I will definitely look forward to taking one of your classes again, as you are the best lecturer that I've ever had. I think it's great the way you communicate information during lectures. I get so much out of them, and I almost never walk away confused. I think it's very valuable no matter how much I hated it at first for you to directly ask questions to students. I did not enjoy having quizzes everyday - did my best but they will ding my grade. I like that you forced us to the git tools doxygen. It will be very valuable to me. More OOP projects would have been better. Speakers didn't help me as they aren't likely to be interested in me, but still interesting to hear from I guess. Thank you!!
-
12. The git give me a great practice of version control and it also provides me the great communication about the issues with my partner
-
13. I dislike the letter grade drop after missing an arbitrary amount of quizzes.
-
14. git GitHub-I think it is great we use git and GitHub. Good skill to know. GTest-good framework to use. Relatively simple and gives a good introduction to writing tests. Valgrind-not much point to Valgrind for the almost all of the projects, but it'sValgrind, no big deal. gcov-liked the statistics it let you see. Doxygen-felt pointless. Projects-I found the products relatively simple, but they were fun to write! Seriously, I had lots of fun. Speakers-considering my past internship experience, I found the speakers to mainly be a nice break from lecture and the quizzes. I think the talks could be tweaked to be a little bit more interesting though. Right now they all felt pretty much the same Overall-great class, lots of fun!
-
15. Only two slight issues I had with the class - some project specs weren't fully given until a day or two before the project due date. - sometimes hammered some students with questions they didn't know the answer too. Got slightly awkward to sit around for

10 mins waiting for a student to try to figure out an answer that he she did not know.

16. I enjoyed the projects, workload was good. Quizzes suck, I got 100s on the projects and an A on the midterm but I feel like I get a 2 7 on every quiz even though I read.

17. I thought the projects were very challenging, but this maybe could be because I didn't have prior C knowledge. Collatz was of course the easiest, and the Voting and Darwin projects gave me headaches. I didn't really like the fact that 20 off was taken for missing files, how about less of a penalty for that. Sometimes when you're in time crunch you panic, and forget stuff.

18. I really enjoyed the class and felt that the projects helped me get a deeper understanding of the c language and OOP. I only have two complains about the class. The first being about the time given to complete the quizzes. They were several times where I was barely getting the blank sheet of paper when you announced that we had a minute left which feels more like 30 seconds . My second complain was the test and probably final . I feel the test had some questions that did not test our knowledge of the material, but tested us on tricky interview questions. I have taken your Software Engineering course and it was the same for that class. Please make the questions reflect the material more than some interview question.

19. awesome class

20. I really enjoyed this class. At first I disliked the GitHub requirements as I had never used it before, but now I am extremely glad we did. After listening to all the guest speakers, it seems there will be few things as useful as this. I think the in class forced participation is the same way. I was apprehensive about it at first, but the instructor was very good about helping people through the questions and was always kind when dealing with wrong answers.

21. I really enjoyed this course. If I could change anything, it would be the daily quizzes. I understand that they are a means of taking attendance, but it was really hard to be able to prepare for a quiz over anything everyday when trying to balance 439, iOS, and my second GEO course. I kinda just treated the quizzes as what separates the boys from the men, or the B's from the A's. All in all, an excellent course and an excellent professor.

22. Overall I enjoyed the course, and I found it extremely helpful in terms of preparing my self to solve complex problems, learning in and outs of a language, becoming motivated in learning the material, problem solving and interview preparing mentality. Projects I enjoyed most of the projects. Tools like the fact that I learned many useful tools like Github, Google Tests, Doxygen, etc. Speakers overall I liked the fact that the speakers showed to class. Though Twitter could has spent moretime talking about exciting projects, not tools. My biggest concern are the exams. They are weighted heavily and only two of them, hard to recover if trip over in the first one.

23. The class had a great structure, and allowed people to learn from trial and error during class. Although, many topics were learned regarding C and OOP, the projects didn't really allow for creativity in designing OOP code.

24. I never had a problem with Github. It's a good tool to learn and continuously use. It might be nice if you could try to get students to use aspects of git other than clone, pull, commit, push, and issues, such as forking, pull requests, merging, etc. I never did figure out how to get my method comments to show up in Doxygen. All projects were entertaining and worthwhile. However, it might have been nice to see some more core OOP principles show up earlier in projects, especially inheritance. I liked the speakers, and it was nice to hear about all those different industries. I would have liked to have heard from a smaller, startup company and or from someone in academia to get even more different perspectives.

25. I learned more in this class than any other I've ever taken at UT. I truly feel like I've become a much better programmer, plus I learned c which I did not expect when I registered . I recommended this course to all my CS friends.

26. I liked hearing from the presenters working in industry. However, I think the Schlumberger presenters were a bit dry and did not excite students as much as the other presenters. I learned the most from actually doing the projects. Some days in class were a bit repetitive in terms of going over the projects. I think that we might learn more by loosely coupling the projects from lecture and just letting us implement the projects in our own without talking extensively in lecture about them, that way we can focus on learning new and different material in lecture.

27. As I have stated multiple times before, to other students, taking a Downing course is one of the most useful things a Computer Science student can do. The courses are extremely practical and force you to gain a deep understanding of the material at hand. I will recommend this course to almost any CS student, as along as Glenn Downing is teaching it. Thank you VERY much taking the opportunity to teach me, professor Downing. You have made a difference in my learning experience at UT. I enjoyed your course. It was the definitely one of the high points of my damn college career. - third year student who took Software Engineering CS 373 AND OOP.

28. I really enjoyed the lectures. The only suggestion I have is allowing more time for the exams. On both exams I did not finish every problem, maybe I'm just slow. Thanks for your hard work!

29. Great Professor and Class!!!

30. The speakers that came throughout the semester were great. Definitely a nice way to learn more about a company and provide a small break from the class. The assignments were fun. The first assignment is a little overkill with the software engineering tools we are expected to use but it prepared us and helped us get familiarized with the tools so when the programs' difficulty rose we weren't trying to also learn the tools. I found the assignments to be fair and he spent time explaining the strategy behind writing efficient code to solve the problems. I do wish we were given a few more minutes on the daily quizzes to answer the questions but it just made me learn to write fast! Overall, this was a great class with tough exams.10 out of 10.

31. . the tools -I was mostly familiar with git, but it was very helpful to learn valgrind and gcov. We had heard of them in our systems classes but they never taught us to use it properly. GTest, as much as I'd hate to admit it, has made me a better tester, and thus, a better programmer. TDD is a pain, but saves alot of time. . the projects -Overall, I enjoyed some more than others. I liked your method of no BS for deadlines. The overhead was a bit much, but I can see the motive in teaching us software engineering practices. . the speakers -Please continue this! It's awesome.

32. Thank you for the excellent semester, Professor Downing. I wish I had more time to do better on your assignments and get to know you, but ultimately, your class has still helped me to land my first full-time job. Thank you.

33. The course was very well organized, and I learned a lot from it. The only thing I didn't like was the fact that there was a quiz

every day. While I understand that the purpose of this is to encourage coming to class and participating, I felt like it discouraged me from doing the readings. The fact that there were 37 quizzes allowed me to decide that reading wasn't worth the effort, as a few bad grades would be dropped or averaged out. As for your points on Piazza - learning about git was great, the projects were great, and I mostly enjoyed the speakers.

34. I really enjoyed the course.. it was very practical which is not something you can say very often! My only complaint is that the first half was too similar to your Generic Programming course that I took over the summer. I really liked the last couple projects where we actually got to design a program rather than just implementing a skeleton. Thank you Professor Downing!

35. This was an incredible class, and a huge thanks goes to Downing, in all of my time at UT I have never met an instructor who has understood the process of learning as much as he does. He guided us throughout the whole class and not once did I feel lost or frustrated, the lectures were simply incredible and the projects were smart and interesting. My one gripe with this class were the exams, as I felt they did not appropriately test how much a student has learned in the class. As an example, I studied about 5-7 hours for the first exam and I studied about 26 hours for the second, and I felt short on time for both exams regardless. Thank you!