

\*\*\* PROVISIONAL REPORT \*\*\*

UNIVERSITY OF TEXAS AT AUSTIN  
 Downing, Glenn P            C S371P            51040  
 E100 EXPANDED

COURSE-INSTRUCTOR SURVEY  
 OBJECT-ORIENTED PROGRAMMING

Spring 2019 DEPARTMENT COPY  
 Grade-eligible 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	1	3	12	49	65	4.7
2 INSTRUCTOR PREPARED	0	0	2	7	56	65	4.8
3 COMMUNICATED INFORMATION EFFECTIVELY	0	0	3	14	48	65	4.7
4 STUDENTS ENCOURAGED-ACTIVE ROLE	0	0	4	17	44	65	4.6
5 INSTRUCTOR AVAILABILITY	2	1	11	14	37	65	4.3
6 COURSE WELL-ORGANIZED	1	1	3	14	46	65	4.6
7 STUDENT FREEDOM OF EXPRESSION	1	0	6	19	39	65	4.5
8 CLASS PARTICIPATION ENCOURAGED	0	0	1	7	57	65	4.9
9 ENGAGING INSTRUCTION	0	3	7	13	42	65	4.4
10 INST. HAD THOROUGH KNOWLEDGE OF SUBJECT	0	0	2	6	57	65	4.8
11 INSTRUCTOR EXPLANATIONS CLEAR	0	0	4	17	44	65	4.6
12 GENUINELY INTERESTED IN TEACHING COURSE	0	0	2	8	55	65	4.8
13 HELPFUL COURSE MATERIALS	1	6	17	18	23	65	3.9
14 ADEQUATE INSTRUCTIONS FOR ASSIGNMENTS	0	5	5	29	26	65	4.2
15 ASSIGNMENTS AND TESTS RETURNED PROMPTLY	0	0	2	21	42	65	4.6
16 ASSIGNMENTS USUALLY WORTHWHILE	1	1	6	22	35	65	4.4
17 STUDENT PERFORMANCE EVALUATED FAIRLY	2	1	6	23	33	65	4.3
18 STUDENT PERCEPTION OF AMOUNT LEARNED	0	0	3	19	42	64	4.6
	Vry Unsat	Unsat	Satisfact	Very Good	Excellent		
19 OVERALL INSTRUCTOR RATING	0	0	4	17	44	65	4.6
20 OVERALL COURSE RATING	0	1	8	20	36	65	4.4
	Excessive	High	Right	Light	Insuff		
21 STUDENT RATING OF COURSE WORKLOAD	1	20	41	2	1	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	0	6	24	35	65	
	<u>A</u>	<u>B</u>	<u>C</u>	<u>D</u>	<u>F</u>		
23 PROBABLE COURSE GRADE	35	17	10	2	0	64	

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. Some projects had very little starting information, so it was difficult to get started, but the course overall was very insightful and interesting.

---

  2. While I learned a lot, I think the workload in this class was excessive given the subject matter. Also, tests were unnecessarily difficult. I would rather have had a major project to complete in place of the HackerRank tests. This would have given much more real-world experience (ability to use the docs!!). Overall, I enjoyed the course. Professor Downing was excellent.

---

  3. I really enjoyed the cold calling technique that was used because it kept me paying attention in class even when I was tired from an all-nighter or just not feeling it that day. The projects were easy and hard in different ways, the actual problem was easy to solve, but all of the other extra stuff we had to do was confusing at times, but the TA's and Dr. Downing were very knowledgeable and were able to help. The makefile really confused me for the last Project, but after finally figuring it out it made sense and I feel like I learned a lot about how it works. I would say Project 3 was the worse Project for me because the time we were given was basically 4 days since many people were not able to work on it during Spring break.

---

  4. Really enjoyed Downing as a professor. Likely the best CS professor at UT.

---

  5. The main gripe I have is banning the use of getters / setters. This forces us to create arbitrary methods that are packed with args and wouldn't see light of day under code review. I agree that theoretically it's good to avoid their usage but industry standards don't follow this ideology. I also wish that notes were provided ahead of time rather than released in the days after a lecture so people like me can read the notes ahead of time and understand the main material. Then we can reinforce understanding in the lecture rather than frantically try to learn the material before being called upon. I did enjoy the course, but it felt more like a C++ prep course than teaching object-oriented concepts like design patterns. Docker was great.

---

  6. I am a little disappointed that there is no partial credit on the exams. I received almost no credit for a particular question on the first exam as a result of a single missing line of code, so I don't feel the grade I received is indicative of my level of understanding. The exams are not so long that partial grading should be unmanageable, so I'm not sure why the current policy is what it is. I felt that the projects could have started being released earlier in the semester and that there could have been more of an emphasis on object-oriented design. I enjoyed implementing an iterator for Allocator and the class inheritance in Life, but I would like to see maybe one more project that really ties all of these themes and more together.

---

  7. The blog should be optional and for extra credit (all or nothing). It did not add much learning to the course and makes more sense for the software engineering course (since that comes with a writing flag). I think that you will still have participation if you make it extra credit.

---

  8. The projects were diverse in topics and helped enforce the topics that were learned in class. They were not too difficult, but not easy either, I felt that they were perfectly balanced. The tools used during the course (used on the projects) were my least favorite thing about the course. Once I got used to them, they were very helpful, and cool to learn. However, some of the machines do not have all the tools, so finding the right machine that had everything was difficult at first. Even then, docker was the tool that I could never get to work, but ended up not needing it on my own personal machine, as it was built into GitLab. Overall the tools were a minor inconvenience, and the course ended up being one of my favorite here at UT.

---

  9. This was probably the most enjoyable CS class I've taken. The projects were hugely helpful, the makefile made the tools easier to use, and I only went to the Alan and Elaine ethics talk, but that was also very entertaining.

---

  10. This class was an excellent learning experience. We learned so many useful tools like astyle, cppcheck, doxygen, gcov, git and many more. The projects were interesting, challenging but doable, and educational. Guest speakers from industry spoke and gave useful advice and shared their insights. All in all, this class is one of the most helpful classes at UT for preparing for a job in industry and learning how to effectively use C++ and related tools.

---

  11. Professor Downing's office hours were often canceled, and at one point they were rescheduled, but I tried to attend and he was not in his office. This was frustrating to me because it was the day before a project was due, and my partner and I were hoping to get help at that point. Additionally, during the first exam HackerRank froze for me for about 7 or 8 minutes. The exam felt a bit pressed for time to begin with, and I ended up not finishing it. I was almost done, and probably could have finished if the site had not frozen. Other than that though, he was a fantastic lecturer - one of the best that I have had at UT. I really enjoyed taking the class, and I learned a great deal about software development and programming in C++.

---

  12. I think Professor Downing explains the lecture material very well. I feel that more explanation is needed for the last project.

---

  13. Great teacher and great projects that help you learn about OOP.

---

  14. I didn't mind using most of the tools, though using Google test felt a bit pointless since by the time I was making the unit tests I already knew that everything was working properly. It didn't feel necessary to test any individual part of my code for any of the projects. The projects weren't very difficult, but I would've liked to see more integration of the object oriented programming principles into them (and into the whole class, really). As for the speakers, I didn't go to the optional speaker sessions since it didn't feel very beneficial for me to attend them. I honestly did not enjoy the sessions with Dr. Rich and Cline, since I had previous knowledge on almost all that they had to talk about and I have almost no interest in it.

---

  15. I feel like overall, Downing is a fantastic instructor. He has a way of explaining topics that is very clear to all students. I feel like all the projects in this course were very interesting and worthwhile learning experiences. Sometimes I felt like I knew the material, doing the projects revealed many nuances. On the tests, I feel like they are fair assessments, however the inability to add print statements to your code makes subtle bugs nearly impossible to find. I did very well on the first exam, but I feel that was partially luck due to the fact that I was able to locate my bug with very intense eyeballing of my code. A few of the quizzes felt like they went over material taught AFTER the quiz was given, but the majority were fair.

---

  16. Class notes can be a little hard to find.

---

  17. Overall, the class was alright. I think that the blog posts were pretty unnecessary as I don't see how they're relevant to the class. Also, I think we could have spent less time on virtual methods and spent more on smart pointers, because we only learned about them the day before the exam, so I didn't have much time to understand the concept. I think the tools we used in the projects were pretty helpful and helped me get experience with industry tools.

---

  18. The tools: Eventually found out the proper way to use each after learning more about makefiles, but future semesters could

probably benefit from a few short tutorials on the tools and how makefiles work. The projects: Projects were fine, only ever had a few issues like a broken assert on ConwayCell HackerRank test 4, otherwise the HR submissions were fine. The speakers: Didn't attend the bonus speeches, but it was nice to see and hear from Dr. Rich and Dr. Cline again. Their presentation was engaging and informative. Extra: The blogs really seem to just be a tedious extra filler assignment to do each week. They really don't "refresh" memory too much to write and a lot of people ignore writing about OOP to just blog about their lives.

19. I thought the tools required for the projects were really helpful, and it was a good learning experience to work with them. The talks were also really interesting, especially Dr. Rich and Dr. Cline.

20. Many tools the instructor introduced are useful. You can easily format your code using astyle, you can check memory using Valgrind, you can test yours and others' code easily using Google Test. These are all tools that I will use in my future career, I appreciate that instructor introduce so many tools. The projects are not hard, but you have to cooperate with different people, which practice my skill of cooperating with others. This's an easy A course if you fully understand what the instructors teach in the class.

21. I usually had some trouble with at least one of the tools for each project submission. I liked GitLab and using CO a lot. HackerRank and Google Test were also nice. Gcov gave me trouble in some of the projects and I spent many hours trying to get that right. Valgrind was nice and definitely useful for the future, but sometimes it would give different results based on what machine I ran it on. The projects were overall good. My only complaint is sometimes the guidelines would change after being posted and sometimes that meant I needed to go back and change things. The speakers were all good.

22. Writing out the notes during class was helpful since I could just focus on what was being said, but they often didn't make much sense when I tried to review them afterwards. Even though I went to class and was paying attention, it was hard to remember exactly what was said, and when I tried to take notes myself I missed information. Also, as much as asking questions was encouraged, it was difficult to get Downing's attention during class. Multiple Piazza questions were left without real or good answers (TAs really carried Piazza). I feel like most of it was more of a C++ class than an OOP class though. Overall though, one of the best upper division electives at UT.

23. More than anything, I appreciate the effort Professor Downing put into reaching out to students that were struggling in the class and seeing how they could be helped to be more successful later on in the course. Without those opportunities, I don't know if I would have done as well in the course. It was more than worthwhile to learn the intricacies of C++, and the projects were at least interesting if not fun. Glad I had the opportunity to take this class this semester.

24. Really enjoyed the class. Brian and Katherin were both great! Very helpful :)

25. I dislike the hacker rank tests, I do not feel as if they present an accurate representation of your capabilities, as the things that typically hold you up aren't issues of lacking knowledge of what the question is but simple, stupid, vague compilation errors that would take less than 30 seconds to Google and solve in a real working environment, but can take a half hour to work through in an exam setting

26. No enough to do the exams

27. The tools were all worthwhile and easy to use. The projects were all okay, except I didn't like how Allocator was due during the same week that it was clearly explained in class, after spring break, which made it rushed. The speakers were all good and interesting.

28. I've taken quite a few CS courses in my time here that I did not enjoy. You'll be delighted to know that this is not one of them. As someone who is interested in software development, I struggle to find good CS classes to take that are relevant to me. I'd recommend this course to anyone in my position. Gripes: 1. Range Iterator is not a good example of an iterator, and it cost me points on the first test. 2. Did not enjoy having to learn something 36 hours before being tested on it, but then again, I'm kinda dumb.

29. The Tools: It was interesting to learn how to use some of the tools throughout the projects. The Projects: They were all pretty interesting and enjoyable projects except Allocator. That one was not enjoyable because of its whole first week being during spring break and not being able to ask questions, and then rushing to finish it once we got back. The Speakers: All generally good.

30. The blogs felt a little like busy-work sometimes, but I understand their objective. I didn't like how about 60 points on our second exams were topics covered in the last lecture (and were a little rushed)