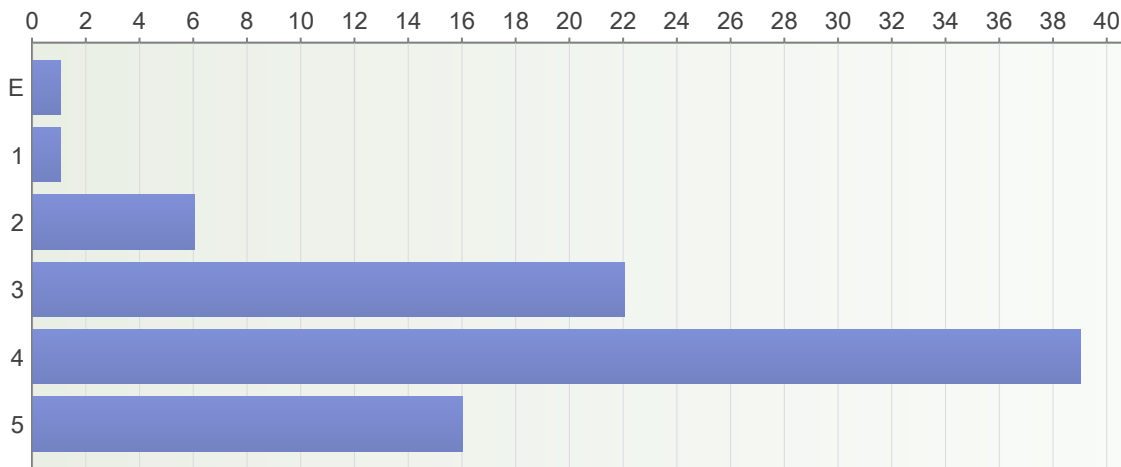


## ELEC-A7151 Object oriented programming with C++ (2020-09-08 - 2020-12-11)

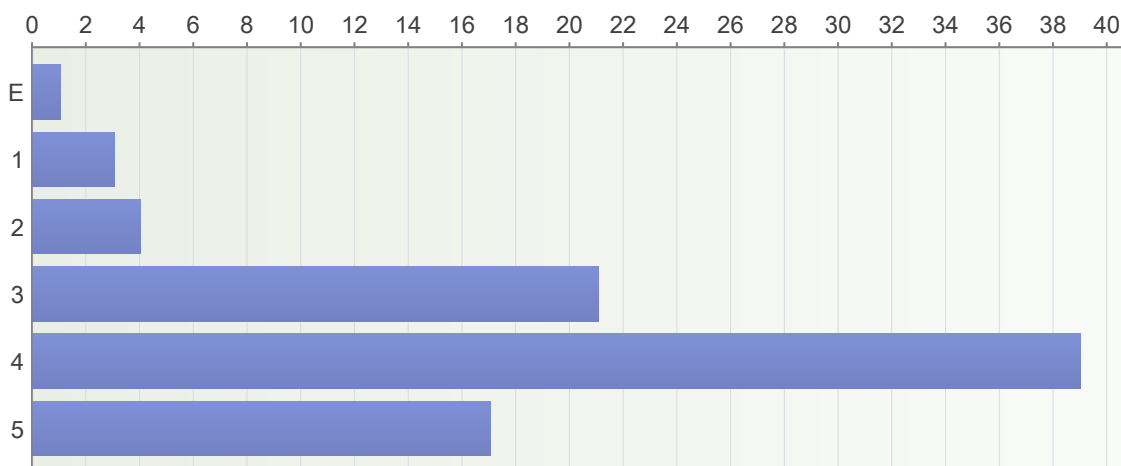
1. My overall assessment of the course E=Not applicable, 1=Fair, 2=Satisfactory, 3=Good, 4=Very good, 5=Excellent

Number of respondents: 85



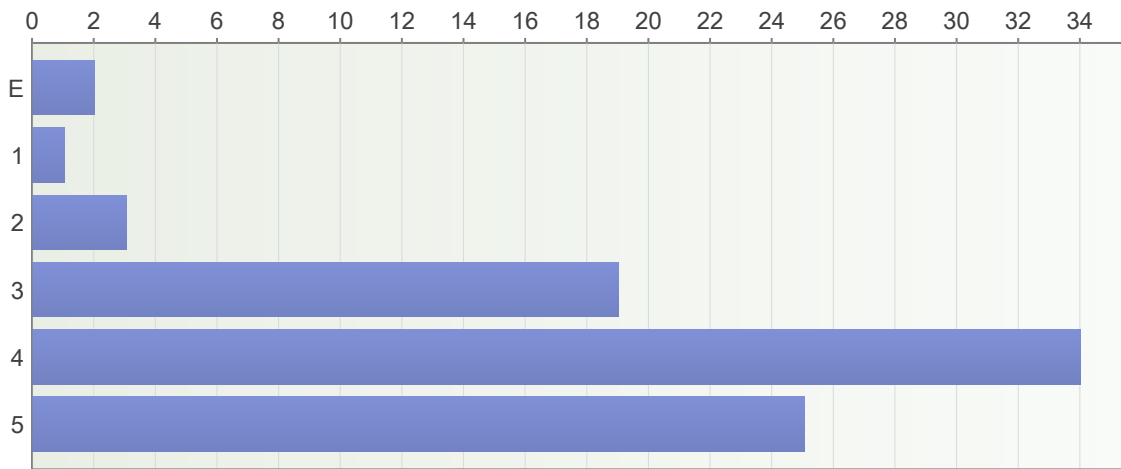
2. The teaching methods (lectures, labs, group work, online study, assignments etc.) supported my learning E=Not applicable, 1=Strongly disagree, 2=Disagree, 3=Neither agree nor disagree, 4=Agree, 5=Strongly agree

Number of respondents: 85



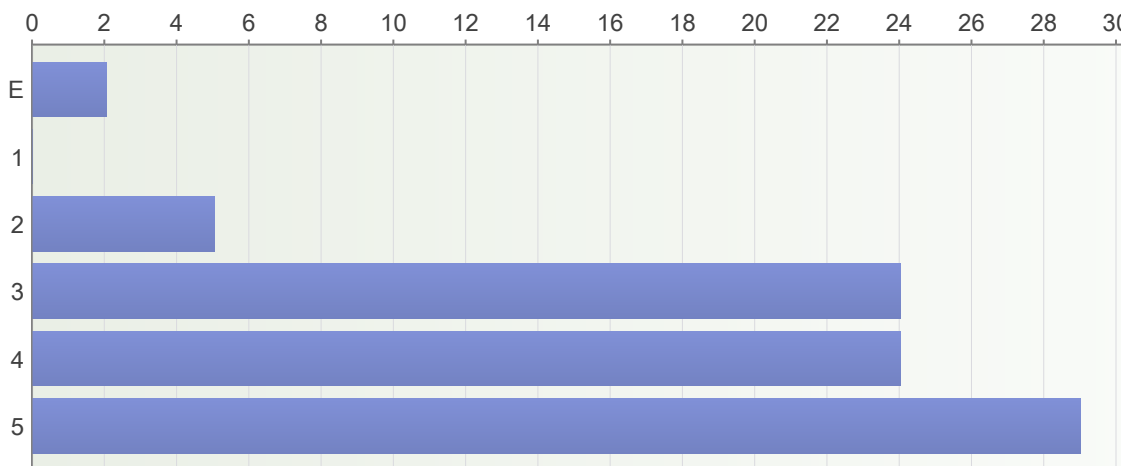
3. I am pleased with my study effort on this course E=Not applicable, 1=Strongly disagree, 2=Disagree, 3=Neither agree nor disagree, 4=Agree, 5=Strongly agree

Number of respondents: 84



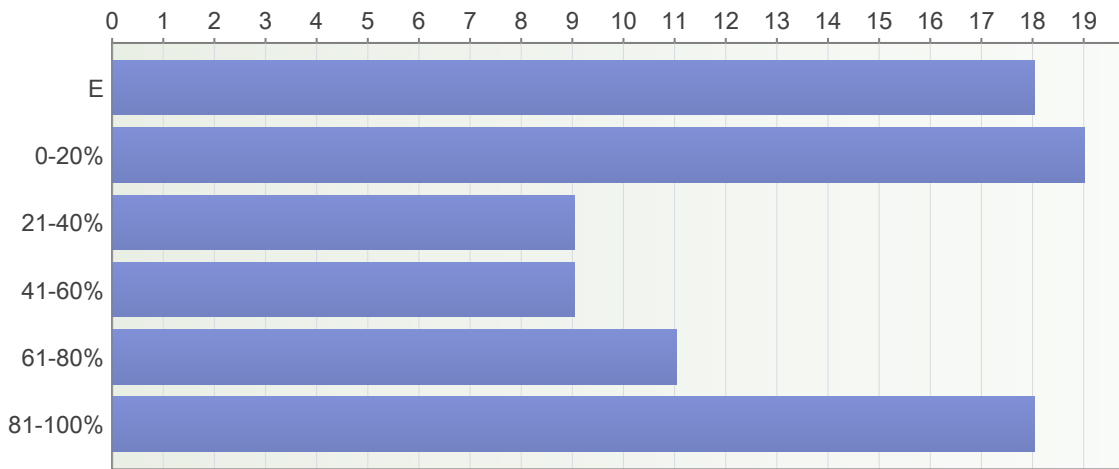
4. How much time did you use to complete the course in relation to the credits obtained (1cr=27h)? E=Not applicable, 1= Considerably less time, 2= Slightly less time, 3= The right amount of time, 4= Slightly more time, 5= Considerably more time

Number of respondents: 84



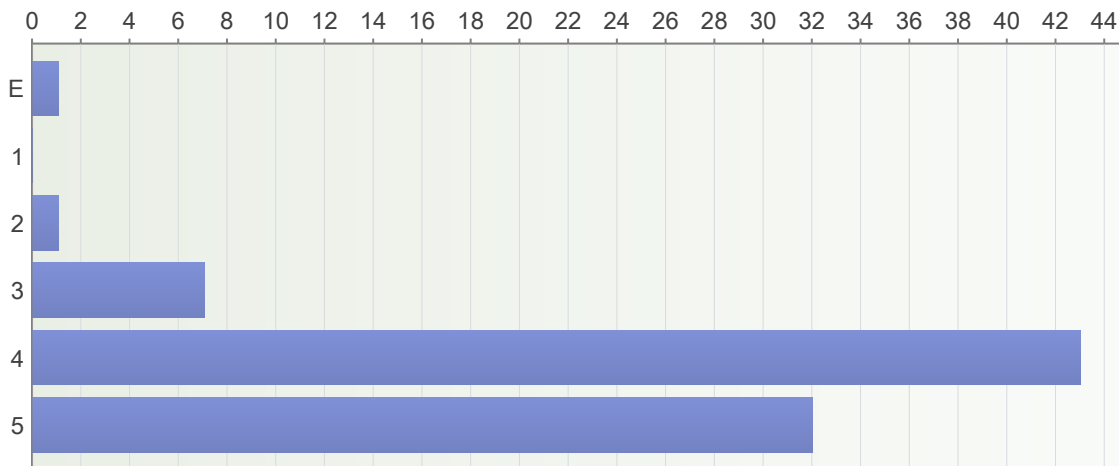
5. I was present and participated in the teaching events (estimation) E=Not applicable, 0-20 %, 21-40 %, 41-60 %, 61-80 %, 81-100 %

Number of respondents: 84



6. I understood what I was supposed to learn in this course E=Not applicable 1=Strongly disagree 2=Disagree 3=Neither agree nor disagree 4=Agree 5=Strongly agree

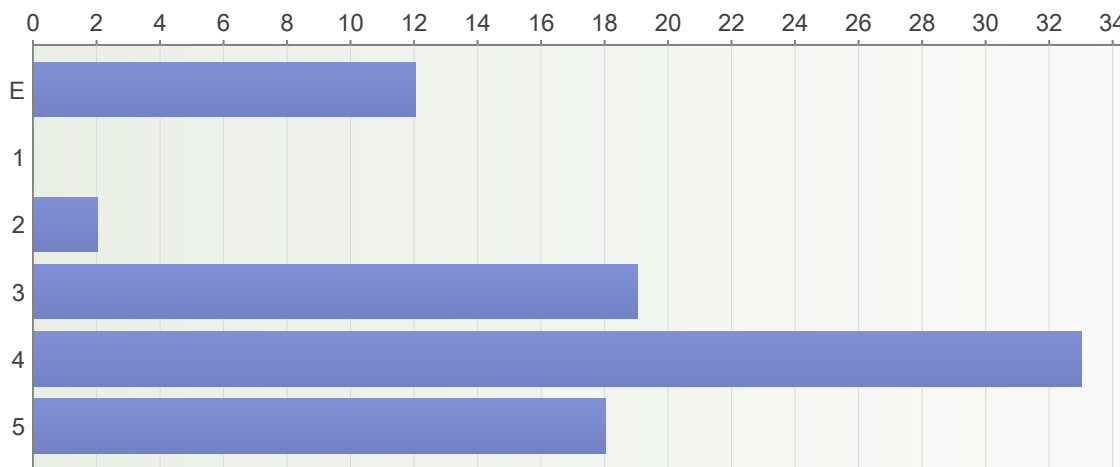
Number of respondents: 84



7. The evaluation was in line with the content and teaching of the course E=Not applicable

1=Strongly disagree 2=Disagree 3=Neither agree nor disagree 4=Agree 5=Strongly agree

Number of respondents: 84



8. The course developed the following working life skills E=Not applicable 1=Strongly disagree

2=Disagree 3=Neither agree nor disagree 4=Agree 5=Strongly agree

Number of respondents: 83

	E	1	2	3	4	5	Total	Average
Teamwork and collaboration skills	1	1	4	12	35	30	83	5.04
Verbal communication skills	4	2	10	24	32	11	83	4.34
Written communication skills	4	4	6	23	36	10	83	4.36
Skills in information and communication technology	2	1	2	24	33	21	83	4.78
Project management skills	1	0	4	16	41	20	82	4.9
Reasoning and problem solving skills	1	0	0	16	32	33	82	5.16
Research and information management skills	3	2	1	27	35	15	83	4.61
Foreign language skills	10	6	5	25	25	12	83	4.02
Practical skills (e.g. measuring)	13	3	11	33	15	7	82	3.67
Total	39	19	43	200	284	159	744	4.54

## 9. General comments and suggestions (on such matters as course material, teaching methods, course arrangements, ways of completing the course/examinations and unnecessary overlaps with other course contents, or other matters).

Number of respondents: 39

- Due to covid i have had very big troubles communicating and working with my team on team project....
- Exercise rounds played a very small part in the course, and was thus not entirely satisfactory for me. Most of the material was pretty easy too. The project has taken lots of time, and the materials for that is pretty bad (lots of mistakes in the information materials => spending lots of time debugging things that shouldn't need to be debugged). For instance, SFML 2.5 exists on the university computers, but we were told to use SFML 2.4. with Ubuntu 20.04 (this doesn't necessarily work by just installing it), and that the university computers also had 2.4. It would also have been beneficial to get some more information on X11 forwarding with WSL beforehand.
- The project has too much weight on the grading. I suggest adding an additional week on CMake or other build software, and reduce the project weight and scope a bit. First 2 weeks of the project were spent trying to get CMake to work on VS Code and Windows 10+Mingw, and we scoured through every guide available, without being able to run CMake properly. Finally we gave up on cross-platform building. My current opinion of CMake as of right now is that it is absolute ass to use and building a C++ project in general is way too big of a hassle to seriously consider using C++ over other languages for any project.
- I am sure that I will need C++ in the future, so that is the biggest reason why I took this course. But there was a lot of work. Especially the exercise rounds took much time.
- The project part was awesome
- Projektiosuutta varten olisi hyvä olla kierros c++ ohjelmien rakentamisesta, kirjastojen linkittämisestä ja cmaken käytöstä jos niitä lähes vaaditaan projektiin toteuttamisessa.
- The assignments were good. The 50% point reduction from valgrind errors seemed high. If the student is able to make the code work with some valgrind errors, it means that the student has understood the main learning objectives properly. If preventing valgrind errors would be one of the main learning objectives, then I would understand it. I liked the project work, but I had to spend multiple hours just to get the coding environment to work. This was unfortunate, as I would like to concentrate on the code itself.
- The group project was a great learning experience.
- I think that the weekly exercises were very well made.
- Overall a good course, but the project workload does not distribute even close to evenly among the group members, which caused this course to be more work for the 5cr than usually.
- The course personnel was not active at all during the team project phase of the course. Would have been nice to have the personal assistant assigned to us to communicate with our team.
- Some of the exercises and their instructions were very confusing.  
For example some exercises had different instructions on course website and inside the exercise file. Also the instructions for the set up of the programs needed were counter-intuitive and could have been made a lot clearer. The order of the instructions should maybe be rearranged. It's also not good that some of the needed instructions were only in video format.  
I felt there was too big of a leap in difficulty and time needed between the exercise rounds 3 and 4.  
The instructions for the project subject felt like some important features were left out of them and needed to be told us by the teacher later which created extra stress which could have been avoided if they were put clearly in the instructions
- Overall, the course is pretty good. It teaches me the basic knowledge of C++ programming and further-developed application software based on C++, such as SFML and Box2D. However, almost all the main learning materials are text. I think that I would learn better if there were programming practical examples involved, like videos and so on. Assistants though Teams were helpful, although there is usually a long queue for help in exercises.
- Group work was the best. The theory taught in the first half made sense in the second half. It was a good for learning to collaborate with others who had experience on different things than I had. It was very good having a premade team of 4 before joining the course. That eased the organizing, delegation and trust so that the project

had a good basis. It could be much more difficult otherwise.

I learned a lot and feel that I was also able to teach my teammates about project management, git, storing data in Json, and something about C++ to like Heap vs Stack etc.

I enjoyed also the recorded videos Pasi made in the beginning of the course going through the assignments. It was a good way to learn after having solved them in own way.

I want to give a big thanks to Michael (or Severi?) Casserly, who helped me during Friday evenings. Thanks to his teaching I was able to learn much more and be less daunted by C++.

Overall a great course, learned a lot and worked very well during the pandemic as well. Thanks!

- This course required a lot more working hours than what a 5 credit course should require. This is a great course but it should definitely be a 10 credit course as it requires so much.
- It could be useful to have one exercise round dedicated to the tools, as programming isn't just about writing the source code but also being able to work the code in different environments. The current round zero works but it could go a bit deeper and have some quizzes and other tasks. The tools round could for instance contain info about cmake, make and libraries (sfml, qt, ...) and go a bit deeper into those what they do, how I can use them. This would also make starting the project easier for students.
- The course materials are well prepared and the mentors gave great support
- 

During the software project there could have been more support from the TA. Not a criticism of our TA, but I feel that it was pretty lucky to gain anything from the 1 half our weekly meet, since ofcourse all the questions and problems arise just after that meet and then they are solved by the next meet etc. I was lucky and had a pretty good team with good communication and planning but I can imagine that in a worse group the second half of the course would have felt very, too independent. Maybe there were more resources to get support, but then they weren't displayed well enough. Maybe like teams channels for all project topics. Or dedicated exercise sessions still in the 2nd period ( I know there aren't enough resources but maybe 1 or 2 2h sessions could be nice)

Also I think there might be huge level differences in groups. If all 4 students happen to be for ex.CS minor students, and have much less practical project experience, like building the whole working environment and getting the file/program structure right.

Not sure if there is anything to do about that or if it is even a problem but i could see some unfairness in grades arising from that.

- The TIM modules were great. Even though I spent slightly more time with those, it gave me a lot. During the group work, it was sometimes difficult to cooperate with less motivated teammates but that's not the problem caused by course staff. In general, good course!
- The task descriptions during the round were too inaccurate, so it was hard to complete the assignments because I did not now what should I had implemented. In a nut shield, too much guessing were needed.

There should be some kind of tutorial how to use box2d and SFML before the project. It took a lot of time to just get them installed and working

- I wish the course has some lectures or exercises related to graphic library and CMake stuff. Since the course and exercises focus is OOP, it is so confusing and complicated to do the non-OOP tasks.
  - 
  - It was nice to have the weekly project meetings with supervisor. They were short and did not took so much time, but it made asking help for smaller questions a lot easier.
- The project was nice way to use the skills learned from the exercises and also learn a lot of new.

- The course was ok. It was able to teach the basics of c++ programming. My biggest concern was in the exercise rounds. There were some tasks that took lot of time, due to a minimal error. Even the course assistants were not able to help, In these cases the program worked fine locally but not on tim. Could it be possible to get a better error handling to the automated tests, to resolve this problem.
- Liian vähän assareita piti jonottaa yli tunti kierroksissa assareille. TEAMS on huono pohja assaroinnille, koska ei ole reilua, että muut opiskelijat näkevät koodisi ja assarointisi kun sinua assaroidaan. Projektiin ei voi suositella QT5 kirjastoksi, koska se ei toimi linuxilla kunnolla ja Cmake tekeminen tälle kirjastolle on ongelmallista. Cmake on lopulta niin kompleksinen systeemi, että siitä olisi voinut olla kierros kurssilla. Linkattu Cmake tutoriaali oli liian yksinkertaista ohjelmaa varten.
- A bit too much work for just 5 credits :).
- I feel like a ton of required Computer Science courses for my bachelor's degree were all in second period, and ONLY in second period. I had 5 courses and 3 of them were required. They all also had similar deadlines and two of them had projects, so I barely managed to finish them on time. (Web Software Development, Computer Graphics being the other two)
- The limited pointer exercise was weird. I didn't really understand how it should have been done with the information we were given. It was also annoying that the next week's exercise depended on it and the instructions said it could be found on the course slack channel which we didn't even have. That caused a lot of confusion.

It would be nice to have some optional content about how group projects are different from individual projects and maybe something about how to collaborate effectively.

I guess it's good to have a project with multiple people, but if I had difficulties with the group then the 15 min assistant meetings wouldn't really have helped. I was fortunate to have a nice group even though I didn't know anyone, but the idea of not getting along or not being able to work together did stress me out before I met my group.

I'd also like to know what happens if for some reason I couldn't work with my group or there was someone who didn't work as a team player. What would the course personnel do about it? It would be nice to know beforehand. Reducing someone's grade if they don't contribute as much is fine I guess, but it wouldn't make me feel much better if I still had to tolerate the situation for a month. I'm also not sure if people would talk about problems with group dynamics in the meetings with the assistant because they might not want to be rude or throw someone under the bus in front of the whole group.

Even if dysfunctional groups are rare I think the issue should be taken seriously, because it can really make someone or the whole group miserable. I do understand that this is a complex issue and that it doesn't really have a simple answer.

I did like the course and learned a lot.

- SW project teaches many aspects of actual sw development and is essential part of the course. But maybe there could be possibility to somehow select a project in which roles of the group members are somehow more predefined by course personnel and also some kind of project skeleton is given for the group. Maybe selecting this kind of project max grade is 3 or something. I think that this could in some cases make actual learning of C++ easier. Also maybe some kind of introduction to git collaborating could be in order.
  - The project was very hard to begin as a group, since there was little information readily available on eg. compiling the written code from scratch.
  - No comments
  - The exercises were great!! I loved them, liked the game of thrones references on lectures and materials as well. The exercises and materials were far more fun to read like that.
- The test that were given to students did not always work. The grading was also bit hard, because the test did not test every function, so if one function did not work you maybe couldn't get points from any other either.

Also, projects. It's almost impossible to do cmake to QT-projects, so I would not recommend Qt without warning on course page because using that will automatically lesser points. Qt is good to use, but it is bit stupid to notice problem with cmake while doing the project

- This course was overall way too laborious and difficult. Too many and too hard exercises during the first period. Too much content to handle (everything from c course, object oriented programming and c++ language) if you are not VERY familiar with these topics already.

Programming project. I think it is weird and way too demanding, that we are introduced with things like Cmake, which is barely mentioned during the course, and then we are expected to "just learn it", when there is also a whole program to write too. Programming project would definitely need more step by step instructions on how to get started with everything (Git, Cmake etc.) to create a buildable project. Our group wasted like 3 weeks on these kinds of things because it was just so unclear what to do, even though we got assistance by group assistant.

- The course project meetings with the assistant were useless, since he didnt seem to have the tools needed to help and guide us in any way
- The weekly meetings with tutors would be more useful if they would, e.g., provide concrete feedback on the code.
- The course was well constructed and the material was good. The only problem was the amount of work required for each exercise session.
- The exersice rounds were quite fast paced. There could have been more support for using software used in the project e.g. cmake.
- Have a unit on git (using its collaborative aspects: rebase, pull requests, merge, etc.)
- Assistants were overworked during the assignments.