

Informatics Student Course Feedback 2017/18

<http://www.inf.ed.ac.uk/teaching/surveys/2017-18>

This report contains feedback from students about a course taught in the School of Informatics during the 2017/18 academic year, in response to the following questions:

- What would you say to students interested in taking this course?
- What did you find most valuable about the course?
- What improvements, if any, would you make to the course?
- Please add any other comments you have about workshops and tutors

Each course organiser receives this report as well as statistics on multiple-choice responses. All these reports, together with student feedback about individual members of teaching staff, are collected and sent to the Director of Learning and Teaching.

Please note that these are personal responses from individual students: some courses only have a few responses and a small sample can be unrepresentative.

Stereotyping and bias, especially unconscious bias, is a serious concern in any survey gathering personal responses. All students received the rubric below before completing the surveys, and you can read a brief introduction to issues of unconscious bias on the university web pages at <http://edin.ac/2iypZBv>

This information is provided for students and staff at the University of Edinburgh: **you may not redistribute or reuse it without permission**. If you would like the information in another format or want to use it in your own publication then please contact the Informatics Teaching Organisation at <http://www.inf.ed.ac.uk/teaching/contact>

Rubric given to all students taking the end-of-course feedback survey

We value your opinions on the courses you take here at the University, as they allow us to shape future delivery and development. We welcome constructive comments about your courses, whether positive or negative, and ask you to give details about any issues in order to help the course organiser to understand and address them.

We encourage you to be aware of the potential for bias in the completion of these questionnaires, so we have developed resources which may be helpful to you:

- Equality, Diversity and Unconscious Bias (<http://edin.ac/2iypZBv>)

You also have a responsibility to provide feedback in a manner which does not breach the University's Dignity and Respect Policy:

- University of Edinburgh Dignity and Respect Policy (<http://edin.ac/1Cq0VZY>)

The results of the questionnaires will never be analysed in a way that seeks to identify individual students from their responses. However, should you wish to remain anonymous, please do not identify yourself in your answers to the survey questionnaire implicitly or explicitly.

Comments Report

What advice would you give to a student taking this course in future?

- Be prepared to work a lot and work hard. The content is by no means challenging but does require rigorous and careful thinking.
- Depending on how fast and good a coder you are, you might find this course moderately easy or challenging.
- Get started with writing a compiler as soon as possible. A lot of work.
- Get the assignments completed early to allow for small tweaks if needed.
- If you are interested in compilers definitely do take it. The coursework is nice and not too hard (others may disagree).
- Start early and don't miss lectures. If you get stuck make sure to get help to fix the issue don't just ignore it.
- Start implementing early. There can always be troubles with the feedback system or with the own implementation.
- Start on the coursework early, because you can only fix something once every 12 hours.
- Start the courseworks early. Implement to pass the tests first, then extend in all the areas you knew you skipped (so you don't end up losing time on less-urgent features, despite how necessary they may be for a full language implementation).
- Start the work early and be very careful and thorough when you do.
- The course is completely applied and does not cover much theory, even though the topic might sound like it. If you are looking for a theoretical course, this is not it.
The coursework sounds difficult (100%, writing a whole compiler) but in fact most of it is given and you just need to apply it - it's not as difficult and as much work as you might think.
- Try to start the course work early. The automated tests are run twice a day. And if you wait until the last minute to start the course work you won't get any chances to see how your code runs against their test cases which they do not give to you. Also this applies for all parts but becomes more important for alter parts, but actually write your own test cases.
- start each part at least 2 weeks in advance

Comments Report

What did you find most valuable about the course?

- Being introduced to how assembly code generation worked was new to me and interesting
- Implementing the compiler by ourselves. This drew attention to some details and helped developing my programming skills.
- Lecture slides were very well organised and easy to follow, and Christophe was a very knowledgeable lecturer
- The course was a great environment to further your skills when it comes to software development, and also learn a lot about how a compiler works.
- The course was clearly structured and it provided a lot of useful information. Generally, I have gained a lot taking this course and I have enjoyed doing the coursework. I also like the fact that it's 100% coursework, more courses should be like that.
- The coursework. Lectures were less valuable.
- The fact that it is 100% coursework and has nice and helpful staff (TAs/Lecturers).
- The practical nature of the projects was extremely useful
- This course was really well organized and taught. If you actually go to all the lectures and pay attention the course work really follows from there. The course work is challenging but is all very doable and they really give you everything you need to know to do the coursework. Lectures are clear and the professor was good at answering questions people had in class and on Piazza.
- Very good introductory course to the compiling industry.
- Very interesting project, coursework was well organised (automatic testing suite, git). I liked Aaron's lectures so far because he can tell about the "real world".
- Well organised testing system and developed compiler's software design. Knowledgeable instructor.
- Working on a large project from scratch. Learning about how compilers work and now being able to apply that to my programming.

Comments Report

What improvements, if any, would you make to the course?

- Actually, I don't think this course is organized very well, though it is not a very hard module. The lecture does not link very properly to the labs and the project. Every time when we started a new section of the course, the lecturer would not introduce us how the previous parts link with the new knowledge in our project. For example, for every time when our project moved to a new part, I need to observe the Main.java class to know how each part is linked together. The lecture slides should contain more information about our project and talk about how each section are linked with each other.
- Better feedback for each part of the coursework would be welcomed. Also having the test suite working at all times.
- Don't have lectures at 9am, it just makes them more difficult to make it to and difficult to catch up if you miss them. Perhaps split up Part 2 into Parser and then name+semantic analyser or allow for more time between Part 1 and 2.
- Fewer lectures, more labs and more self-study.
- For my own interest more than anything; feedback on the approaches/design choices of our compiler would have been nice, since the coursework is a large scale project, which I haven't had much exposure to.
- I would rather not have to hand-implement the parser/lexer
- HAVE ADAM FIX HIS SCRIPT
- More coverage of formal topics (automata theory, complexity, proofs and formal analysis) in the lectures, they seemed to just prepare for the coursework.
- More feedback on our code and structure - not just automated marking.
- More theory, instead of (almost) only the relevant things for our implementation.
- The course was badly organised; the blackbox test runs were not frequent enough before deadlines, occasionally, the test runs didn't even succeed or failed for everyone, yet no deadline extension was offered with the comment "the blackbox tests are not supposed to be primary means of evaluation", yet two thirds of our mark was determined by those tests.
- The only complaint I have is that the lab section was pretty big and sometimes it was hard to get help.
- There were some technical issues with the online scoreboard at times which was frustrating.
- Though I understand the motivation behind making hidden tests something under the control of the staff (in the real world, you don't know who will be using your language and finding new ways of breaking it), I overall found it very stressful in terms of time management. Rather than be motivated by the setup to develop an extensive test suite, I found the amount of course content to absorb and implement quite overwhelming. Having this "real world" sentiment in the learning environment in combination with my other course deadlines actually ended up stopping me from enjoying the latter half of the course, and it's turned me off a bit from compiling techniques as an area of study.

Comments Report

Please add any other comments you have about workshops and tutors

- No student comments