

# 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?

- Check the slides and textbook before the lectures.
- Do all the tutorials before class, make sure to start courseworks early (there was a time when we had to alter our submission a little before the deadline because of some last-minute clarifications)
- It's a nice course. Take it if you like. :-)
- Just take, it's useful, quite easy and the exam is open book
- Learn how to test before you start the courseworks, because the supervisors will not help you
- Many of the software testing approaches may seem intuitive and even obvious but I highly recommend taking the course simply to gain the useful confidence that will pay off in any software engineering interview or job.
- Show up and make sure you get your work done. Ask questions if you're confused.
- This course is worth taking if you are interested in software testing methods and concepts.
- it's a pretty easy course, but I'd recommend paying attention as the concepts can come in useful when working in industry later on, and the coursework is quite fun.

## Comments Report

What did you find most valuable about the course?

- Both the courseworks are really good. I was not able to secure good marks, but I enjoyed it
- Different software techniques.
- Exploring various testing techniques and doing assignments with them was very helpful.
- I liked the professor's explanations, and found everything fairly straightforward.
- Learning TDD
- Probably the tutorials. Although the lectures were also good and I very much like Ajitha as a lecturer. Also, I very much liked that the coursework was done in groups (which I could choose). This little thing made it a much, much better experience.
- The different techniques in software testing
- The fact that we have had such a deep dive into all software testing methods and the pieces of coursework which were a good practical exercise that will help in real job situations.
- The practical part
- The techniques taught and the courseworks.
- The tutorials were very thought-provoking.
- This course is quite useful for work seeking.
- learning how to use JUnit extensively

## Comments Report

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

- Change the TAs, literally never answer Piazza questions and extremely unclear instructions on courseworks
- Coursework 2: Mutation testing was just a "random" try and error approach which took so much time! I think this coursework task did not meet the definition from the class of how mutation testing should be applied (definition in class: software tester know mutations and checks if they fail a test otherwise write new one; coursework: get code with mutations, try to fail test with mutations, but if you don't know what the mutations is, this looking for possible mutations can take forever, one mutation was not catchable without decompiling [the one where "a" had to be the template pattern. Finding this without decompiling is nearly impossible cause it is such a specific mutation.]
- From my point of view, the course can be organized more clearly. I know many times the teacher would help us understand the structure and the relationship between the sections. It would be much better to add these into the PPT.
- I think the lecturer needs to put in more effort to prepare her slides, instead of just taking from the internet. What she teaches during lecture was quite different from what is written in the slides.
- I would add even more examples in the lectures. BUT I'd definitely have them better prepared and present more of them either as actual code written on PC (it's faster and closer to reality) or by using such code as the starting point (as simple as, say, taking the screenshot of a small program in Eclipse and then drawing whatever def-use or other FSM graph stuff in the picture). If this was done with the tablet, it would (I think) save much time, meaning more time for more examples! Writing code by hand is tedious and time-consuming...  
Also, I did not quite like the mutations part of cw2. I think an ideal assignment should be doable, challenging and enjoyable. Here, many groups caught most of the mutants simply by applying their tests from cw1 (shouldn't the two courseworks NOT build one on top of another mark-wise?) and then some of the other mutants were particularly difficult to discover (especially the one that inserted "a" as a key in entryMap) which I (and many others) found simply discouraging and not enjoyable. I myself ended up de-compiling the code which, in the end, was definitely more fun than blindly trying to kill the annoying mutant.
- I would focus a bit more on making the structure of the ST methods more clear and easier to digest. I understood where each method was placed in the dev cycle and what it aims to achieve only during the last lecture (the revision lecture).
- Later parts of the course were covered too quickly, maybe because of the strike? Would be a good idea to add one more coursework for the uncovered testing techniques.
- Should include not just testing but keeping something secure too.
- The feedback received on assignment 1 was minimal at best, and assignment 2 was very difficult if you did not know what you didn't do for assignment 1. Assignment 2 seemed to require us to guess at what we had been marked off on in assignment 1.
- The speed of the lecture is a little bit fast.
- The second coursework involves "guessing" mutations from JAR files without any extra information given, which is kind of impossible without having a look at the actual source files.

## Comments Report

Please add any other comments you have about workshops and tutors

- Having a tutorial was good, I just struggled with understanding the way my tutor was explaining things.
- I really enjoyed the lecturer and I think she did a great job at explaining and making sure we understand what is going on. She seemed to be one of the few dedicated teachers I've had during my 4 years at UoE.
- The tutorial leaders did not always know what was going on, but generally were useful and kind.
- The tutorials were awesome! There was a lot to go through though. What about having more tutorials?
- Would be helpful
- being able to work on the tutorial step-by-step during the tutorial was quite useful