

# Informatics Student Course Feedback 2019/20

<http://www.inf.ed.ac.uk/teaching/surveys/2019-20>

This report contains feedback from students about a course taught in the School of Informatics during the 2019/20 academic year, in response to the following questions:

- What advice would you give to a student taking this course in future?
- 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, tutorials or labs on this course
- Please add any other comments you have about the presentation of course materials online and their accessibility

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

Types and Semantics for Programming Languages [INFR11114\_19-20\_SV1\_SEM1\_ONLINE\_CACORE19] -

What did you find most valuable about the course?

- Having two TAs and one lecturer basically devoting three hours each week to help you with working through the exercises, giving you instant feedback on how you're getting on with the course material. Phil Wadler's enthusiasm made me appreciate the material a lot more than I otherwise would have, so thank you for that Phil.
- It's really hard to say - the whole course was incredibly useful and interesting.
- Phillip Wadler is a wonderful teacher. Labs are good too.
- The casual, social tutorials where solutions can be freely worked on collaboratively
- This is probably the best course I've done at university. Phil is clearly very passionate about his subject area, and he's a great lecturer. He knows his stuff and is always happy to answer questions, and makes sure to check throughout the lecture that everybody understands what's going on. The amount of time and attention he gives to the course is excellent - three tutorials a week, where you can get help with the coursework, from Phil or from the TAs.

The final coursework (25% of final grade) requires students to formalize a paper to the best of their ability. The paper assigned this year \*only came out that year\* and was being formalized to a more professional standard by Phil's colleague at the same time. You really are working with up-to-date, cutting edge research for this final coursework. The coursework is challenging, and I think is a fair way to meet the university's common marking scheme.

If you're at all interested in going into academia after university, I'd strongly recommend this course - even if programming languages aren't your area of interest. I've no interest in going into academia but I still really enjoyed myself. As someone who enjoys computer science well enough, but wouldn't say I have a "passion" for it, this is the first time I've felt like I want to \*keep learning in my own time\* after finishing a course. The discussion of open research problems and recent papers is very interesting.

- Tutorial sessions are very helpful

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

- Go slower through part 2. It is easy to lose track of what everything is for.
- I felt we could have covered more material, even if it was not examinable. For instance, it might have been nice to discuss the various ways of formalising semantics (operational/denotational), or maybe look ways of formalising impure/imperative code (as immoral as such code is). I think it would have been interesting, but also not sure how much value it would add to the course.
- I would spend less time on part 1 and take it slower on part 2.

While I understand the rationale for the new assignment which has been added this year, I think it's a bit too daunting in it's current format. I, and most people I have spoken to, would like to attempt it but genuinely do not know where to even start. Perhaps spending a bit more time on the concepts developed in part 2 of this course would alleviate this issue as well.

- Part 2 of the textbook is a \*lot\* more challenging than Part 1, and significantly harder to digest. I think it should be given more time and Part 1 should be given less time.
- Scrap the essay. The University of Edinburgh needs to fix their rubbish common marking system. Adding a 25% essay just to bring down marks is the dumbest idea ever!
- The layout of the room was not ideal for a lecture, and the lecturer was not always fully prepared/would sometimes start late due to back-to-back timetabling

## Types and Semantics for Programming Languages -

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

- Do it! While it is undoubtedly challenging at times, the payoff is immense and it ends up being incredibly rewarding.
- Don't worry if things don't make sense at first - just keep attending the tutorials
- Keep up with the coursework, do the reading, attend the tutorials and ask questions and it's hard to go wrong.course
- Once the second half of the course gets underway, read some papers in other domains that apply the principles, such as those discussing functional quantum languages
- Part 1 of this course is teaching you theory which you already know, but making you implement it in ways you won't be familiar with. As such it is simple (but not always easy), yet a lot of fun. For the first month of the semester, working on the courseworks for this course became my way to procrastinate from other work I needed to do, as it was simply that much fun.

Part 2 of this course is anything but simple. While interesting, the material can be very hard to absorb. As such, the difficulty of the course rises super-linearly as you progress through the semester.

In summary, I recommend this course if A) you are genuinely interested in PL theory, or B) you are willing to trade having a ton of fun for the first 5-7 weeks for having a bit less fun in the last few weeks (I was, and I don't regret it). Also, Phil Wadler is a very passionate and funny guy, which adds a lot of value to the course.

- The exam is not as hard as the assessments. If you can do the assessments, then you will be fine on the exam. Push through the second half of the course — it's very hard and you might not understand what's going on, but you don't need to know everything for the exam.

## Types and Semantics for Programming Languages -

Please add any other comments you have about workshops, tutorials or labs on this course

- Both of the TAs and Wadler himself were all very helpful and whenever I had a question about something in the courseworks they were happy to help my figure out how to approach it.
- Great amount of tutorials, with plenty of attention from the lecturer and the TAs.
- Very helpful, would be good if they went ahead even if there isn't a lecture
- Very useful

## Types and Semantics for Programming Languages -

Please add any other comments you have about the presentation of course materials online and their accessibility.

- Does not use Learn, and uses an open source book on GitHub. The best course website I have ever seen.
- Often the textbook was stuck to too rigidly, the interactive tangents were very helpful
- Very accessible, very well presented. Though I'm a big fan of infweb, I must admit that PLFA's website is amongst the best I've seen.

Thank you -

Thank you very much for taking the time to complete this questionnaire. Your response and comments will be fully considered.

-----

Please provide any additional comments you may have about the course, the teaching on the course or the resources that support it in the box below.

- A lot of the lectures were composed of kind of reading from the textbook, and then jumping into Agda/Emacs occasionally. I don't know how I feel about this - on one hand, Phil is a fantastic teacher, and this definitely worked as a method. But it also felt a little chaotic and unprepared at times. Obviously the textbook and course very much go hand-in-hand, and the dynamic of the course doesn't necessarily lend itself to a formal lecture with slides, so I'm not sure what an alternative would be.

Anyway, really enjoyed the course, thank you!

- Learning aims are somewhat unclear with respect to how much students should understand about different lambda calculus designs. I can understand the lambda calculus papers seen in class at an informal level, and roughly understand how they can be formalised; if this is the goal then I think this should be made clear. This is interesting stuff to look at, but I think the course risks being too intensive if students need to understand more of this than is needed to do the assignments, mainly due to tutorials and assignments taking up a lot of time each week. The flip side of that is that the tutorial heavy style forces students to keep up and in turn learn more from the course, although in my opinion a lot more is learned about agda and automated proof writing than lambda calculus theory.
- i came out of most lectures in part 2 confused.