

# 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?

- Attend all of the labs and tutorials, they are the best opportunity to learn programming. And invest at least 10-20 minutes to prepare for tutorials, it is not much of an effort needed but that way you and the students in your tutorial group can actually profit from that class.
- Be prepared to find helpful information from outside sources.
- Buy the proposed text book and study from the first lecture from there.
- Code every week at least for two hours outside the labs and definitely go to all the labs as practice is the only way to really understand the topic.
- Dedicate time to practice, learn and revise alone
- Do the lab sheets! Even if some are very challenging. And don't be scared to ask a million questions in the labs even if they're stupid. You need to fully understand what you're doing.
- Do the labs and spend time on them until you understand them
- Don't expect an easy ride if you are new to programming and do as much practice as you can especially when you get to the later more difficult parts of the course.
- Don't fall behind on lab work.
- It's just amazing.
- Learn some C programming beforehand. Use online resources to help aid your understanding of course material.
- Make sure to attend as many labs as possible
- Make time to finish all the lab questions
- Start working from the very beginning, try to keep up with labs because otherwise you might be overwhelmed with the workload just before the exam, which makes it even more difficult and challenging.
- Study for the course in advance, because it assumes some previous knowledge and it gets quite difficult.
- Take your time to put in this work and understand things before you move on.
- The lectures are useful for introducing you to tools you can use, but sometimes reading the textbook or especially googling and reading around an area/problem provide a much better understanding. But most importantly, the labs and self-practice give the best results. Coding is best done in practice through trial and error, reading around and asking questions when particularly stuck. But fundamentally it is breaking down a solution into its smallest possible steps and finding the shortest/most efficient way to write that to the computer.
- Think before you touch the keyboard! It's easy to paint yourself into a corner if you try to code while you're still deciding what your goal is.
- Think twice if you are into programming.
- This course requires you to put in time and effort in order to do well. It is a good basis for programming knowledge
- make sure to do the labs and go over them, looking at what they got wrong and improving on those things

## Comments Report

What did you find most valuable about the course?

- Being able to understand computer programming at a much more fundamental level using C
- Coding was explained at the level of the machine.
- Good lecturers.
- Guided lab sessions with help if I needed it.
- I think the most important part about the course has been the push to think logically and improve one's problem solving skills.
- Lab
- Lab exercises are challenging and helpful.
- Lab sessions where I could practice the skills taught in the lectures.
- Lab sessions.
- Labs (2 Counts)
- Learning a new skill
- Programming skills that are very useful in situations outside of the computer lab
- Solving actual problems by using programming
- The depth of the course is very good. It is very appropriate to the beginners like me.
- The lab time offered and the ability to go to labs whenever you pleased to work on computing.
- The labs
  - The labs are very good, it is useful to get experience of programming with help from tutors. I also like the dashboard thing that gives almost immediate feedback - it's very useful and a nice change from having to wait 3 weeks for anything to be marked in humanities.
  - The logic way of thinking required to construct a programme.
  - The overall course structure: with two lectures, one lab and one tutorial the time is used very well to teach programming.
  - The practical programming sessions and opportunities to practice through labs. Additionally unlimited access to the labs in Appleton Tower is very helpful for practicing alone.
  - The understanding of the underlying processes, which allowed a more in depth idea of how the code should be played out. Also helped tremendously with coding in Python in another course and should be good footing for transitioning into any other language that is more applicable in today's coding.
- The weekly labs and the small groups in tutorials
- Weekly labs, 24/7 access to computer labs
- somehow forcing students to work individually on improving their skills

## Comments Report

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

- Although the course has no prerequisites it feels as though you are expected to know how to code from the very start, not in the sense of the actual topics taught or labs but in the way the course is delivered as during lectures most of the topics are taught as if you would have prior experience with it. Although lab time did help with teaching and understanding I feel if there were more helpers around since I feel like their knowledge and help is stretched thin making it difficult to get the adequate help needed to understand the topics.
- Be more aware of using unexplained jargon in lectures - I sometimes struggled to understand things. Also possibly either more labs/lab time or more tutors in the labs, often you could be waiting a while to get help when stuck on something which wastes a fair bit of time.
- Better layout of lecture notes. They are very hard to read through and pick out the main points.
- I feel like it began nicely, but since about week 5 the difficulty of the course has increased by an incredible amount. At this point I am behind every aspect, even though I have worked regularly. Compared to other courses, even though this one is very interesting, I found it too challenging and time consuming. Also, the tutorials were not very useful for me, because it was all on paper. I would rather learn in the lab in small groups with a tutor.
- I find the tutorials too short as we rarely have had time to go through all the exercises and there is never any time to discuss additional matters.
- I found the course easy even though I did not have much programming experience beforehand so for me it could have gone a bit faster, but I know that there are many other people who found it challenging.
- Introduce more coding during lectures- as it was in the end of the course.
- Lower the pace of the course as compared to any language, after having learned the commands the student needs time to understand and be able to use them. However, in the course for every command taught there were demanding and complicated programs followed that made the slides very difficult to follow.
- More clear points on exactly what level/topics are required. Also, sometimes the slides could be a little unclear, for example the structs and enums section. Finally, I felt that the course was a little slow, but I am aware that there is a large spread of abilities and backgrounds which can make this hard/impossible to balance.
- More explanation and examples of how pointers work as that part of the course was very challenging
- More guidance for how to write code with more complicated concepts, like structs and pointers.
- More labs, less lectures. Maybe 2 labs and a lecture instead of the other way around
- More lectures during which a program is written, labs with guidance (a lecture in the lab)
- More live coding in lectures would be helpful when looking at some less intuitive ideas later on in the course.
- More practical learning
- More practical work
- Slightly slower pace and more accommodation made for complete beginners
- The course becomes exponentially more difficult halfway through, which is jarring and makes the class too difficult.
- The hardest parts of the course were skimmed over whereas some of the basic concepts were elaborated in lots of detail. As a result, I feel I have only grasped the basics.
- Tutorial answers should be provided beforehand. The lecture slides can be very confusing especially from arrays, pointers and onward. I wish we would look into more practical applications of the C language. It would be good if extra sessions for revision were organised for struggling students. The learning curve for this course is fairly high and it does sometimes feel like previous experience in programming is needed to keep up.

## Comments Report

Please add any other comments you have about workshops and tutors

- It often seems like tutors are spread too thin in computer labs, which can be a problem if you have an issue with your code. Tutorials feel like a bit of a waste of time.
- [Deleted] is disrespectful towards her students. She does not create an environment that welcomes questions-- and when students do ask questions, she belittles them. I have examples, but I do not want to be specific because I fear retribution; this is the extent to which she has created an environment which promotes discomfort and unease in her classes. Her explanations are extremely unclear, and when students do not understand she becomes openly rude. She will blatantly not do portions of her job; she was not open to answering questions about lab work, even though it is specifically stated that students should contact their tutors if they have questions. It is unacceptable.
- My tutor Stefan Ivanov was particularly helpful.
- Nice, small groups.
- The tutorials could definitely be more helpful. I felt that my tutor wasn't always informed enough and that there wasn't really an opportunity to learn from our tutorial sessions. I think this is a shame as being able to discuss confusing topics and to ask for help would've been very useful.
- Tutorial is really helpful and the tutor explained the content and questions very well
- Tutorial sheets were often either trivial or impossible with very little in between
- Tutorials needed a little bit more structure
- Tutorials were not very helpful, it would be more helpful if they were conducted as smaller and more supervised lab sessions, in addition to the larger lab sessions
- Tutorials were quite difficult, and it was not always clear whether they should be done purely on paper/in our heads, or in the lab. Would prefer more lab time, instead of tutorials.

[This report has been edited to remove identifying details. The report, with the name of the tutor, has been passed to the Director of Learning and Teaching.]