Honest opinions and experiences from people on the other side to help you choose your course. Please take all opinions with a pinch of salt or ignore them. Whatever.

Also see: <http://www.imperial.ac.uk/students/academic-suppor+t/student-surveys/ug-student-surveys/ug-sole/ug-sole-results/>

SOLE: Useful for gauging the quality of courses.

[**Computing**](#_Toc137452345)

[Autumn](#_Toc137452346)

[60007 The Theory and Practice of Concurrent Programming](#_Toc137452347)

[CO343 Operations Research](#_Toc137452348)

[CO333 Robotics](#_Toc137452349)

[CO332 Advanced Computer Architecture](#_Toc137452350)

[CO337 Simulation & Modelling](#_Toc137452351)

[~~CO349 Information and Coding Theory~~](#_Toc137452352)

[CO382 Type Systems for Programming Languages](#_Toc137452353)

[EE2-13 Computer Architecture](#_Toc137452354)

[CO572 Advanced Databases](#_Toc137452355)

[Spring](#_Toc137452356)

[CO304 Logic-Based Learning](#_Toc137452357)

[CO317 Graphics](#_Toc137452358)

[CO318 Custom Computing](#_Toc137452359)

[CO316 Computer Vision](#_Toc137452360)

[CO322 Communication Computer Science in Schools](#_Toc137452361)

[CO331 Network and Web Security](#_Toc137452362)

[CO338 Pervasive Computing](#_Toc137452363)

[CO339 Performance Engineering](#_Toc137452364)

[CO347 Distributed Algorithms](#_Toc137452365)

[CO395 Introduction to Machine Learning](#_Toc137452366)

[**Mathematics**](#_Toc137452367)

[Autumn](#_Toc137452368)

[M2PM2 Algebra II](#_Toc137452369)

[M3PA23 Dynamical Systems](#_Toc137452370)

[M3S8 Time Series](#_Toc137452371)

[M3A49 Mathematical Biology](#_Toc137452372)

[M3S4 Applied Probability](#_Toc137452373)

[M3P5 Geometry of Curves and Surfaces (aka differential geometry)](#_Toc137452374)

[M3P20 Geometry I: Algebraic Curves (aka algebraic geometry)](#_Toc137452375)

[M3P8 Algebra III (aka ring theory)](#_Toc137452376)

[M3P10 Group Theory](#_Toc137452377)

[M3P17 Algebraic Combinatorics](#_Toc137452378)

[M3P65 Mathematical Logic](#_Toc137452379)

[M3P14 Number Theory](#_Toc137452380)

[MATH60015 Quantum Mechanics 1](#_Toc137452381)

[MATH60024 - Computational Linear Algebra](#_Toc137452382)

[MATH60023 - Numerical Solution of Ordinary Differential Equations](#_Toc137452383)

[Spring](#_Toc137452384)

[M2PM5 Metric Spaces and Topology](#_Toc137452385)

[M3P21 Geometry II: Algebraic Topology](#_Toc137452386)

[M3P11 Galois Theory (aka field theory)](#_Toc137452387)

[M3P12 Group Representation Theory](#_Toc137452388)

[M3P15 Algebraic Number Theory](#_Toc137452389)

[MATH97069 Dynamics of Learning and Iterated Games](#_Toc137452390)

[Introduction to Statistical Learning](#_Toc137452391)

[MATH60017 Tensor Calculus and General Relativity](#_Toc137452392)

[MATH60018 Quantum Mechanics 2](#_Toc137452393)

[MATH60025 – Computational Partial Differential Equations](#_Toc137452394)

[External](#_Toc137452395)

[Embedded Systems](#_Toc137452396)

[Horizons](#_Toc137452397)

[Korean Level 3](#_Toc137452398)

# **Computing**

# Autumn

## 60007 The Theory and Practice of Concurrent Programming

**CW:** there are two courseworks, Azaleas coursework is just following the proof templates in the slides, very nice, you’ll probably end up loosing anywhere from 2-5 marks out of 40 if you literally follow the slides and construct your proofs correctly. Ally’s coursework requires use of thread sanitiser, and for the last part, in my year noone got the question correct but at least it was worth like 10-20% of his cw – he said last year only one group got it right and the course has only been running for 2 years for me 💀.

**Spec:** Ally and Azalea are really my favourite lecturers from previous modules and the fact they collaborated for a module is bless. Azalea is a keeper and super interactive <3. Highly suggest you attend in person and raise your hand. Ally had me wanting to uninstall my life but the exam is alright, I think I may have done better on his part than azaleas.

* Anton 22/23

Agree with Anton. Seeing a module taught by either one of these lecturers is a huge win, never mind both. Went into this module hating concurrency from poor pintos experiences and generally not understanding all the concepts when they were mentioned in previous years. However I cant overstate how good the teaching is. I went to the first 2 lectures while deciding module choices and it was enough to get me hooked. Azalea’s half is theory and lots of proofs. If you didn’t do models of computation in second year, Azalea will give a slide deck of just the relevant parts and I would recommend reading it all. Her half in this module is pretty much the same as her half in models, but now with multiple threads (who’d have thunk).

Alistair’s half is practical in c++. By no means a requirement to have prior experience with that. He gives a crash course lecture at the start and if like me you skip it despite being the exact demographic its aimed at, youll be fine. Theory coursework is just like questions in the slides and tutorials, fairly chill. Practical coursework is chill until that last question. It was deliberately waited lower so it was less painful if you didn’t get it but honestly not worth the investment, no one in my year got it.

TL DR: **Take this module**, favourite of first term, maybe of the year – Alfie (22/23)

I enjoyed this module especially because both of the lecturers are great. There weren’t many concepts introduced beyond what is already taught about concurrency in previous years. Azalea’s part is Models of Computation Part 2 so maybe avoid this module if you really don’t like that. It was nice to get some C++ exposure and teaching from Ally’s side and the algorithms he teaches aren’t particularly difficult or anything new. I would generally rate this course very highly – Zaki (22/23)

I really liked this module, definitely my favourite of the year. Azalea’s part was basically an extension of Models of Computation (lots of proofs), and Ally’s part was more practical (C++). Although Ally’s marking of the coursework was a bit harsh, they gave loads of practice material and the exam was pretty similar to these, so it was fairly easy to get a good mark overall. - Alex, 2021/22

It is a really fun module, with a good combination of theoretical and practical parts. As a JMC who did not take models of computation in year 2, the proof part could be picked up very quickly. Personally I just needed to additionally watch about 1-2 lectures in models of computation to get familiar with the programming language and the proof style. The coursework is challenging and rewarding (May not be in terms of marks haha). Exam questions are similar to tutorial questions and mocks, so could definitely get a rather high mark. – Aris, 21/22

One of my favourite modules at imperial. Introduces you to a lot of concepts that are useful. For the first part, the exam questions are based on tutorial questions and the tutorial questions have fairly mechanical solutions. The 2nd part is more hands-on and is thus slightly harder but at the end of this coursework working with concurrency is a lot easier. - Pranav, 21/22

## [CO343 Operations Research](http://www.imperial.ac.uk/computing/current-students/courses/343)

Understand how to do Simplex and you’d pass the course. - Benson (2022/23 Computing)

Overall enjoyed but least favourite of ones I took this term. Lots of maths and the slides aren’t always helpful. If you struggle with *unnecessary* matrix notation it’ll take you a while to understand the formulas, but the intuitions behind them are well explained. First half (Dario) and second half (Ruth) link well and it feels like a well planned course. Except theres 1 lecture in the middle on game thoery which doesn’t feel like it fits so well. But its super easy and the only coursework is based on it. The coursework is literally the example in the lecture with like 1 number changed so it’s a free 100% (was my year at least) - Alfie (22/23)

Really great course if you’re interested in the maths side of optimisation and are okay with a bit of Linear Algebra and game theory. Giuliano’s stuff seems confusing at first but he’s usually a really nice marker and doesn’t set too hard exam Qs and Ruth is amazingly passionate about the course and teaches her side of things really well. Tiger (2019/2020)

Ruth Misener was easily my favourite lecturer of autumn term. - DH (2019/20 Computing)

One of the most interesting modules I’ve taken (personal opinion). Noteworthy topics include linear optimisation (aside: this is a prereq for CO477 Optimisation and CO422 Computational Finance), game theory and minimax. Applications of knowledge in this course include optimisations in business (which business decision?) , finance (which investment?) and logistics (which route to take and where to build warehouses?). Worth noting that an ex-CEO of Singapore Airlines did a Masters in OR at Imperial (not sure if he applied it for route-planning, which--interestingly--appeared as a past year question in this module).   
First half (Casale) is a tad bit slow (but some parts are heavy), and the second half (Misener) is well-paced and well-taught. I genuinely felt motivated to learn the material, and while there are some proofs to memorise, they aren’t heavy and come naturally once your understanding is sound. Coursework is doable, exam can be difficult (and contain some unseen material) but okay if you do the tutorial sheets (ditto Kelvin).  
For EIE: No extra effort required. Slightly useful to go through some basic material in Logic to make sense of some logical statements (in Integer Programming), but not strictly required. For those considering taking EEE Maths for Signals and Systems: while both modules cover a lot of linear algebra, there isn’t much overlap except for some matrix fundamentals. I did not see taking both having an advantage/disadvantage.  
- LH (2019/20 EIE)

I suck at linear algebra but I sat down and forced myself to understand the maths/symbols in every slide and gave myself enough time to make flashcards and write down all the proofs to memorise. If you have faith in yourself to do that, you'll do well in the exam since the lecturers are good, the questions from tutorial sheets *will likely come up in the exam* and are very useful, and all the techniques are useful and somewhat learn-by-rote. Definitely do if you did decision maths for your A levels. - Kelvin (2019/20)

Like D1 and D2: mostly about Simplex and some other miscellaneous stuff. Probably useful for \business modules, but very tedious, not super interesting, fairly easy CW, but exam is random. Sometimes super easy, sometimes a total brainfuck i.e. 2018 paper. Overall: would recommend as a 3rd/4th module for the term if you can’t choose anything better, fairly straightforward, but not very fun to do. - Dima (2018/19)

Not very enjoyable and not very easy but overall fine and I’d recommend it over other worse courses.

The whole thing is about linear programming problems, the lecturers are good, with plenty of tutorial sheets and resources. I found the exam fairly tricky but with enough work it’ll be fine. - Rohan Pritchard (2018/19)

Easy course and the exam is also fine - Blair (2019/20)

Incredibly dull in my opinion. Coursework marked harshly. Would not do again. - JZ, EIE, 2020/21

One of the easiest modules I’ve taken at my time at imperial. Practice two weeks before the exam so you remember how to multiply fractions and then you’re chilling. Also quite interesting applications, and ruth is an interesting and nice lecturer – Lucas, 2022/23

## [CO333 Robotics](http://www.imperial.ac.uk/computing/current-students/courses/333)

Quite an enjoyable module and generally has pretty high grades. Far easier than any other module I’ve done at Imperial although it requires coordination between a large group to do coursework. The content is very shallow and high-level so don’t expect to learn anything meaningful about Robotics. The practicals are quite fiddly and it can be frustrating to take good measurements in labs but overall I had a good experience. It’s fun to play with Lego and this module is a bit more creative than most – Zaki (22/23)

Taught in second term for us. Good fun and not too hard. Practicals are fun (but make sure you get a good group) and make up 30% of the module. Although the first one is very time consuming tunning parameters. This is by design to teach you how painful it is to blind control a robot with no sensors before teaching you all the better algorithms to use instead, so don’t stress getting your robot super accurate for this one.

Work load is super chill since its 1 hour lecture a week and 3 hours in labs. Although the lectures could definitely be more interesting than they are. Exam this year was harder than past papers also just saying – Alfie (22/23)

Overall the course is really interesting and you get to basically apply the maths you learnt in high school like Pythagoras to actual robots which become your baby for the term. CW does take up a fair bit of time but the material needed for it is given in most of the slides / lectures. The lectures aren’t the most interesting but the practical side is cool and the exam is really easy at the end (he asks the same questions every few years with subtle differences.) Tiger (2019/2020)

19/20 - Most most most important, make sure your robot can properly drive in a straight line at the beginning, or you’ll be in for hours of pain.

19/20 - Annoying and time-consuming coursework, with lots of little magic numbers to play around with. Nice exam though, CW prepares you really well for it imo. If i were to choose again, I would probably still take the course.

2019/20 - CW Percentage upped to 30%. CW starts easy but expect to spend late nights on the final one if you want near full marks. Exam has reasonable questions, do the coursework properly and you'll be prepared.

Agree with 19/20 above me. CW isn’t too hard but time-consuming. Exam-wise, pretty nice. He recycles questions a lot and it’s basically just implementing variations of functions that you’ve done in the CW. Only like 6 lectures of actual content and none of it is very hard - just need to make sure you do the CW properly and exam should go fine. - Avish (2019/20)

Do this course if you want to optimise for good marks! Coursework took a long, long time, but if you get involved in the coursework, take time to understand the slides and remember the few, easy maths derivations there are in the slides, you're guaranteed to do very well. There is only about one hour of lectures per week and the rest is the robotics gorup project. - Kelvin (2019/20)

Well. CW == Exam, so if you are willing to put the hours in, you will do super well. CW itself, however, is rather shit. Teams are not well balanced, tasks are not well balanced, still only 15% for about 30 hours you spend on that CW etc. Content is rather dull, not too difficult, overall would pick as the easier module if nothing else. Maybe 1st or 2nd choice. - Dima (2018/19)

Highly recommend. Course was relatively fun and exam is easy and questions are recycled. However, the courseworks take up a lot of time so manage your time well and choose your groups carefully. (2018/19)

Fun course, concepts are simple and there’s not a lot of content. You’ll just get very frustrated with the hardware breaking all the time 😂 - Tom (2018/19)

Amazing. Best course i’ve done at Imperial so far. Ethan (2019/2020)

Really well taught. Not that much lecture content to go through since most of the content is practicals. Practicals etc generally pretty easy. Taught remotely in 2021 using Lua + CoppeliaSim. Get a good group but also make sure you put in the effort (really useful for prepping for exams) - JZ, EIE, 2020/21

Pretty easy, very well taught. Good programming skills will make the practicals pretty much trivial for you. Lecture content and practicals perfectly prepared me for the exam. No complaints at all, one of the best courses I’ve done so far. Note that I did it remotely with CoppeliaSim (which uses Lua code), so your mileage may vary. - Nat K (20/21)

Hype – Akshat 2021

## [CO332 Advanced Computer Architecture](http://www.imperial.ac.uk/computing/current-students/courses/332)

Really really interesting module, and Paul Kelly is a great lecturer. Fair warning: there’s a lot of content



Just do it, **best module** I've done at Imperial so far. Paul Kelly is an amazing lecturer and is so passionate for the module! - Robbie Buxton 22/23

Ditto – it was great – Rushil Patel 22/23

You will never feel like you understand everything or anything, be prepared to go back and re-read content and do independent reading. Courseworks are interesting and challenging, pay attention to writing style and readable and well labeled graphs.

Exams are a bit of a unknown but you can usually wishy-wash a sort of answer to anything. (2019/20), and the CWs involve doing an investigation and writing it up in a report of around 5 pages in length. The exam involves reading a modern article on computer architecture and being asked questions on it - IMO a very nifty way of examining a module while also keeping the content interesting. Bonus: the content overlaps with Hozlger’s section of Performance Engineering. Would highly recommend ACA, but be careful if you’re planning on taking other high-time-commitment courses as well (e.g. Robotics, Type Systems). - Fawaz (2019/20)

Interesting module but can be very time consuming. Courseworks are not really difficult but will take a while to do well. Paul is a great lecturer and can make the dryest of lecture material interesting. Also he answers everyone's questions and keeps the lectures interactive. Exam requires actual understanding of content but isn't too challenging if you revise properly. - Jamal (2019/20)

Paul Kelly is a good lecturer, except his slides are not understandable without him explaining. He talks about a lot of interesting concepts in architecture, but the coursework took quite a long time. Requires non-trivial amount of work. Exam involves an article to read about a specific given architecture. - Fangyi (2018/19)

Way too much content, really badly explained, unsure what parts of the content we actually need for exam. However, he is relatively nice with his marking in the exam, and will try to give you marks for saying sensible stuff even if not really correct - Ethan (2019/2020)

Lots of content but it’s all cool and some current stuff as well e.g. Spectre/Meltdown. Discussion-based so turn up to lectures, actually really fun. CW is ok, exam is not that hard. - Omar (18/19)

Excellent course and lecturer. Really interesting and engaging if you’re into computer architecture. There’s a lot of content, but the coursework and exam is really fun and Paul challenges you to think in the lectures, coursework and exam. The marking in the coursework and exam is really kind, since they know it’s a hard course - Matthew (2019/2020)

Great lecturer who cares about the students. We get to ask questions and he makes a point to answer everything. Good module if you like Computer Architecture and are genuinely interested in reading research papers or going a little bit over the spectrum of last year’s course. - Maëlle

A must-have if you’re keen on computer architecture. For me, it was definitely the best module I’d taken in term 1 (my other choices were Simulation and Modelling and Intro to ML). Lectures were enjoyable and I really liked the exam format (based on studying a recent research paper) even though it made revising with past papers difficult. - Maks (2020/21)

I think you should be careful about taking this module if you’re **not** interested in computer architecture and you only want easy marks. ACA will definitely not give you easy marks since Paul seems to expect deep and concrete answers in his exams (unfortunately, I realised that only after seeing my results and the general exam feedback). On the other hand, if you’re interested in computer architecture, my previous opinion above still holds: it’s a great module and you should take it! - Maks (post-exam addition, 2020/21)

Honestly best course and best lecturer. Paul Kelly can seamlessly explain any complicated concept, and even though everything was remote, his lectures were great. In terms of the course content: when it comes to coursework, Computing students might enjoy it slightly more than EIE students, as most of it rotates around the high level perspective of computer architecture (in CW1 you optimize a processor’s configuration for a specific program, and in CW2 you optimize a C program for a specific processor). On the theory side (lectures), you touch on many complicated and fundamental computer architecture concepts to design a processor or optimize software when interacting with it, and you can use those in other modules or projects or even low-level / hardware interviews (especially EIE). For the exam, it’s different than the usual, you get a paper on a recent processor to study a month before and most exam questions are about applying the lectures content to that processor (even though the exam went okay-ish, I'd take that module again). A definite go if you are in EIE. EDIT: after the module, lookup pwn and hardware exploits from CTFs, you’ll know the hidden value of this module. I think there should be some merge with netsec in some way! – Jaafar 2020 / 2021

Easily my favourite course from Autumn term. Paul Kelly is really passionate about the subject and always happy to answer questions. Coursework was way too time consuming though imo. Exam was actually quite interesting since it’s based around a research article. Even if you aren’t sure of an answer – if you explain your reasoning, he will generally give credit I think. For revision, a bunch of us got together and answered past paper questions using this year’s article (also, make sure you know the article really well) - JZ, EIE, 2020/21

**Pros:** Contents very interesting and useful (probably the most useful course in term 1). A must-take if you want to do anything related to architecture, high performance computing, OS/Compiler (any low-level stuffs), etc. Paul is very passionate about lecturing and explains most concepts very well. Extremely well-designed exam, which tests your understanding instead of memorization. **Cons:** Lengthy lectures; coursework is not very well designed (you need to spend loads of time but you don’t learn that much); bad slide and no lecture notes, so very difficult to revise without re-watching the lectures. Overall, 8/10. - Kaiyan 21/22

I’m ngl this was probably by far my favourite course from Autumn term, and arguably of my entire Imperial degree experience thus far (although still with time to go). While the material may seem abstract at first, Paul Kelly always manages to find a way to competently and clearly explain it in a way that makes you feel like you always knew the intuition behind it, and always presents real world examples underlying all the concepts we learned. I particularly enjoyed the sidechannels section of the course, as it overlapped CompArch with Security giving even more practical flavour. The coursework took a fair bit of time though this year, requiring quite a bit of exploration, you have two tasks which vary little in topic year on year but vary on code. The first task requires you to modify a simulator processor’s configuration (cache, exec units etc), in order to run a benchmark program with minimal energy consumption, the second task requires you to optimize code to hasten the exec time of a given piece of code, these courseworks took quite a lot of time investment but for rather low returns (only 10% each). Exam is pretty interesting in that you get a recent article on a new-ish processor specification about a month in advance to pore over, and you get asked questions based on this tying into the course, some may like it and others may not, but imo it was a pretty cool way to assess our understanding. The tutorials help reinforce understanding and have good discussion, size isn’t overtly large so everyone more or less gets a say, and when you’re lacking from knowledge from a given topic, the course textbook compensates for this and then some, it’s well supplied and well rounded. Honestly, if you’re EIE and you loved first year DECA, and you loved second year ISA&C, this module is an absolute no-brainer, take it. - Aryan 2021/22.

As others said, the CW is very lengthy so start your CW early! You have 2 separate reports to be completed with each having 2 weeks between the day they’re given and their deadline so make sure you put like 2-3 hours in per day straight from day 1 otherwise you end up having to cram towards the end like I did which resulted in me falling behind on content from other modules. - Nelson, 2021/22

It’s certainly an interesting course, but there is quite a lot of diverse content. The exam is also quite difficult to prepare for, since this isn’t really a standardised course – you're answering questions based on a real-world paper / article about a processor. - Nat K (20/21)

Course content is super interesting. However, coursework was frustrating, it was like asking you to write cos(x) as a sum of some quotient number without teaching you the concept called Taylor expansion. There is not much material online for this nor do they provide guidance during Q&A sessions(coursework gets updated each year, I heard this year’s coursework is more relevant to the course content 22/23 ). Best way to learn the course material for me is to get a transcript of the lectures, memorise the key concepts, then do the past papers, you should be ready for the real exam at this point. I would recommend this course, a real eye-opener. Yi Lai 21/22

## [CO337 Simulation & Modelling](http://www.imperial.ac.uk/computing/current-students/courses/337)

**Exam:** Speaking to everyone after the exam was a sight to see, everyone thought they did badly. As for the content, it was pretty cool just pure aids to revise. You have to solve like 4 questions but in the 2h its kinda doable.

**Content:** For Tony, the first lecture is the hardest, with the rest being just maths stuff. For his part, you just need a cheat sheet with the formulas and some intuition from the Probability and Statistics module last year. Gulliano really changes the game with his part. Its essentially following graphs with probability and rates but on crack. Formulas get sticky very quickly but overall his part of the exam was probably a bit better than tony’s.

**CW:** coursework was chill, and if you managed to do it then you’ve got one of the four questions in the exam on lock.

* Anton 22/23

Mixed feelings here, probably because I picked this module as a filler rather than something I really wanted to study. Giuliano's part was more boring for me than Tony’s (= Tony was more interesting) and I seemed to struggle more with the Giuliano’s stuff, especially in the exam. - Maks (2020/21)

Overall, fairly interesting course with a terrible coursework. Tony’s half is, as expected, very engaging and feels a bit more practical/hands on given the content is more about simulating. Guiliano’s half is more theoretical and can definitely start to feel a bit too separate at times, but an interesting way of looking at the topic. Initially, can be a bit overwhelming with the different formulae, but once you get them together and see how they all relate, it can become fairly straightforward. The coursework however was **horrible** in my experience from Tony’s side. It was some fairly straightforward maths and “proofs” for the first half, and then half of the marks was weighted towards an open-ended experimentation section that gave little to no guidance on the expectations of what should be looked into. Was also limited to a page limit due to previous students writing too much for the research part, which then impacted how much we could actually delve into it and write about. Had to email in after mark scheme/feedback was released to query marks which were missed. - Prab (2020/21)

Tony was great, found the course interesting but felt niche. Giuliano's side of the course feels very separate from Tony's not sure how much collaboration went on. - DH 19/20

Tony’s half is more interesting but he whizzes through it pretty quick, Giuliano’s is more difficult and he can be quite monotonous. Some A-Level maths (graph transformations, differentiation) involved as well. Overall, quite an engaging course. - AN 2019/20

19/20 - Tony bae as usual. Interesting course, I really enjoyed it. Didnt go to any lectures from Giuliano, but he is great on Panopto x2. Memorising formulas is a bit boring but you have to do some of it because you wont have time to deduce everything in the exam! Still, glad I took the course. (Caz)

Overall pretty great course, mainly involves memorising equations and some fairly easy maths (read as: take this if you’re JMC). The exam is fairly standard every year. Giuliano can put you to sleep though. - Fawaz (2019/20)

For some reason, I could take this in 4th year. Tony’s an outstanding lecturer as always; the course is a lot of noting down formulae and regurgitating them for tutorials/coursework/exam, which worked a treat for open-book exams. Looking at others’ opinions for previous years here, it seems like Guiliano’s half of the course now follows on more naturally from where Tony leaves off. The coursework was half programming (in any language you want; it wasn’t too taxing), and half applying formulae like mentioned before. - Nat K (21/22)

## [~~CO349 Information and Coding Theory~~](http://www.imperial.ac.uk/computing/current-students/courses/349)

This course no longer exists as of 2019/20 as Mahdi is no longer teaching at Imperial.

I really enjoyed this course, this provided a lot of theory if you enjoy that to balance the other very practical subjects. I’d say my favorite course this term, a bit like what I imagined theoretical computer science to be like back in High school. It didn’t have any practical applications but IMO really fun to learn, a bit like brain puzzles. - Matej (2018/19)

Oh shit here we go again. This is a choice for true masochists among you, it is not panoptoed. Has an in class exam as CW (that is useful as a learning activity, but so poorly done. I already complained, so Tony promised to change the way they do the in class assessment next year) which could fuck you up, but I think is a great revision method. There is a fuck ton of content, becomes very chaotic towards the end: some stuff you literally have to memorize and there is a lot of it. Overall, the exam was not pleasant, the content was very spread out, there was a lot of it, but the content itself was interesting to learn. Quite mathsy, so comptech is a massive plus (I didnt do it and still did ok knowing that i fully missed an entire qn). Was my 2nd choice, but would not recommend unless you have done comptech; are genuinely interested in the content, go to lectures and put the hours in. Of all 4 that I took, this consumed 40% of my work time, but I enjoyed doing it nonetheless. - Dima (2018/19)

First half is do-able, because the exam is easy, but no panopto. Second half is not do-able, exam is difficult, slides not too helpful. Requires maths and matrices and imagination. - Fangyi (2018/19)

P.S. You cannot avoid “tensors” in Herbert’s courses.

## [CO382 Type Systems for Programming Languages](http://www.imperial.ac.uk/computing/current-students/courses/382)

Overall, I am happy I took this module. The notes are quite good, while the lectures can be a bit confusing the first time you are listening. The exam content itself is not too challenging as the proofs are omitted from the exam and the questions are expected/repeated (even for the TRAs). I believe it is the module that requires the least amount of effort in the 3rd year. Also, as it is with Bakel, you must be very precise with what you are writing. If you are, you can achieve quite a high grade at this module. However, some of my friends received unexpectedly poor marks from this module, so be careful. - Roko (2020/21)

Agree with other comments that this is a good module, and that exam is very doable (if you put in effort and time to learn definitions, which are then used in derivations). While proofs are difficult and are bulk of the lectures to motivate applications and derivations, fortunately they are unexamined. Steffen is an amazing lecturer who loves the content and does his best to motivate you to appreciate type systems. While this course mostly deals with theoretical CS, its applications are not too far-fetched (e.g. implementing a type inference system (duh)). This course also serves as an ad for Steffen’s research topics, which he encourages students to consider by the end of the module.  
For EIE: DoC students will have an advantage of knowing lambda calculus and functional programming (though these aren’t strict prereqs and he will go through some relevant content). I found it helpful to go through Dr. John Wickerson’s lambda calculus lecture PDF: <https://johnwickerson.github.io/talks/modcomp_lambda_2015.pdf>, as well as a Haskell crash course: <https://www.youtube.com/watch?v=02_H3LjqMr8>). It is possible to score well without revising these fundamentals though. Also, this course is (very slightly) helpful if you proceed to take HLP in Spring EEE, which uses F# (F# implements a Hindley-Milner type system, briefly covered in this module)  
- LH (2019/20 EIE)

Nice module. If you took Steffen's Discrete Structures lecture in first year, you'll be as, if not more, confused during his lectures in this year. However, the coursework is relevant and you can get by by printing the notes at the start of the term and understanding them + going through their questions in your own time. (Notes have more detail than the lectures, so bear that in mind and don't study unnecessarily.) Exam is pretty much the same as every year -- you can get by well if you do all the past papers. Steffen seems to use Panopto nowadays (at least, in our lecture). - Kelvin (2019/20)

Pretty nice module overall. Coursework is relevant and almost all the questions are similar to the ones you'll be asked in the exam. The exam doesn't contain any proofs bu but coursework might. The exam just uses the questions from vious papers and the exercises. Notes can be decent sometimes but they have a lot of detail which isn’t necessary for coursework or exams and might end up confusing you. Steffen really loves the content and you can tell but he also understands that it's not as easy for us and will explain stuff if you ask. - Jamal (2019/20)

You will almost certainly be confused as hell during the lectures and Steffan ends up spending most of the time going through proofs that are unexamined and just confusing the heck out of you.

However, the papers are basically the same every year so you can very easily prep for them and get good marks even if you don’t understand what’s going on in the course most of the time. The stuff that I did understand was interesting, albeit a pain to understand. - Sam Trew (19/20)

The content is actually pretty interesting once you get around to understanding it, of course it's taught by Steffan so his notes are usually much more detailed and succinct than his lectures. He goes through a fair few examples from the notes in the lectures, and even points out quicker ways to write out answers. Even though I enjoyed this module, it was one of my worst modules as he can be very strict when it comes to marking the exam papers. - Anindita Ghosh (19/20)

I did not enjoy this course, the lectures, coursework or exam. My advice is to only take this course if you know you will be interested. Yes, the exam (and seemingly coursework) tend to be repeats year on year, but as a result the scaling is absolutely horrible. Make one small mistake and game over. This is surely not a problem if you know you are interested in the content, but I’d really advise against taking it just because it’s seen as easy. Many students this year received much lower results than expected on this course - Matthew (2019/2020)

Easy exam (he takes questions from tutorials and past papers) If you’re into programming language, this is one of those “must-take” ones. The course involves some proofs and derivations, similar to those in Models of Computation. Steffen uses whiteboard and jumps through notes, but things are reasonably organised. Notes are well-written and self-contained. - Fangyi (2018/19)

Probably the only course useful if you’re into theory. Exam is very predictable (80% of the questions are the same every year) and the coursework is very doable, but there is no panopto >:( (I only attended 4 lectures in total and started learning what a type is 3 days before the exam and still did fine, so good course lol) The background required is just models of computation in the second year - David Ang (2018/19)

If you enjoyed Models of Computation, take this. It’s theoretical and proofs require some concentration, but once the ideas behind each presented type system click it becomes very easy. Exam was recycled from past papers so it was pretty straightforward as well. It’s also fun if you’re into strongly typed programming languages or programming languages with rich type systems - Radu Lacraru (2018/19)

## [EE2-13 Computer Architecture](http://www.imperial.ac.uk/computing/current-students/courses/EE2-13)

(N.B. past paper solutions are official and be had from either <http://www.padkay.co.uk/ee2> or <https://sites.google.com/view/iclep/year-2> )

## [CO572 Advanced Databases](https://www.imperial.ac.uk/computing/current-students/courses/572/)

N.B- This module is now called Data Processing Systems, taught entirely by Holgoat Pirk. The module was very well taught and the exam was fair- coursework was interesting, C++ skills are handy here. Content is manageable, 8x ~2 hour lectures, would recommend – Miles 22/23

Quite hard and confusing (for me). It is getting harder year by year, maybe because it becomes open book these two years. Do take a look at the past papers for 2019-2020 and 2020-2021, it removes almost all simple and standard questions compared to previous years. The slides do not show every detail, and you need to search/ask by yourself. The tutorial sheets and slides only contain easy questions and provide simple examples, while the exam is way harder and complex. It requires you to fully understand every aspect of the material. If you go and check the answers for previous years, some questions may contain 4-5 alternative answers, and no one know which is the correct one (for most of the time I would come up with a 6th answer). Also, for EIE, do teach yourself SQL before you choose this module, though it does not mention so. (2021/22)

I’ve found the stuff I learnt in this course the most useful whilst doing my Industrial placement and the knowledge from it was also really helpful in preparing for interviews. Holger is slightly unorganised and his stuff is difficult to prepare from. Expect his exam Qs to be different each year, but on peter’s side he gives you loads of mock questions and then in the exam the Qs will be basically the same but with slight differences. Tiger (2019/2020)

Incredibly useful course, Holger has definitely upped his game and will probably continue to do so. Peter's section has plenty of resources and the exam is always variations on the same questions and Holger is a really fun lecturer. (2019-2020)

Really useful for learning more about algorithms/data structures and how to actually use them in the real world (even if the focus is on databases). Holger is very enthusiastic but the second part of -the course is very different and really focuses on real-world applications (like distributed databases) which imo was less interesting. - Sam Trew (19/20)

Really fun course and practical coursework that lets you play around with the areas that interest you. Holger’s part in particular is really fun, the second coursework is exacting but on the easier side. The exam is difficult, especially in the time available, but easy to prepare for - Matthew (2019/2020)

Don’t do it unless the course changes. Shame, because the course itself was relatively well taught (at least the Peter McBrien’s half was) and almost interesting, but the exam was a complete disaster. You aren’t provided with any tutorial questions for the first half of the course and barely any of the past paper questions were still relevant. The exam went really badly for most people but if the course is improved based on SOLE feedback then maybe it’ll be fine. (2018/19)

I think this is a great course, Holger’s section lacked resources but I do think he’s working at changing this (he taught Performance Engineering in the spring and finally upped his game). It’s mega helpful stuff too, applied plenty of this knowledge at my placement this year. - Rohan Pritchard (2018/19)

Highly enjoyed this course. Extremely useful, and Holger is great fun (and very helpful). - Tom (2018/19)

It’s a good course with a lot of interesting content and Holger’s coursework was pretty cool. Didn't think Holger’s calculation questions were covered enough in the course given how much of the exam they take up. Peter’s side was more ambiguous, took a lot of time to get through everything in his part of the exam. - Alex, 2021/22

# Spring

## [CO304 Logic-Based Learning](http://www.imperial.ac.uk/computing/current-students/courses/304)

Quite an interesting module, very different to most other modules because it is far more theoretical. Taking Symbolic Reasoning beforehand helped quite a lot conceptually. This module has lots of content and is more challenging than most options in my opinion. The teaching was engaging but it was hard to follow sometimes: you really need to stay up to date on content because everything builds on each other. The exam is quite formulaic in manually executing algorithms so there shouldn’t be any major surprises. - Zaki (22/23)

Found the module interesting, in short: teaching a computer how to think like a human. Exam was very preparable (just spam past papers, content doesn’t change much year on year). Didn’t do AI the year before but found catching up doable. Alessandra is a very engaging lecturer and approachable – Andrew (JMC 21/22)

Regret taking this course. Probably one of the least enjoyable modules I’ve done. LBL is basically just a bunch of lengthy and dull algorithms you have to follow and ‘logic-based’ isn’t as logic-based as I expected. – Ezra (JMC 21/22)

One of my favourite modules this year. Alessandra taught the entire course this year, and even remotely, she managed to make the content super engaging. I didn’t take Intro to AI last year (and know quite a few others who didn’t), so we had to do some catching up at the start, but Alessandra covers this all in a recap lecture at the start and was very happy to answer questions from those of us who were unfamiliar with it. The resources are excellent – there are comprehensive slides, lecture notes and tutorials for every week of the course. CW was a bit tricky but doable. Exam was quite similar to past papers, even when open-book.

Although the applications of this course are for machine learning, this course isn’t really anything like the other ML modules - it’s all based in Logic as the title of the course suggests. Follows on nicely from first-year Logic and Prolog from second year.

- Rini (2020/21)

I agree with the comment above, I really enjoyed the module. As someone who completed Intro to AI in the second year, it was less challenging as I had no catching up to do. However, a majority have not completed the module and Alessadra put in a lot of effort to have everyone learn the basics. She even organized an additional lecture to go over the questions from the recap lecture again. - Roko (2020/21)

Great course, Alessandra is a great lecturer, very enthusiastic and willing to help. However, her lecturers aren’t the clearest and if you didn’t take Intro to AI last year then it will be a LOT of catching up at the start (she’s willing to put on extra lectures and help you though). Mark is a great lecturer and has lots of tutorials which he intersperses into the lectures which is really good with understanding. The tutorials & solution for the second half (Mark’s) are great, and he responds on piazza immediately, Alessandra’s half is a bit more hit and miss though. Same with CW - Ale’s is a bit of guess work but Mark’s is extremely easy. Exam is very standard, do two papers and you’ve basically done them all because there are 2 main formats they follow. - Jack Morrison (19/20)

Alessandra is a really good teacher, even if she does speed through slides and talking. She says she doesn't panopto, but can easily be convinced otherwise (get a couple JMCs to complain about maths clashes and you're in). Mark is nice, but not quite as interesting as Alessandra. Lots of tutorials, like half the time is tutorials. Its tempting to skip them, but honestly skip the lectures, go to tutorials since it's so much more helpful. Easy exam, exact same as past papers/tutorials and just algorithm recmembering - Emma (JMC one, (2018/19))

## [CO317 Graphics](http://www.imperial.ac.uk/computing/current-students/courses/317)

Good lecturer and very interesting course. Courseworks 1 and 2 are nice and shouldn’t take more than a few hours each. Last CW takes a disproportionately longer amount of time - I probably spent a total of like 20-25 hours on it. They are all quite fun so could be worse. Maths wasn’t too difficult (I am a JMC but I don’t feel like it makes a big difference) and like posters below said, similar to FP3 vector/matrix stuff. The hard thing about this course ended up being understanding the wording of the exam questions - definitely useful to look at past papers early on. Would recommend if you can dedicate enough time during revision period. - Avish (2019/20)

It was a very fun module, but the content does take a while to go through. We were lucky enough to have it open book due to the pandemic, hence I skipped learning the tiny details of each lecture, but some did end up appearing in the final paper. Past papers are your best bet as he usually repeats some questions. Coursework takes a very long time to wrap your head around - Anindita (19/20)

Do not. Simply don’t. Do not do this to yourself. You have to be very good at comptech and learn a fucking ton of material. I easily spent 60-70% of my term 2 revision on this and I think I failed. Fuck this course. - Dima (2018/19)

“Graphics is fucking aids, your girlfriend will leave you, your life will leave you” - Oscar Zhang (2018/19)

It fucked me too, coursework was fun, exam was a shit show (and I think it was even a fairly easy paper this year… still fucked me) - Rohan Pritchard (2018/19)

I can confirm the above ^. I took this course thinking it was going to be easy, even though I was not interested in Graphics. Coursework was fun, but I had trouble with the theory and the exam. Only take if passionate about Graphics - Radu Lacraru (2018/19)

It wasn't that bad… a lot of FP3 Edexcel vectors and a lot to memorize but nothing too mathematically challenging. - Isaac Hutt (2018/19)

Great course and good lecturer, but tutorials weren’t that relevant to the exam and courseworks take several days to complete for me. Basic linear algebra only in the beginning, so required background is first year/second year linear algebra, and a lot of cool concepts were introduced near the end. Exam was fun, and sort of predictable. - David Ang (2018/19)

Good lecturer, CW is fun, not hard and marked leniently. Good content, physics stuff isn’t examinable. Will need to memorise lots of stuff for the exam. The maths is easy, even if you’re not JMC. - Omar (19/20)

Do an online course e.g. CMU15/462, 6.837 instead. This is an introductory level course, there’s a long way ahead to be “good” at graphics, but they made it hard to understand everything, so you ended up spending ages online finding the definiens of every unexplained/not well explained definiendum/concept/keyword in the lecture slides. Don’t waste your time. Useful link if you do want to do this course: https://www.khronos.org/opengl/wiki/ Yi Lai 21~22

## [CO318 Custom Computing](http://www.imperial.ac.uk/computing/current-students/courses/318)

This module is very niche. I loved it but you need to enjoy the hardware stuff, from logic gates upwards. If you enjoyed Comp systems and Comp architecture from first year id say go for it, although it feels a bit easier than architecture was (but then again I kinda fucked first year). Probably my favourite module this year but I understand not everyone will agree.

First half (Wayne Luk), the content is confusing at first but its pretty much 2 hours of lecture and 2 hours of tutorial a week so if you go to every tutorial eventually it clicks and then its super easy from then on. Uses ruby (not that ruby, kinda like haskell) to design block diagrams for hardware computers. Don’t worry about building a whole cpu, the whole point of custom computing is to build stuff that’s more specialised than that, and hence better at its job. Nothing as complicated as designing a whole cpu like was done in comp architecture.

Second half (Tim Toddman) is way easier than the first with the same split of lectures and tutorials so plenty of time to make sure you understand everything and work through examples. Designing custom computers in a different way using data streams. Code is written in java to build a dataflow graph which can then be sent to an FPGA to implement your hardware design. Although you don’t actually have to go to all that effort, its easy enough to see if your code is right without ever simulating it.

Coursework is pretty chill and fairly representative of what the exam will be like but with 3 questions instead of 2.

Also, the first 2 lectures are weird and boring. Its kinda about FPGAs and their use in industry but honestly I didn’t enjoy it and its not represenative of what the rest of the course will be like (also not sure if its even examinable since it hasn’t been in any of the past papers but isnt explicitly stated as non examinable)

TL DR: Favourite module this year but its niche, you may not agree – Alfie (22/23)

Relatively interesting content but the lecturers are quite dull. Not a bad thing to have literacy on though – Lucas, 22/23

**Updated after completing the module and taking an exam:**

Custom Computing feels very niche, the number of students attending live meetings and tutorials was really small (~5 undergrads and postgrads) compared to other modules I took in term 2. However, niche =/= bad! I would call it a hidden gem for people who are interested in FPGAs and thoroughly enjoyed learning hardware / digital circuits and computer architecture in year 1 (regardless of who taught them), advanced computer architecture in year 3 and maybe code generation in compilers in year 2.

The first part is about Ruby, a functional hardware description language. It’s a research language not widely known in the FPGA field, but it teaches you to think how to design efficient and correct designs, like Haskell does in the software engineering world. Ruby is not worth learning alone (you’ll definitely not make a career only with it), but it’s a very nice thing to know alongside other languages used in FPGAs like Verilog, VHDL or high-level synthesis. You may think at this point that this module is useless because it apparently teaches only Ruby: wrong! It’s just the first part.

The second part of the course is about MaxJ, a language using Java to describe FPGA designs, something like high-level synthesis probably. Unlike Ruby, MaxJ is a bit more popular and used commercially and I hope to use it as a bridge to other high-level synthesis languages like Vivado HLS (C++ for Xilinx FPGAs). This part also teaches you concepts which I’ll probably find useful in my FPGA-related placement, for example performance metrics in FPGAs.

Overall, both parts are well-connected to each other and it was only after completing the module and practicing for the exam that I truly appreciated stuff taught in Custom Computing. I feel it’s going to be a springboard for my further study about FPGAs. Beware though that the module might seem confusing or useless to you at the beginning (because of the first impression you may have about Ruby). I was in this exact situation and I was close to dropping Custom Computing because of that (I’m glad I didn’t).

When it comes to exams, they are rather predictable and some questions tend to be recycled (even in the online year), so when you go through a few recent past papers, you should be ready to take the exam.

**December 2021 addition (after getting the exam results):**

The last paragraph about how to prepare for exams still holds, this is exactly what I did for revision and I got 80%+ for my exam answers.

- Maks (2020/21)

* This is a tough module, especially Wayne’s half as it took me quite a lot of practice to get the key Ruby-related concepts, and Rebecca can be very fiddly to work with (they should improve the slides on that aspect, because the slides speak Ruby and not everything in Ruby works in Rebecca). It took me quite a while to start getting the key ideas of his half.
* However, I’ve had one of the best support *ever* from this module, and it mostly comes down to Tim Todman, who is a *star* and worthy of an award. Passionate, helpful and active on EdSTEM (he’s fully familiar with Wayne’s half of the course) and in-person when he’s teaching the second half (more of a third) of the course. While Wayne is similarly helpful in-person when he teaches his half, he goes to sleep on EdSTEM and similarly the TAs normally show up only during the tutorials.
* Speaking of which, the tutorials are helpful because both Wayne and Tim (during their sections) will come to you and happily discuss (one-to-one) and clarify any of the difficulties or concepts in the module.
* The coursework is not lengthy but as I said earlier, it can take a lot of practice and time to get the concepts right, and Rebecca can be a pain at times. Again, Tim’s there to help. Marking was on the lenient side and was done in about a week.
* The exam is moderate – not as easy as Maks may suggest but indeed preparing from past papers help and I personally found Tim’s section easier to study than Wayne’s. I found myself thinking more than “cranking” though in the exam as with some other modules.
* TLDR; as Maks says. Ruby is hard but don’t let that make you drop the module. This may be relatively less popular, but that should not stop you either (and is actually kind of an advantage). Very interesting material that’s taught reasonably well and there’s help if you need it. After all, this is one of the modules you’re unlikely to find elsewhere.

-- Leaderboard, 2021/22

First half was quite boring imo, honestly I didn’t put in enough time to learn Ruby properly (had a lot going on), lectures I didn’t find that engaging. The problem sheets are definitely worth doing. Todman is super engaged with the course and really responsive on EdStem. His half is easier than the first half imo. JZ, E

As an EIE student, this was a really interesting course for me as it ties in with quite a few other modules like Advanced Computer Architecture, Digital Design, Information Processing. Like others have said, Ruby is not the most intuitive language and there aren’t many resources on it online (not to be confused with the more popular interpreted language),

## [CO316 Computer Vision](http://www.imperial.ac.uk/computing/current-students/courses/316)

Module has a lot of useful and relevant content which is worth learning. There is not insignificant overlap with Intro to ML so it reinforces concepts and adds some practice here with use of Python for ML. At times, the module felt a little repetitive where you learn different methods for a task over and over. Not particularly hard but I found the volume of content a little overwhelming towards the end. Lectures were generally quite unengaging but the resources for the module aren’t bad. Pretty standard module choice and I don’t think you can go too wrong taking this. - Zaki (22/23)

Well-taught module by Wenjia and the course content is very useful - goes through \*somewhat\* outdated algorithms like SIFT, but the fundamentals / ideas behind these algorithms are very applicable to state-of-the-art algorithms. No mention of Vision Transformers this year. Overall, fairly easy course, definitely take if you are into AI/ML - Freddy 22/23

Not sure if I’d recommend or not. Lets just say don’t get baited. The first few weeks are very easy but still quite interesting. **This is not representative of the whole course**. There is a sudden difficulty spike. Interesting module at a high level but going into some of the details it gets a bit confusing and loses interest. By the half way point the module is just intro to ML 2 electric boogaloo, but with worse explanations and more confusing. Would recommend doing the tutorials, I didn’t and suffered revising before the exams. The tutorials are pretty similar to questions that could come up in the exam **and theyre not all taught in lectures.** So you may find yourself suddenly learning extra stuff before the exam. Cheat sheet came in clutch.

Courseworks are pretty ok if youre up to date with lectures. One asks you to impleent some image filtering algorithms from lectures. The other asks you to implement a convolutional neural network of a specific architecture which is covered in the lectures. Although my year there was a lot of variance of marks depending on which TA marked you. Me and a friend had the exact same answers (despite variable names) to 2 questions. They got 24/24 and I got 8/24 for these questions. Pending a response about this :(

Least favourite module this term but its ok, just not what the first few weeks advertise with respect to difficulty – Alfie (22/23)

Really useful course for establishing a baseline in computer vision, goes well with Intro to ML and has quite an easy CW and Exam. The lecturer (Wenja Bai) is really nice (but not the most interesting). however he does give a revision lecture at the end where he goes over pretty much exactly what’s going to be in the exam. Tiger (2019/2020)

CW was pretty easy and the course is well-structured. The examx contents are all covered in the slides and tutorials but definitely harder than CW. The lecturer Wenjia replied Piazza not frequently. - Blair(2019/20)

Would recommend it. I didn’t find the lectures very fun (even boring) but the content of the course is quite interesting when you revise it in the end. The lectures go in depth into mathematical proofs but most of them are not examinable and the exam turned out to be very easy (questions similar and sometimes same as the ones in the tutorials) . Courseworks take very few hours as well (1st one is on filters and 2nd one on Convolutional Neural Nets, a bit disappointed that we didn’t cover something else practically). In the end, you get a good understanding of most Computer Vision methods and challenges. Lots of content is also redundant with Intro to ML but actually explained better in Computer Vision - Theo (2019/2020)

CW was relatively easy, exam (\*seemed\*) pretty easy too. I found the content very interesting and goes very well with the other courses I did: Graphics and Intro to ML. Reasonable amount of content so would recommend starting earlyish for this. Wenjia started teaching the course two years ago so we only had a few of his tutorials and one past paper but hopefully there will be more practice for future years. Would recommend. - Avish (2019/20)

Loved it, 12/10. Wenjia really holds your hand through the course and definitely makes it clear when something is examinable and w hen it isn't. CW1 took like 2 hours, but CW2 took significantly longer, something to keep in mind as the 2nd half of the term tends to be assessment heavy. Tutorial sheets usually contain some variation of his past papers, so worth sticking around for that. There's a few mathematical bits but it's mostly application + describing certain processes. You also finish by week 9 so you get an extra week to start revising for the module - Andy (19/20)

Pretty interesting course and you learn loads about computer vision and some ML stuff to go along with it. However, it is very content heavy and so can be harder to properly revise for. The CW is almost a guaranteed 90-100% and is a nice intro into using python libraries to do everything for you lol. I didn’t find the exam that easy but if you revise properly you should be okay - Sam Trew (19/20)

Courseworks were quite short and fun. Exam was predictable and on the easier side. I found it relatively easy even after switching halfway through the term - Matthew (2019/2020)

Totally recommend: enjoyed a lot, easy CW, useful for intro to ML, not too difficult. Exam was quite easy (but scaling omg kill me), CW took me a total of 2-3 hours for both parts with little to no python knowledge. Lecturer isn’t super interesting, but the material is good and you can do well without going to any lecture. Panoptoed. My solid first choice for term 1. - Dima (2018/19)

Highly regret not taking. Sounded fun and exam was apparently easy. If I were to redo third year I’d definitely take this (instead of Advanced Databases 😡). (2018/19)

Very easy but apparently will get harder for you guys.. - Tom (2018/19)

## [CO322 Communication Computer Science in Schools](http://www.imperial.ac.uk/computing/current-students/courses/322)

As Jack said below it’s not an easy course. Although, it is a really rewarding and fun one. Beware, sometimes it may not turn out as you’d expect; I got matched up teaching computing to a class of 30 5 year olds and I think that limited how easy it was to get a good mark as I wasn’t able to teach them python or anything. However, it was super rewarding to be able to help out a school without a computing department and inspire some young children to be interested in Computing. Tiger (2019/2020)

If you want to do this save it for 4th year so it’s worth more of your degree (BIG BRAIN)

I took this is third year instead of fourth because I wanted to take more DoC modules in fourth year. It’s not an easy course like people say. We’ve not had the marks back yet so I’m not 100% on how it went, but you need to invest a LOT of time, I’d say I invested more time on this course than any other in 2nd term (maybe 5/6hrs a week making materials, then 10 hrs a week travelling and teaching, I was in Upminster which is a good few hours of travelling). You can rank your school choices and then choose a day/time with the school to teach, but they usually already have something in mind. You need to do a lot of chasing up with the teachers to get them to give you feedback etc., so if you’re not very outgoing it may be a struggle, especially with controlling the class (I had year 10s and they were a great laugh, but it took creative thinking to get them to actually work). It’s a great fun course if you want to have a go at teaching and see if it’s for you (I can confirm it’s not for me ngl), but don’t just write it off as an easy module, because it’s defo not. - Jack Morrison (19/20)

I really recommend this course if you don’t hate children and want a change from info-heavy exam modules. I was able to volunteer at my old secondary school, which was a nice, rewarding experience. This module requires you to be very organised as you must coordinate effectively with your selected school and prepare at least 8 lessons. The coursework is not as intense as other coursework-only modules as you have the spring break to complete the main 3000-word report, but the presentation and other pieces of coursework are due during exam week so please don’t leave those to the last minute! - Beauty (22/23)

## [CO331 Network and Web Security](http://www.imperial.ac.uk/computing/current-students/courses/331)

In short: the module content is really good and relevant, but the exam is not so good unless you treat preparing for it like preparing for math olympiads at high school and therefore attempt **a lot** of CTF competitions over longer time. This is especially important as the lecturer gives hardly any revision materials that are actually useful (no access to VMs from past papers; no access to prepared websites from past papers; this year’s website with flags was entirely deactivated shortly after the exam, so people who failed to find flags didn’t even have time to figure out what they should have done during the exam; most tutorials covered security topics in isolation). - Maks (2020/21)

Would recommend it. Content is very interesting and fun, lots of practical labs (although you can find plenty of better and free labs online). Lectures are a bit boring though and Maffeis doesn’t panopto which is a real drawback (you should consider recording it for revisions). Exam is really hard if you don’t have previous experience, practice CTF heavily. - Theo (2019/2020)

A very fun course. I didn’t think I was into security going into the course, I came out realising I was. The coursework and exam is challenging but doable, and scaling helps you out if it’s too hard. Sometimes the lectures were a little slow, but the practical components of the course really make up for it - Matthew (2019/2020)

The stuff you learn is pretty useful and interesting and that’s just if you read the slides. Note: the lectures aren’t recorded and you will miss out a lot by not attending and so may not be for everyone’s learning style. Due to pandemic we didn’t have the same practical stuff so can’t compare CW and exam but the CW had some nifty and interesting things about it. Would recommend at least reading over the notes even if you don’t take it (They’re not long) - Sam Trew (19/20)

Simple: take it. Not much to say, best course I ever took, even though the exam is done differently, I would totally do it again. -Dima (2018/19)

Really Great exam format, actually had fun for the majority of the exam, if you have some background in it the content isn't that difficult and can be learnt in a weekend. Practice takes a bit more time though - Isaac Hutt (2018/19)

Probably one of the most fun exams - it’s a CTF with lots of shit jokes, and plenty of exercises to practice. Very useful for industry too. - Tom (2018/19)

Overall, I found that this course was the most interesting of the spring ones I did (although I am EIE, so the EE modules leave a lot to be desired in most cases) - the lectures were genuinely interesting with weekly Q&A sessions to help top up understanding if anything was missed – the tutorials/labs were also well written, allowing you to push yourself and learn new things without being hand-holdy, which helps you break out and learn imo. It was cool to try out all the different sorts of attacks and learn new niches or tricks to help with a certain exercise and the Ed discussion page was active so any questions were answered reasonably.

The coursework involves (bear in mind this was the last year of remote, so I am not sure what 22/23 will hold), being given a piece of literature on a specific security topic covered in lectures asking you questions on it and to test your understanding, the other questions rely on you finding particular holes or exploits in a snippet of code provided. The examination is more of the latter, with you being given a few ‘black box’ exercises here and there along with relevant understanding questions, if you’ve had CTF experience before you’ll love it – and it breaks out from typical exam formats – these ones can be a bit more involved but tbh aren’t that bad if you did the labs and understood the lectures – I think it was overall a nice module and would take it again, definitely not as overbearing as other modules can be. - Aryan (2021/22).

It seems like it is (Networks) + (Web Security) but is actually (Networks + Web) Security. Overall, a great module but requires you to really go through the tutorials every week in depth. It also uses PHP and JavaScript in the later parts which are annoying to deal with. - Pranav (21/22)

## [CO338 Pervasive Computing](http://www.imperial.ac.uk/computing/current-students/courses/338)

Great lecturers that were very enthusiastic about the module. The course provides great background on Systems/ Networks and the kind of considerations engineers consider desiging networks/ systems. If you are interested in Systems Architecting / Design or anything about Systems, definitely take a look! However, the coursework took an insane amount of time for the 2020/21 batch. In total ~100 hours were spent on the cw, split unevenly across 2 parts (Should be around 45 hours according to the ECTS mark of the coursework). This was because the coursework was more a research project that had a vague specification. However, the core part of the coursework involved looking at research papers and writing C code for their implementations on Contiki and ran on the Cooja simulator, which was very interesting – Terence 2020/21

I also dropped this course halfway through, since I was not on top of the courseworks. However, I regret it, since it was a very fun course with great lectures and enjoyable (if time-consuming) courseworks. I enjoyed the very practical focus of the course, and since the intake was so small for our year, we seemed to have almost as many helpers as students! I’d call this one a hidden gem - Matthew (2019/2020)

Ended up dropping. Very interesting content, multi-discipline, ended up just going to lectures and not taking the exam. 33% CW, but it will eat ALL of your time. It is borderline impossible to pull it off in the last week of term. Very difficult, super time consuming, only take if you can afford to do less well and want to learn something super super useful. -Dima (2018/19)

Courseworks are very annoying to do and are very time consuming. Need to write a report for both and work with a clunky simulator for the first one and unreliable hardware for the second. The content is mostly very boring but can be useful for IOT/low level protocol stuff. The first coursework especially is badly designed and vague when it comes to details. You have to ask a ton of clarifying questions because they miss out critical information and most of the AESE group are not very helpful. Exam isn't too difficult though. - Jamal 19/20

## [CO339 Performance Engineering](http://www.imperial.ac.uk/computing/current-students/courses/339)

Holger’s part was really interesting, pretty well taught imo. Recommend taking ACA if you’re thinking of taking this since he talks about architecture quite a bit. Giuliano’s half was a lot more dull, but the slides and tutorials are good prep for the exam. Holger’s half is harder than Giuliano’s imo. - JZ, EIE, 2020/21

Holger’s part is similarly delivered to advanced databases, however it’s made slightly more practical with labs where you profile things hands-on and the coursework is cool. No one ever seems to have a clue what on earth goes on for Giuliano’s coursework but he was super nice marking it and when it comes to the exam, his questions require you to follow some simple logical steps to get an answer. Tiger (2019/2020)

Another very fun and enjoyable course by Holger, with good, interesting courseworks and a relatively challenging but doable exam. If you’ve taken Holger’s courses before, there shouldn’t be any surprises - Matthew (2019/2020)

Very useful course. Holger’s side is quite accessible even if you don’t have much experience with C++, which he uses frequently, and the lectures are pretty interesting (his side overlaps a lot with Advanced Computer Architecture as both cover branch prediction, speculative execution, vectorized instructions, and a few other topics). Giuliano’s side is fairly easy and exam questions are similar to tutorial sheet questions in terms of structure. - Fawaz (2019/20)

This was an awesome course, super underrated. Holger massively upped his game in this course and provided plenty of resources this time (we had to complain for a while but he got round to it), he’s a really enthusiastic lecturer and there was a sensible amount of content too, giving you enough time to learn all of it thoroughly. I expect this to be one of my best exams. - Rohan Pritchard (2018/19)

Really useful course. Definitely still maturing but was by far the most relevant course you can take in third year if you want to go into software engineering imo - Isaac Hutt (2018/19)

Again, Holger made this course a lot of fun. Don’t go if you want an organised lecturer, plenty of problem sheets and like “learning by the book”, as there’s none of that here. However you will learn many useful techniques. - Tom (2018/19)

Hype – Akshat 2021

Where to begin... I found the course content pretty interesting, but overall it was a complete shambles. Holger’s content was well organised but his coursework was extremely long and ambiguous, and he only loosely clarified what he wanted 3 days before the original deadline (it got extended). I easily spent ~20 hours on this coursework (for only 10% :’) ) and it left me behind in all my other modules. Lluis just simply didn’t seem to care about the course. His content overlapped way too much with Concurrency and he provided no representative practice exam materials of any form, despite several requests from students. I’d recommend it for the content but be prepared that it’s a time sink and there’s little practice materials to go by. - Alex, 2021/22

* Holger is slightly above average – not the best I’ve seen – but shows raw passion and knows his material. The material in general is very interesting and I would agree with others in that there’s a lot of useful material to learn, even for industry. His half consists of watching an online ~2 hours video, with a discussion section of equal length (Lluis’ section on the other hand was purely hybrid).
* His (Holger’s) coursework was interesting, but very ambiguous (and personally found it relatively lengthy) and this is something which I struggled a lot, and resulted in my getting the lowest coursework mark I’ve ever had at Imperial (despite opportunities for getting bonus marks). Please note that it looks like a “just me” case.
* Now, for Lluis. His material is similarly interesting even though disjoint from Holger’s (and has almost nothing in common with what Giuliano would teach in this class in the past), and more hardware focused. There’s some overlap with concurrency.
* For his (Lluis) half, we were supposed to have a standard 2-week coursework, but for some reason he cancelled a week’s classes (travelled somewhere for some undefined reason) which I think should have been handled better, which threatened the coursework to be too close to the exams. It was replaced with a smaller one-week coursework which was easy (though occasionally ambiguous) and could be done in about a day or so.
* The exam for both parts were very lengthy and this is the exam I most struggled to complete on time. Notice that unlike some other modules, Holger does not recycle questions (but preparing from past papers still help as some themes do recur), and always gives a good rider.
* TLDR; not an easy one, but interesting and useful (both for industry and otherwise), and I would recommend taking this module. One issue in general is that there’s little material to practice for both parts; you need to “get” the material on your own.

(Now called System Performance Engineering) This was probably the best module I have taken at Imperial. Holger is an amazing lecturer, truly helpful and invested in your understanding. Lluis is a good lecturer too, but not particularly outstanding like Holger. Holger’s part was quite self contained and very useful for understanding fundamentals of optimisations – Lucas 22/23

-- Leaderboard, 2021/22

## [CO347 Distributed Algorithms](http://www.imperial.ac.uk/computing/current-students/courses/347)

Quite an interesting module. Learning how to program in Elixir teaches you a new way of thinking which is cool although I would warn you of the learning curve. The module doesn’t really have that much content, introducing mostly consensus algorithms, but it’s quite conceptually difficult. The CW is quite chunky and difficult but gives really good insight into the topics of the module. I found the teaching pretty underwhelming but there’s enough stuff online. - Zaki (22/23)

They switched the CW back to multi paxos this year and it was very challenging. Yet the course is fine overall. (2020/21)

As the comment above mentioned, they switched back to multi-paxos for the cw, which was slightly challenging but definitely quite exciting when you finally understood it and how it all worked. Debugging your own code was definitely the hardest part, not really knowing whether the results you got were realistic or not. There was not a massive amount of actual lecture content other than the algorithms and concepts that were introduced, so fairly straightforward. Elixir was also fun to mess around with, but some of the language features did sometimes interfere with what would otherwise be intuitive. Would overall recommend, fairly useful and engaging content – Prab (2020/21)

The multi-paxos cw shouldn’t be done by yourself, find a partner. They reference a book a lot and realistically, for the first half of the course, you could just read the sections of the book they reference and not watch the lectures and be fine. - Ben (2020/21)

CW is fine this year since they changed it to RAFT. Many group seems to be getting 20 out of 20. Normally only 2 hours of lecture per week, so it doesn’t really have too much materials. Content is interesting, but teaching can still be improved. - Leonard Ge(2019/20)

CW was actually fine. RAFT is fairly straightforward and easier to understand than PAXOS (as intended lol). Elixir is still a bitch but you will survive. High key the exam requires a little no a lot more extra reading and intuition because ya boy got destroyed - Chandler Low (2019/20)

Stopped turning up. CW is pure cancer, Naranker is super boring, elixir is not taught and most course was simply too much for me with all the other subjects. Theory is useful, new language is always good: but neither of the two was taught well, therefore I dropped it. Did not even bother going to lectures like in pervasive, because they simply put me to sleep. Overall it is a no from me. - Dima (2018/19)

“Your girlfriend will leave you, your life will leave you” - Oscar Zhang (2018/19)

This course is not an easy one, the coursework does take a long time too (although I think they’re looking to reduce the scope of the CW next year). Nevertheless I don’t regret taking it, it’s an interesting course and I think that it’s really relevant, will certainly be useful for the CV. - Rohan Pritchard (2018/19)

I’m using a lot of this stuff in my placement - it’s useful and very interesting. Fair warning however, you will have to debug your own multi-Paxos implementation. - Tom (2018/19)

Really interesting course. The coursework is hard but you could get good marks even with a solution that was kind of broken (lol). Slides are long and you need to do some outside reading to really understand what’s going on, but overall I enjoyed it. - Alex, 2021/22

Hype – Akshat 2021

## [CO395 Introduction to Machine Learning](http://www.imperial.ac.uk/computing/current-students/courses/395)

A great overview of machine learning concepts which every Computing student ought to know before graduating. Not particularly difficult (there’s no hard maths) and the course doesn’t move very quickly. Covers more than just standard neural networks so the variety is nice. The courseworks are manageable and provide good insight into the concepts you learn in class. I would recommend everyone take this for the knowledge it provides. - Zaki (22/23)

Very beginner friendly & practical course with weekly labs. Not very maths heavy. I recommend taking this and if you’re interested in more theoretical fundamentals in ML, self-study Stanford’s CS229  
(<https://www.youtube.com/watch?v=jGwO_UgTS7I>) - Freddy 22/23

Was compulsory for me but wouldve taken it anyway. Was in first term for us. Good lectures, interesting content. Not too much to say. Very good as an intro course, it gets you in slowly and difficulty increases very smoothly so you don’t find yourself completely lost at any point. Hardest part is the derivative stuff with back propegation (which pretty much is always on the exam) so spend some time to get used to it.

Courseworks are both fun and not too hard, but can be a bit time consuming

Would recommend – Alfie (22/23)

Pretty well taught, good introduction. Get a good group for the coursework. If you’re EIE: do the DoC ML module if you want more practical stuff. Haven’t heard any good things about EEE ML course, which is also a lot more theoretical. - JZ, EIE, 2019/20

It’s a good and broad introduction to ML, I’d definitely recommend it to anyone interested in this topic. The module made me realise there was more to machine learning than just neural networks. For example, I enjoyed learning stuff -=-=about evolutionary algorithms. - Maks (2020/21)

Honestly a good course well worth taking if you’re fresh to this area, it gives you a rounded-base of competency in most of the useful and applied ML techniques and I hear has a lot of overlap with Spring Term’s Computer Vision. The Coursework took a little bit of time to do but ultimately was not that hard and with a good group you can bang out 90+% on both of them, as it’s just a matter of passing LabTS tests and having a legible report. Make sure to do that so you have a good cushion for the exam. Exam itself was not that bad this year, there are tiny curveballs here and there to compensate for the online setting and a reduced amount of recall questions to boot, and you’ll be asked predominantly questions based on the techniques you learnt which rely on quite simple applications of maths to do. If you want a decent grade while also learning stuff that’s more relevant than most, take it.

(As an EIE student, be aware that by taking this module over the EE ML module – which seemed to just be entirely theoretical, you do forfeit taking the EE Deep Learning module in the Spring Term, which has a lot more application, as EE ML is the pre-req, don’t deep it though because DoC has a DL module in the 4th year so you’re calm) - Aryan (2021/22).

Really useful course for teaching you the basics of ML and also how a lot of well known ML algorithms and models are implemented, since the labs make you do this in groups yourself (For both Decision Trees and Neural Nets) and write a short report on it. The lecturers try really hard to be interesting but couldn’t get people to participate or calm down that easily, although I’ve found the concepts they taught to be really useful in my placement. The exam isn’t too bad either - Just learn standard stuff that comes up in past papers. Tiger (2019/2020)

Hype – Akshat 2021

Recommend it for the courseworks, which will take most of the time you spend on this course but it’s a good practice. The lectures themselves are a bit useless and shallow, especially if you take Computer Vision which explains the same concepts more in depth. Exam is easy and asks you definitions and to execute common algorithms by hand.- Theo (2019/2020)

Definitely an introductory course but I didn’t have much ML experience so was useful. Covers a reasonable amount of ground but nothing in depth as mentioned above; I think this is the best approach at this stage (third year) as gives you nice foundation to build off for future ML work (or gives you just enough info if you don’t plan to do more ML). Not a lot of content and most of it is reasonably easy. Would definitely recommend. - Avish (2019/20)

Very much ML 101, covering the basics and a few applications of ML. The two CWs are worth 30% of the module with the 2nd one being weighted more than the 1st. Both CWs take a lot of time, so be sure to make time with your group to really crack on. Nice overlap with Computer Vision, recommend taking on both. - Anindita (2019/20)

The course itself it pretty interesting (if you pay attention and read the notes lol) and the CWs are actually really cool. It felt like one of the most useful and practical applications of computer science I’ve had and really gave you a good and deep understanding of the module. Would highly recommend you do the CW together as a group as otherwise it really won’t mean much. The exam is also pretty standard and straight forward. - Sam Trew (19/20)

Fairly easy as a module, very tedious CW with a lot of report writing. Content is very manageable, fairly interesting as an intro to ML, but lectures are not worthwhile. CW can easily push your grade up, as it is 33% rather than 15%. Overall would recommend: not too difficult, enjoyable. - Dima (2018/19)

Would also recommend, the coursework is good (and in groups which makes it more fun), exam this year was probably easier than they’ll make it next year but I think it’ll still be approachable, and I’m using a lot of this stuff in my placement. It’s definitely an intro course… They touch on a lot of subjects but don’t go very deep into anything, which has its pros and cons. - Rohan Pritchard (2018/19)

Fairly bland course but taught well. Would revere recommend. Isaac Hutt (2018/19)

This course is fine, even for someone like me with little experience of Python. Well taught (good use of Quizizz as a revision activity which I want other modules to do), and all of the three lecturers are (while not outstanding) perfectly fine. The labs are optional but do help with understanding the course material, and so are the tutorials (though it would help if explanations would’ve been provided). For the coursework, it does help to get a good team as doing well can easily get you a near 100% (and 30% of the module), though I would have liked to see an individual component. The exam is OK but very lengthy and Antoine’s section had a tricky sting-in-the-tail. There’s some overlap with Maths for ML (which has a more mathematical treatment).

Overall, this is a good “filler” course (and presumably important for those more ML-focused than I am) and taught well. Just that compared to the other modules I took, this was slightly boring.

**Update (February 2022):** I didn’t do as well as I thought. Was this exam scaled down? – Leaderboard, 21/22

# **Mathematics**

# Autumn

## M2PM2 Algebra II

Great lecturer, super engaging and fun. The course was fun but damn it’s hard. You don’t really need any prior knowledge since Martin redefines everything from scratch anyway. Ties in with first-year group theory and linear algebra, and will be useful for future algebra modules.

Topics: Groups, rings, vector spaces.

- Omar Tahir (19/20)

## M3PA23 Dynamical Systems

Just an awesome, fun course. Best part: NO PREREQUISITES! Don’t need topology, analysis or differential equations at all. All you need to know is what metric spaces and open sets are (5 mins on Google). Really interesting, proves lots of cool stuff to do with chaos. Basically, if you’re sad that the chaos & fractals course is gone then take this to make you feel better. Jeroen is an awesome lecturer, I missed a class test and he sent me a personal email asking if everything was alright, what a lad. Class tests are fair and if people struggle he bumps everyone up. Although topology and measure theory are useful, he redefines everything from scratch just so you don’t have to do any prerequisites!

Topics: Chaos, circle maps, topological conjugacy, statistical properties of dynamical systems.

- Omar Tahir (19/20)

## M3S8 Time Series

Best module of the year. Don’t know if I’ll ever use it but it is interesting, very well taught by Ed and above all coherent. There is a lot of content but everything makes sense in the end when you learn it. Exam is doable if you do the problem sheets. Courseworks have some practical questions (code) but you have to type the working out. They are tricky to do in 1st term while doing the 3rd year project. - Theo (2019/2020)

Very interesting module, especially if you’re into finance or economic predictions. Courseworks don’t require too much time, although the last one is set during DoC revision season so bear that in mind. You should be fairly comfortable with M2S2 level stats (e.g. covariances) and Fourier transforms before doing M3S8.

BONUS: overlaps with Performance Engineering and Computer Vision a tiny bit 😉

## M3A49 Mathematical Biology

Not too difficult conceptually and a very interesting course. Had Phillip Thomas this year who imo is a bad lecturer. Not the best explainer and constantly makes little mistakes (which can be an issue if you’re like me and a bit slower with following lectures) but the content is engaging and his Latex’d notes are lovely so overall it’s fine. There is one CW set in early Dec due early Jan - very nice for JMCs who have lots of other work during term. Will update this after exams in May/June. - Avish (2019/20)

## M3S4 Applied Probability

Low-medium difficulty. Once you understand Markov Chains in general, the final two chapters shouldn’t be too hard. I had A. Veraart who was one of the best maths lecturers I’ve had - very enthusiastic about this module. Lots of problem sheets (including two which you can get marked but they won’t count for your grade) and two assessed tests spread out nicely for JMCs. Will update this after exams in May/June. - Avish (2019/20)

The course is fine but the exam this year is hard.

## M3P5 Geometry of Curves and Surfaces (aka differential geometry)

Relatively easy module. Lots of computations in the proofs but none are particularly hard. Interesting results at the end that connect local stuff (geometry) and global stuff (topology), which may include the proof that any surface can be triangulated and given a fixed invariant (the Euler characteristic). Courseworks and examination are easier than other pure modules. Lecturer was Tom Coates last year (different this year), who was very clear but the overall pace was kinda slow so some topics were not covered. No prerequisites in particular, but a good understanding of first year analysis/calculus would be good. No topology required, probably. This module would be essential for later modules like M4P52 Manifolds and M4P51 Riemannian Geometry, and useful but not necessary for other geometry modules. Handwritten notes are available. - David Ang

## M3P20 Geometry I: Algebraic Curves (aka algebraic geometry)

Relatively easy for the first half of the module (very explicit equations for curves) but gets weird/tough near the end (Riemann surfaces, Riemann-Hurwitz formula). Personally felt that the content is a bit too much, and some of the proofs are very long and detailed (Bezout’s theorem) but not all are examinable and it’s not that easy to determine which are. Courseworks are doable and examination was fortunately quite easy in comparison to the length of the content. Lecturer last year (different this year) was very fast to cover the sheer amount of material but explanations were clear. Prerequisites include complex analysis (mainly holomorphicity) and topology (Hausdorffness, compactness, facts on homeomorphisms). This module could be useful for M4P33 Algebraic Geometry but takes on the complex analytic perspective. - David Ang

## M3P8 Algebra III (aka ring theory)

Easy module especially if you have played with rings before, and the first half is a repeat of stuff from first/second year algebra courses. Nothing particularly interesting in terms of big theorems (maybe the Hilbert basis theorem?) but covers a LOT of ground (even algebraic geometry) to tackle later algebra courses. Lecturer was David Helm (same this year) who could be quite fast at times and offers lots of examples from number theory in his explanations. Courseworks and examination were tough (as for any algebra module, honestly, but last year was quite a huge leap in difficulty). No prerequisites in particular, but the first half will be a breeze with some knowledge of ring theory. This module could be useful for M3P15 Algebraic Number Theory (but the required content will be restated here), M3P11 Galois Theory (just a tiny bit), and M4P55 Commutative Algebra (50% content overlap), as well as basic cryptography. Fully typed notes are given in website. - David Ang

## M3P10 Group Theory

(Did not really attend the classes here, but went through most of the content with people who did and took a good look at the examination paper) A step up from first year group theory without enforcing a prerequisite of second year group theory since everything is restated from scratch. Not a fan of the lecturer’s style of teaching (typed notes on the visualiser with his own explanations, will go along the same lines this year) but is a unique experience if you wish. May be useful for later algebra courses but only indirectly (not necessary for M4P61 Infinite Groups). Fully typed notes are given at the beginning. - David Ang

## M3P17 Algebraic Combinatorics

(May not run this year, but given that M3P80 Graph Theory was cancelled, it might be readded) Relatively easy module in terms of content that is split into concrete three parts (error-correcting codes, strongly regular graphs, and t-designs) but are unified at the end. Personally found it very interesting due to the many unexpectedly open questions (Moore graph with d=5 and k=57) and highly-structured proofs yet seemingly come out of thin air. Lecturer was quite engaging (different this year and still TBC), with doable courseworks but not 100% sure about the exam. No prerequisites (maybe the definition of a ring?) and may not be useful for future modules(possibly M4P61 Infinite Groups?) but certainly gives a mathematical understanding of error-correcting codes (seeing that Information and Coding Theory is not available this year). - David Ang

## M3P65 Mathematical Logic

Relatively easy module at the beginning due to the significant overlap with first year propositional and first-order logic, but gets weirder/tougher (personal opinion) nearer to the model theory and set theory bits. Many of the definitions (syntax/semantics and soundness/completeness) are tackled from a mathematical perspective that is much more rigorous than first year logic. Lecturer was David Evans (same this year, with someone else) who was very fun but it might get dry with his style of teaching. Courseworks are relatively doable but the examination was unexpectedly very long (although not too hard). No prerequisites and has no continuation modules (maybe M4P49 Model Theory if it exists?) but is itself enlightening in content. Handwritten notes will be produced every lecture. - David Ang

Just don’t. Ghadernezhad is a really bad lecturer and it’s 90% definitions with no real explanation as to why. Propositional is easy, first-order with model theory becomes a nightmare. Would’ve been interesting if he’d explained why we were doing any of this. Haven’t done Evans’ part yet (whoops) will update after exam. - Omar Tahir (19/20)

## M3P14 Number Theory

Relatively easy module in terms of content except the fourteenth/last chapter that has a huge proof (which may not be examinable this year, certainly wasn’t last year), but most of the other proofs are examinable. Nothing particularly interesting in terms of big theorems and covers the same stuff every year but examinations are rather unpredictable excluding the obvious questions (congruences, sums of squares, continued fractions, prime numbers). Could get difficult given the sheer number of hand computations (adding many four-digit numbers) under exam conditions. Lecturer was Toby Gee (same this year) who was very clear with huge handwriting. No prerequisites (maybe the definition of a ring?) and most certainly won’t be useful for future number theory modules in terms of content, but will serve as motivation and history as well as applications to conventional cryptography. Fully typed notes are given in David Helm’s website, but typed notes are produced chapter-by-chapter as the lectures go on. - David Ang

## MATH60015 Quantum Mechanics 1

Content not too difficult, Eva-Maria is a great lecturer and very engaging – Andrew (JMC 21/22)

## [MATH60024 - Computational Linear Algebra](https://www.imperial.ac.uk/computing/current-students/courses/math60024/)

My first choice out of all coursework-only Maths modules I have taken (I have taken 4 in total and this one was the best out of them). If you like Computational algorithms, and just coming up with code to speed up Maths calculations, definitely take the module. As you would expect for a coursework-only module, it is quite time-consuming, the weekly exercises are marked and there are also 3 or 4 courseworks (3 for Year 3 students and 4 for Year 4 students). The only bad thing is that on purpose we are given about 3 days to complete each coursework, so I don't know if anyone slept during these three days, because there is not enough time to finish. However, I appreciate that you are done in 3 days for each coursework and then you are somewhat free. You also have the Christmas break to finish up with the exercises and the last coursework which gives you some more time. The lectures are well-structured, there are notes in the form of a website, feedback is given on time and the module team is very responsive. - Zlatina, JMC 21/22

## [MATH60023 - Numerical Solution of Ordinary Differential Equations](https://www.imperial.ac.uk/computing/current-students/courses/math60023/)

A tempting choice for people who want to do coursework-only Maths modules. Definitely better than Computational Partial Differential Equations, but I would prefer Computational Linear Algebra or Methods for Data Science if I had not already taken those, just because in those 2 modules the courseworks are less difficult (not easy, but less difficult). Overall, this was a very interesting module if you enjoy computational mathematics and want to learn about how to algorithmically solve differential equations – this is quite useful and I learned quite a lot of techniques. As a warning, this module is very time consuming – each week there is an assessed tutorial, which counts for about 2-3% of your final mark, and you submit online and are given around 3 hours to complete – but these 3 hours are fixed so you basically have to free your afternoon let's say each Friday (depends on when the tutorial slot is) to solve and submit this tutorial. On top of that there are 4 courseworks (I did it in year 4 but for Year 3 students there are 3 courseworks) and the third and the fourth courseworks are very long and quite difficult. There were many people who just gave up on solving the fourth coursework because it required a massive amount of effort. With that said, it is not impossible to do – I had maximum marks for the third and fourth coursework, but I ended up having to not sleep much and invest many full days in these courseworks. I appreciate that this saved me from an exam, though, so there is a trade-off – you decide. Last thing I wanted to say is that there are nicely written notes with examples and explanations, but that does not mean that you will not need to write your algorithms in the courseworks from scratch – they cannot be taken word for word from the notes. - Zlatina, JMC 22/23

# Spring

## M2PM5 Metric Spaces and Topology

Great course. Doesn’t need any analysis, does need some algebra (M1J1/2 is fine I think). Very pure, doesn’t really connect to any previous ideas besides some analysis stuff which will all be redefined anyway. Need it for algebraic topology. Alexei is great, provides good examples when needed and can explain and prove things very clearly. No notes although he says he follows his chosen book pretty much exactly.

Topics: Metric spaces, topological spaces, homotopy and fundamental groups.

- Omar Tahir (19/20)

## M3P21 Geometry II: Algebraic Topology

Harder module in terms of content, especially if unfamiliar with topological notions. Covers roughly the first two chapters of Allen Hatcher’s book (the fundamental group and homology), but was shortened a bit due to lack of time last year. Lots of rather involved proofs of big theorems (Seifer-van Kampen, Mayer-Vietoris sequences, Excision), all of which may look intimidating at the beginning (still is for me) but is extremely rewarding. The lecturer last year (different this year) was very clear, with kind courseworks and a very kind examination (mostly computations and even a true/false question) despite the content. The main prerequisite is a thorough understanding of topology (connectedness and path connectedness) and basic homotopy theory (fundamental group), as well as group theory (free groups, free products). This module would be essential for any topological modules such as M4P54 Differential Topology and possibly M4P57 Complex Manifolds, and useful for motivating homological algebra in M4P63 Algebra IV. - David Ang

## M3P11 Galois Theory (aka field theory)

Easy module in terms of content (basically only the fundamental theorem of Galois theory) but not so much in terms of abstraction (normality and separability conditions). The style of lecturing changes every year, so it could start off directly with the main theorem and its proof, but goes into very minute details later. Once the main theorem is fully proven, there will be lots of examples (the last ten lectures of last year were just examples) and computations/applications. The lecturer was Alessio Corti (same this year) who was certainly very interesting and will provide lots of interesting perspectives to take. Courseworks are easy but are in the form of tests, and the exam had a large computation (fewer proofs and a true/false question) component. Prerequisites could be first and second year algebra modules, or better M3P8 Algebra III. This module would be useful for any number theoretic future work, as well as being helpful to stuff like M4P33 Algebraic Geometry. No notes will be given at all, but a short summary is available on his website. - David Ang

Notes were given this year but they weren’t that good (didn’t follow lecture order and are missing proofs and some lemmas near the start). Corti taught it. Struggled with the weekly tests and found the proofs in the exam hard – Ben (2020/21)

## M3P12 Group Representation Theory

(Attended only the first few lectures and asked several people who did this module) Definitely a harder module in terms of content (there is a tonne of it, split into many chapters) as well as the courseworks and the examination. The prerequisites are first and second year algebra, and this module will be essential to M4P46 Lie Algebras and M4P72 Modular Representation Theory, as well as any form of algebraic future work. Fully typed notes are given in the beginning. - David Ang

## M3P15 Algebraic Number Theory

Easy module in terms of abstraction (computations here are very concrete and number theoretic with some tools from algebra) but could be quite long content-wise. Fortunately, M3P8 Algebra III (and to a much smaller extent M3P14 Number Theory) covered literally 75% of the content last year, leaving a lot of room to revisit the proofs and focus on the computation-heavy last bit (on ideal class groups and unit theorems). The lecturer (different this year) was aware of this and went slightly faster but still gave easy courseworks and an easy examination. The prerequisites are first and second year algebra (or just M3P8 Algebra III), and is definitely useful for any number theoretic future work. Typed notes are given over the lectures. - David Ang

Well-taught course with good notes. The exam was heavy computation based which meant that it got scaled down (needed 74% for a first). It’s a nice number theory progression from Algebra 3 but Alg 3 isn’t required, just useful. - Ben (2020/21)

## MATH97069 Dynamics of Learning and Iterated Games

All cw based, the oral part is just a presentation about your coursework where you get to “highlight” your strongest parts. You can ignore the parts of the course that don’t relate to your project. Alongside your project, you need to submit some exercises but they’re relatively easy. For the project, it takes about 2 weeks to do but it’s due after Christmas so just say goodbye to your break and you’re fine. The notes are good and the lecturer explains them in lecturers pretty well. - Ben (2020/21)

## Introduction to Statistical Learning

The course introduces some many ML methods that are interesting -and not too well known. However, some concepts are abstract and hard to grasp. The problem sheets don't help much in the understanding of the content. There are a lot of R code samples included which is great but don’t really use it. Coursework is dead easy (this year was: find an interesting dataset and apply scaling methods on it) and the exam questions were almost only about proofs from the notes (would have been hard to learn everything without open book) similar to M2AA3 style.

(Content: Ridge, Lasso, Principal Component Analysis, Classical and non-metric Scaling, Kernel Smoothing, Wavelets, Smoothing splines, Local Polynomial, …)

Theo (2019/2020)

Statistical Modelling II

## MATH60017 Tensor Calculus and General Relativity

Honestly a very disappointing module. Very watered-down version of GR that included bare hand waving which made everything so much more difficult to understand because things weren’t adding up. Easy exam – Andrew (JMC 21/22)

## MATH60018 Quantum Mechanics 2

Probably one of the best modules I’ve taken at Imperial. Ryan Barnett is a great lecturer, always available for questions and did a good job of introducing complex material– Andrew (JMC 21/22)

## [MATH60025 – Computational Partial Differential Equations](https://www.imperial.ac.uk/computing/current-students/courses/math60025/)

I would avoid this module if I could – if you are looking for a coursework-only module, you can take all other coursework-only modules before you resort to this one. First, I could actually understand most of the things in this module, because I took Numerical Solution of ODEs before that and there was overlapping content at the beginning, otherwise I don't think I would have been able to. If you are not very familiar with PDEs DO NOT TAKE IT, since there is not introduction content – the lectures jump straight to new content. The module doesn't have well-written notes, they are very short and quite confusing at times – the topics don't flow nicely from the previous to the next and while doing the courseworks you might end up wondering where the content that this coursework is actually testing is. The lecturer is not very responsive on Ed and this was frustrating. Another frustrating thing is the courseworks were not released on time, some of them had a delay of more than a week, so if you have made your schedule for studying with the other modules, you will need to revise it over and over again. And because of the big delays, although extensions were given, they ended up going into the April break, after overlapping with the Computing exams, so the overall period for completion was shortened. There were 4 courseworks (For Year 4 students and 3 for Year 3) that were quite ambiguous and for me they were very difficult to do – yet another module which required me to give up on a lot of my sleep to be able to finish it. Feedback is not given on time, in fact term 3 started and we still don't have the feedback for our second, third and fourth coursework. - Zlatina, JMC 22/23

# External

## Embedded Systems

This module is 100% CW. This will eat into your free time I can guarantee it. However the first half is building an IOT device and it’s pretty cool and normally is more programming than hardware. The second is programming a motor to do various things (min bitcoin, play music etc.). This is pretty low level programming and is much more complex. You will need someone in your group who does EEE/EIE or at least has lots of hardware experience as otherwise it will be close to impossible. A decent and interesting course but be weary. - Sam Trew (19/20)

# Horizons

## Korean Level 3

Wonderful course, absolute banter with the teacher. Lessons were a lot more focused on the discussions of Korean culture and history than learning new vocab or grammar. Quizlet link [here](https://quizlet.com/class/12569714/) for all the new vocab and grammar covered in Level 3 (you're expected to know Level 1 and Level 2 stuff). 2 hrs of classes a week but each coursework takes like 6 hours to complete (5 courseworks in total). You prepare for speaking in the last few weeks of class and the written exam is similar to stuff viewed in class. Definitely take it if you've done Level 2 or if you already know conversational Korean. - Andy (19/20)