## Potentially useful links

* <https://twitter.com/theotheredmund/status/1349453230762196992>
* <https://exams.doc.ic.ac.uk/>
* SOLE: [UG SOLE results | Imperial students](http://www.imperial.ac.uk/students/academic-support/student-surveys/ug-student-surveys/ug-sole/ug-sole-results/). Useful for gauging the quality of courses.

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. Cry. p

# Contents

[Potentially useful links 1](#_Toc2027477818)

[Contents 1](#_Toc1597311639)

[Autumn 1](#_Toc227488074)

[CO404H Separation Logic: Local Reasoning about Programs 1](#_Toc304693899)

[CO408 Privacy Engineering 1](#_Toc1666026732)

[70023 Scalable Software Verification 2](#_Toc757368323)

[70008 Concurrent Processes 3](#_Toc1408078872)

[CO410 Scalable Systems for theDepp Cloud 3](#_Toc397497613)

[CO424H Reinforcement Learning 4](#_Toc646772573)

[CO438 Complexity 5](#_Toc141125874)

[CO445H Advanced Security 6](#_Toc172531016)

[CO471 Advanced Issues in Object Oriented Programming 6](#_Toc817102347)

[CO496 Mathematics for Machine Learning 7](#_Toc1446674183)

[CO499H Modal Logic 8](#_Toc901138518)

[CO422 Computational Finance 9](#_Toc1158383314)

[70068 Scheduling and Resource Allocation 10](#_Toc2044620969)

[Spring 11](#_Toc2046818476)

[CO409 Cryptography Engineering 11](#_Toc272166518)

[CO416 Machine Learnin for Imaging 11](#_Toc210606927)

[CO417 Advanced Computer Graphics 12](#_Toc1305705835)

[CO433 Advanced Robotics 12](#_Toc2084137243)

[CO440H Software Reliability 12](#_Toc467214527)

[CO446H Applied Network Security 13](#_Toc50364733)

[CO447H Advanced Security in Smartphone and IoT Systems 13](#_Toc229685653)

[CO460 Deep Learning 13](#_Toc613157225)

[CO466H Inductive Programming 14](#_Toc1742181779)

[CO467 Principles of Distributed Ledgers 14](#_Toc1971916827)

[CO468H Probabilistic Programming 15](#_Toc808636251)

[CO449 Probabilistic Model Checking and Analysis 15](#_Toc180417174)

[CO475 Software Engineering for Industry 15](#_Toc495759626)

[CO477 Computational Optimisation 16](#_Toc782387849)

[CO490H Natural Language Processing 17](#_Toc1362076986)

[CO493 Probabilistic Inference 17](#_Toc2026719115)

[CO498H Logics for Strategic Reasoning in AI 18](#_Toc475923106)

[70066 Decentralised Finance 18](#_Toc104808644)

[70030 Knowledge Representation 19](#_Toc1437263556)

[Mathematics 19](#_Toc397548006)

[Autumn 20](#_Toc207638725)

[MATH70005 Optimisation 20](#_Toc1259745262)

[PHYS70009 Quantum Information 20](#_Toc1012104383)

[M4P32 Number Theory: Elliptic Curves 20](#_Toc945239645)

[M4P52 Manifolds 20](#_Toc1053584952)

[M4P55 Commutative Algebra 20](#_Toc1380442172)

[M4P58 Modular Forms 21](#_Toc1289553495)

[M4P61 Infinite Groups 21](#_Toc1015915182)

[M4P72 Modular Representation Theory 21](#_Toc165986628)

[MATH70024 - Computational Linear Algebra 21](#_Toc186330144)

[MATH70023 - Numerical Solution of Ordinary Differential Equations 22](#_Toc183257881)

[Spring 22](#_Toc2064152384)

[M4P33 Algebraic Geometry (variety theory) 22](#_Toc457061831)

[M4P46 Lie Algebras 23](#_Toc1475487169)

[M4P51 Riemannian Geometry 23](#_Toc165566831)

[M4P54 Differential Topology 23](#_Toc591295699)

[M4P57 Complex Manifolds 23](#_Toc1362502131)

[M4P63 Algebra IV (homological algebra) 24](#_Toc617641672)

[MATH70025 – Computational Partial Differential Equations 24](#_Toc1133191491)

# Autumn

## [CO404H Separation Logic: Local Reasoning about Programs](http://www.imperial.ac.uk/computing/current-students/courses/404H)

Nice course, except large coursework and a long exam. This is a programming language course, so it is expected that you learn even more new symbols (yay). You’re expected to do quite a lot of mechanical and boring stuff, e.g. drawing derivation trees, long proofs. Philippa said this course is going to grow into a full course, so there are more on semantics and soundness, which makes this course more interesting. - Fangyi

## [CO408 Privacy Engineering](http://www.imperial.ac.uk/computing/current-students/courses/408)

Very interesting module, although beware of what you’re signing up for and don’t get catfished by the first two lectures. The module is split into two halves, one about how to break privacy in data (and mitigations to such attacks), and the other with more of a mathematical / theoretical approach towards privacy preserving mechanisms.

Pretty good course when I did it 2 years ago, the lecturer (if it isn’t Naranker) is really into it and there is some super interesting stuff you can learn. Also the coursework isn't that hard.

It’s very interesting. Naranker is still doing half of it - confirmed not too helpful. Other guy is very good.

First half is very concept based but also based around real world applications, useful tutorials to help drive home the things you learned, also Yves is a very engaging lecturer, and his flipped classroom style definitely worked well for remote learning. On Naranker’s side, more concept based than real world based; slides are also very chunky and kinda hard to understand. Coursework was based on Naranker’s half and was doable in a few days. Overall, a decent course though you do have to put in a couple of hours to really wrap your head around the stuff covered in the 2nd half. – Andy G (online learning)

My opinion on the content is probably biased as I figured out too late I just wasn’t interested. However, the course is very segmented with lectures not really tying together at all. Everything remains very separate and taught by 3 different lecturers. Also, in our year, half of the entire module ended up relying on about 1/10th of the content which was pretty shocking if you weren’t as comfortable with that specific bit. - Sam Trew (online)

I found this module quite boring in all honesty. The coursework was kinda stupid as well. A bit meh all round I think – wasn't that interesting but I’ve taken worse modules before. JZ, EIE, 21/22)

This module was incredibly boring, one of the worst prepared/delivered in my time at Imperial. Although the content was interesting, Yves put absolutely no effort into delivering it in an engaging way (just handing us lecture recordings from 2019) and about 90% of the lectures (if you went) were spent listening to Yves monologue or watching students deliver their coursework presentations on papers. The coursework setup was ok but marked harshly and kind of pointless as it was way more useful to read the papers anyway. Have taken worse, but would have been much more engaging with a set up like Scalable Systems & Data – Alex, 22/23

## [70023 Scalable Software](https://www.imperial.ac.uk/computing/current-students/courses/70023/) Verification

If you liked the first half Models of Computation (with Philippa Gardner and Petar Maksimovic) i.e. operational semantics, Hoare logic, then you’ll really get along with this course. Philippa puts a large focus on intuitive understanding of the content, which fits together very nicely with the more formal parts of it. A lot of time was spent in open conversation and going through examples, rather than just dryly blasting through content – without rushing anything, too. The 2 courseworks were a little weighty, but not unreasonably so. I personally recommend it. - Nat K (21/22)

Really awesome course if you’re into operational semantics & Hoare logic. Phillippa made a lot of effort to ensure the content was understood (although sometimes she actually made it more confusing lol), as well as linking the content to wider industry with talks from Infer@Facebook and Peter O’Hearn. The courseworks and exam are a lot of writing, but it’s not too bad. - Alex, 22/23

Very well prepared course, plenty of helpful tutorial to go through and predictable exams. Lecturer is really nice and dedicated and the course material isn’t hard. If you are into formal logic and enjoyed Reasoning or Logic in Y1 or MOC in Y2, this course is it. Courseworks can be slightly lengthy but not hard – Mike 22/23

Great course! It had very well-written notes and tutorials with similar problems to what you will get on the exams. The courseworks are quite long, but they are worth the time invested, because they prepare you very well for the exam and help you understand some core techniques. I would say the 22/23 exam was twice longer than I would have expected, but I think it was experimentally that long and will not be in 23/24. Also, not many people take the module, so I would not count on scaling to save me if I had not prepared well enough, and I had not learned how to approach all the problems in the tutorials. - Zlatina, JMC 22/23

## [70008 Concurrent Processes](https://www.imperial.ac.uk/computing/current-students/courses/70008/)

(when it was a half course with the name CO406H) Theory course. Nobuko said that this will develop into a full course. In its current form, the course is quite straightforward. – Fangyi

Didn’t like this one (could be partially a case of having insufficient background, though Nobuko explicitly noted that knowledge of λ-calculus isn’t needed). The lectures are weird – they give you some examples but no answers, and they didn’t discuss this in the only live tutorial for that week! This made understanding the material very hard, the module very stressful, and I ended up dropping this one. – Leaderboard (21/22)

I really liked how this theory module is delivered. It is clear what was expected to know in the exam with almost all types of questions covered in tutorials (for which answers are given 2 weeks after the tutorial session – hopefully, they release them earlier in the future). In fact, this was the only module I took in the first term that had tutorials.  
However, it might be more challenging for non-DoC students. One task with structural induction does always occur in the exam which is not covered. I do not think there was any other prior knowledge required but I guess also doing Models of Computation and Type Systems does make understanding concepts easier.  
Moreover, the CWs are doable in a reasonable amount of time (less than 20 hours combined) with the first one being released early in the term which is helpful as you do not have anything to do at that time. The first one helps you prepare for the exam while the second one involves writing a bit of Go code.  
Finally, there was no lectures, but we were only given short 5-minute guidance videos released from day 1 and told to read the slides ourselves before each week's tutorial session. I quite liked this approach, but it might be changed in the post-Covid world. - Roko(21/22)

The content was interesting enough – it mostly centres around pi-caulculus, which is somewhat like lambda calculus (albeit easier to reason about, in my opinion). The difficulty came when thinking about edge cases and smaller nuances of working with pi-caulculus. Also, Nobuko (at least on her pre-recorded lectures) is extremely difficult to understand. I largely got by on this course through intuitive understanding. On the bright side, the workload is quite light. I think all this course needs to be much more successful is for the lecturer to actually provide answers to exercise questions; tutorial answers are given on a 2 week delay (??) with insufficient explanations, and the single Q&A each week wasn’t nearly enough to cover all the misunderstandings and gaps in content. - Nat K (21/22)

## [CO410 Scalable Systems for theDepp Cloud](http://www.imperial.ac.uk/computing/current-students/courses/410)

*[when it was called CO412H Large Scale Data Management]*

You won’t learn anything from Thomas Heinis

The above statement is correct.

Most interesting course I did in Computing. If you want to find out what it takes to build systems that power Google-scale operations, then definitely take this.

1st half is more large scale data management, which is interesting + a fair bit of revision on storage management. 2nd half is looking at scalable systems and a few papers on how real companies scale their systems and what technologies they use. Easy to follow for both halves; you have to read 4 papers in the 2nd half but he goes over them in a summary format in the lectures anyways. There’s two courseworks: first one is writing MongoDB and SQL queries on a dataset; second one is answering a few questions related to a paper you’re meant to read (really good prep for how the exam q’s are). Exam q’s haven’t changed a lot over the years so lots of papers to help prepare. Overall a very interesting and engaging course, and definitely one of the few useful modules I’ve taken at Imperial – Andy G (online learning)

It’s now Scalable Systems and Data. First half is given by Thomas Heinis. It’s quite boring and probably nothing useful. The second half is more interesting.

2021 Fall edition: the first part on Data is atrocious, the second part is phenomenal

Somehow a module involving Thomas Heinis isn’t the worst one in Autumn term (looking at you, RL). Really enjoyed this overall, Peter’s half was well set-up and the class discussions were really interesting. Thomas’s half is basically another databases course and it's not terrible. Courseworks weren’t too bad either. - Alex, 22/23

Basically half of course is 5 papers (somewhat interesting), the other half puts you to sleep. Not a hard course, but one of the most uninteresting ones so far. If you have more interesting alternatives, I wouldn’t take it. It’s still good marks if nothing else interests you in term 1. - Mike 22/23

Not a horrible course but definitely very long IMO (4h of lectures per week compared to the 2h most courses have) and if you’re not a big fan of reading and analysing papers then this course isn’t for you. For the 2nd half Thomas Heinis’s voice made me fall asleep. 22/23

## [CO424H Reinforcement Learning](http://www.imperial.ac.uk/computing/current-students/courses/424H)

Quite a mixed bag here. Aldo doesn’t really seem to care too much about the students’ opinions on things and any useful information is scattered across 100s of threads on piazza. I mean that literally, about a week into the first coursework there were nearly 200 independent questions where people were just confused. The actual taught content is fairly interesting though. Bear in mind the first coursework was abhorrently unfair depending on random values you got assigned (although they might be fixing that).  
The 2nd lecturer was great and he had a great split between his content and the coursework.

Multiple choice half in exam is complete BS so bpre careful there (they did change the answers as so many people got it wrong...). - Sam Trew (online year)

Agree with the above comment. The first cw is unfair, the second one is quite fun. Both will take a long time to do well. Don’t expect high marks in the cw. The exam multiple choice questions were rather ambiguous. -Davide (online year)

I didn’t want to take the course at first, but I’m glad I did. Whilst there were some parts of the content that were slightly dull at first it definitely picked up and the end result was very interesting. However, the first coursework took a ridiculously large amount of time (100+ hrs), and put me behind schedule for the rest of the term - this was down to a random number that we were allocated that seemed to determine how difficult the coursework was (they may change this). The second coursework was very interesting, but also took quite some time (to do well). TL;DR: - Giovanni P (20-21)

This module is great for those of you who want to spend 72hrs manually tuning hyperparameters! (Update: 144 hours). Prepare to be stressed. Otherwise, really beneficial.

This course is a cheap copy of David Silvers RL course at - <https://www.davidsilver.uk/teaching/>. This first lecturer (Paul B.) has literally copied the slides. He also reads complete sentences out of the Sutton and Barto RL book and calls it a day. My advice is to follow David’s silvers course (at least for the first half) as this man is just useless. That being said, the courseworks prepare you well for the exam (and are fairly interesting as well) and you do get a good feel for reinforcement learning and there is enough content online to support you learning. - Pranav B.(22/23)

Do yourself a favour by ***not*** taking this module. The teaching and quality of assessment is **astoundingly poor.** (Courseworks were marked with extremely robust mark schemes which the lecturers were desperate to uphold and the exam was so bad they had to put in place a **class mitigation**).

If you’re interested you can learn much more from this playlist ([DeepM](https://www.youtube.com/watch?v=TCCjZe0y4Qc&list=PLqYmG7hTraZDVH599EItlEWsUOsJbAodm) ind x UCL RL Lecture Series - Introduction to Reinforcement Learning [1/13]), and your grades won’t suffer either – Alex, 22/23

Please DO NOT take this module as this will pretty much dismiss ALL your interest towards reinforcement learning or even computer science in general. This year (22-23) there are lots of people outside DoC taking this course and totally baffled and confused, and eventually left the comment that “they were totally discouraged against RL and CS as a whole”. The teaching quality is so bad that students from 3 departments (including DoC) have to write emails (in fact one of them being 4300-word long) to the head of department to request investigation over the massive sea of issues in this course (all of the teaching, coursework, and the exam (both online and in-person) are super bad). As mentioned above, this course is completely a cheap copy over the Stanford and UCL RL courses. Do yourself a favor by self-learning through these better institutions :) and not learn from here.

I would not recommend this module. I ended up getting a good mark in the end, but I think of it as a magical number that appeared for every student taking the module in the end. I put a huge amount of effort in both courseworks and in preparing for the exam, however, there were no appropriate tutorials to help you prepare for the exam and the problems that came up on the exam were literally all unseen for me – I don't remember solving anything similar in the module. We were advised to look at past papers but they also did not have anything similar. The first problem of the exam is multiple choice, but you have to choose some unknown number of correct answers from say 8 possible answers and it is very, very difficult to get it right. As previous comments mentioned, there were many problems with the courseworks and during the exam, which obviously could not have been mitigated fairly for everyone. We did not receive any final feedback explaining what measures were taken to ensure fairness and the mitigations for the exam. - Zlatina, JMC 22/23

## [CO438 Complexity](http://www.imperial.ac.uk/computing/current-students/courses/438)

Really rated the course. Iain isn’t the most engaging lecturer, but his slides are good and he delivers content well. Plenty of problem sheet questions and past papers means that you’ll enjoy this module if you learn by doing. - Andrew (JMC 22/23)

Amazing course. It’s fairly hard but introduces you to a lot of theoretical comp sci that is useful to have in mind when thinking about larger problems

It’s a challenging theory course. Lecture might be a little boring, but the content is super fun (if you’re into it). - Fangyi

Very fun course content with Iain as the lecturer who was nice but could be quite dry sometimes. Coursework was easy if you went to the lectures and did not skip the tutorials, which were very helpful. Probably the only actual course comparable to a fun proof-based course in the maths department (Inductive-based proofs are not proofs). Average exam, but most people found it far easier (?) than I felt it was, although admittedly I skipped almost all of the lectures. - David Angf

Quit halfway... Even though the content seems interesting and useful, please don't consider if you do not have a huge passion for mathematics theory. It took a lot of time in the first half of my term. And then I realised it's not worth for me to sacrifice other modules like that. (21/22)

Personally, I think this course isn’t as mathematical as it seems – all the very dense, formal definitions are frontloaded, and once you’ve wrapped your head around it, the course kinda turns into a series of fun puzzles, i.e. “how can I use an algorithm that solves problem A to solve this other problem B” - it leans far more towards being able to reason about algorithms (particularly about graphs, like Travelling Salesman), than pure maths. Iain is incredibly chill and his slides are (for the most part) nice and clear. - Nat K (21/22)

I thought this was a bit of a boring module at the beginning, but this was because it is not an easy module at first and I just did not understand the content very well. If you take it, prepare yourself to invest more time in the beginning or in the middle of the module to deeply understand what is going on and once you get it, it actually becomes quite interesting. The content is well-organized and there are tutorials to solve – in my opinion they are quite difficult, as well, so you will need to wait for the solutions to come out to understand some of the problems. However, I appreciate that the exam mostly comprised of problems that are similar to what is done in the tutorials and the lectures so if you have worked on the problems given in the module, you will be fine. This was, in fact, the exam that I found the most chill out of all. - Zlatina, JMC 22/23

## [CO445H Advanced Security](http://www.imperial.ac.uk/computing/current-students/courses/445H)

anyone ?

## [CO471 Advanced Issues in Object Oriented Programming](http://www.imperial.ac.uk/computing/current-students/courses/471)

Sophia is a nice lecturer and she explains things in detail. It’s an interesting course if you’re into formal stuff (e.g. semantics, type system, soundness theorems etc.) Exams are pretty deterministic. - Fangyi

## prog[CO496 Mathematics for Machine Learning](http://www.imperial.ac.uk/computing/current-students/courses/496)

Oh man, I have words for this one. First half taught by Mark, stunning. Although the pace is either painfully slow or skips 10 steps in working, he guides you through the concepts at a beginner level (would’ve been nice if he gave some chunky tutorials). Second half is… interesting. You see, his prerecorded lectures were rehashed from his slides, which were rehashed from the 2nd half notes, so you could’ve gotten away with only learning via the notes however the notes are not incredibly comprehensive and also skip steps regularly. This half is more rote learning for the exam questions but really requires you to have discussions and stretch your brain; the tutorial questions are good preparation. There were two cw’s for the 1st half (python programming); we were meant to have two cw’s for 2nd half (matlab) but didn’t due to too many other cw’s. Overall, only take this module if you are good at maths (i.e. taken Stats and CompTech in 2nd year) and you’re willing to put in the time and effort to understanding the content. If you’re thinking of taking it due to prerequisites for other modulesw, please think about your choices carefully – Andy G (online learning)

Ngl, I have no idea why this module is required. Basically no one else othing we learn is used elsewhere.

Anyway, the first half by mark is fine but either has been taught before (into to ML) or is overly mathematical to explain relatively simple concepts. The CWs were fine (albeit quite long) and helped solidify understanding.

Part 2 was basically nothing. I didn’t understand what was going on at all and the lecturer didn’t really explain at all what was going on. His side of the exam was (and is) and complete dumpster fire and usually took about 10 people over 3 hours to answer part a of one of his papers. - Sam Trew (online year)

Mark was not bad at teaching, but the organization of the first half was awful. He released lecture (online year) at an unregular schedule and didn’t provide any practice material up until a few days from the exam. We got the CW marks in late December after all the exams were done. At least the material covered is quite interesting.

I thought that the first part of the module was not great, but I wasn’t prepared for the second one. The lectures felt rushed, and the interaction on piazza was basically non-existent. Only good thing is that the exam questions are recycled. -Davide (online year)

An extremely useful course, which massively helped with second term modules (especially probabilistic inference). Whilst the teaching pace was quite unstructured for the first half, Mark was great and the actual content was very interesting. The second half was a shambles, but he recycles content every year so it’s easy to pick up the second 50% of the paper for free. - Giovanni P (20-21)

Mark’s section of the course is interesting and there are elements of the course which are really useful to know (especially multivariate gradients – it isn’t obvious how useful they are, not just for this module but even in general). While Yingzhen clearly tried, her side was harder to understand – the slides are quite dense and the pace is fast in general (though it could be due to the material too). One issue is that there isn’t much in the way of preparation material and exercises – there is some for Mark’s bit, but barely anything for Yingzhen’s, so it was hard to prepare for her bit of the exam (though Mark’s portion wasn’t easy either). That being said, the TA sessions were kind of helpful even though it wasn’t hard to bring them down to their knees, and both lecturers were active on EdSTEM.

The coursework is short and easy – perhaps *too* easy as they are just LabTS tests and should get you a free 100%. Which is important because the exam was brutal (like a “bloodbath”) – very long with annoying typos and we didn’t have a “recycled” part 2 with the SVM part removed that year and I don’t think I did well at all despite preparing for it.

**February 2022 update**: I’m not sure how this exam was scaled (if at all), but this result for me was the worst by a significant margin for this term and made me quite disappointed. I’m a bit surprised given that the median last year was 70.7, but again, could be just my incompetence. Leaderboard, 21/22

Maths for ML is definitely the worst module I have taken so far during my 4 years at DoC. Even though the topics covered have potential of being interesting, they are taught with few or no relevant exercises and numerical examples at all (especially Yingzhen’s topics, but Mark’s topics aren't much better either). At the same time, the exam is clearly written with an assumption that you can come up with solutions immediately (3 rather lengthy questions to do in 90 minutes). Because learning maths is about “practice, practice and practice”, I have no idea how you can reconcile both of those things. At least the coursework is an easy 100% worth 30% of your grade.

Avoid this module in its current form unless you have to take it (like me) or you’re OK with getting your confidence hurt. - Maks (2021/22)

I didn’t think I was bad at maths, but boy did this module really screw with my confidence. Like everyone above has said, you’ll most likely get 100% in the coursework. But it’s just not worth it because the exam was horrendous. I revised a ton for this module – I did every past paper and every exercise listed on the book, but it still didn’t prepare me well enough for the exam and I fear (alongside a lot of other people) that I’ve failed the module. I only took this module because it’s a prerequisite for other modules, but now I’m hesitant since I don’t want to experience the same distress felt with this module. I put more effort in this module than my other 3 modules combined, which wasn’t wise of me, but I’m more likely to get a distinction from those modules than this module. So yeah, if I was to go back in time, I would tell myself to 100% not take this module. 21/22

Avoid if possible unless completely revised, 21/22

Take it only if you wanted to do maths but somehow ended up in Computing or have to take it as prereq, otherwise avoid at all costs, 2021-2022

21/22, take it as prereq, or don’t!

💀 - 22/23

## [CO499H Modal Logic](http://www.imperial.ac.uk/computing/current-students/courses/499H)

This is like first year logic, but this time with boxes and diamonds. A fun theoretical course if you enjoy these stuff. Proofs are examinable. - Fangyi

After 2019: CO499 = CO498H + CO49[CO422 Computational Finance](http://www.imperial.ac.uk/computing/current-students/courses/422)

Not required to work in finance but helps to understand the world. JMCs should take this, it’s an easy A

Very interesting course - it’s purely maths though, you won’t be doing much “computation”

Would not say its ‘easy’... Unless of course I am not smart enough lol.. Coursework is an easy 90+.

Exam is another story, I think. Lots of proofs via arbitrage and lots of lagrange optimisation

Not the easiest exam. CW, as mentioned, is easy.

An interesting course, and a nice follow-on from ‘Finance and Financial Management – BPES' with a more mathematical focus (a lot of maths that requires practice - start early). The coursework was free marks. I \*thought\* the exam had gone quite well until grades were released... look out for that final question that we weren’t taught that was worth 20% of the paper! This question was actually a tutorial question from Computational Optimisation in second term. - Giovanni P (20-21)

The content is very interesting, with Panos being a competent but not outstanding lecturer (though his use of Zoom irritated me since I don’t use that much). The coursework was fine (questions recycled from past papers) – easy to get 80% but there were an ambiguous subquestion (and I wasn’t the only one that was confused) which made getting 100% a bit harder. The exam was odd – surprisingly high amount of bookwork (+ a subquestion completely recycled from a tutorial) and one subquestion which he barely talked in the lectures *and* the slides, but otherwise reasonable.

What I did *not* like was: the coursework marks being late to release (which was put down to only three TAs for 200 students, though I would still say that then lecturer should have communicated this to us earlier rather than two days before the exam) and the fact that the lecturer told us to hold off questions for the revision lecture and then slyly cancelling it 45 minutes before the lecture citing the strike (which I found in poor taste). Unlike other modules I took, communication on EdSTEM was poor – the lecturer barely used it and the TAs didn’t do much better, with questions remaining unanswered for more than a week times.

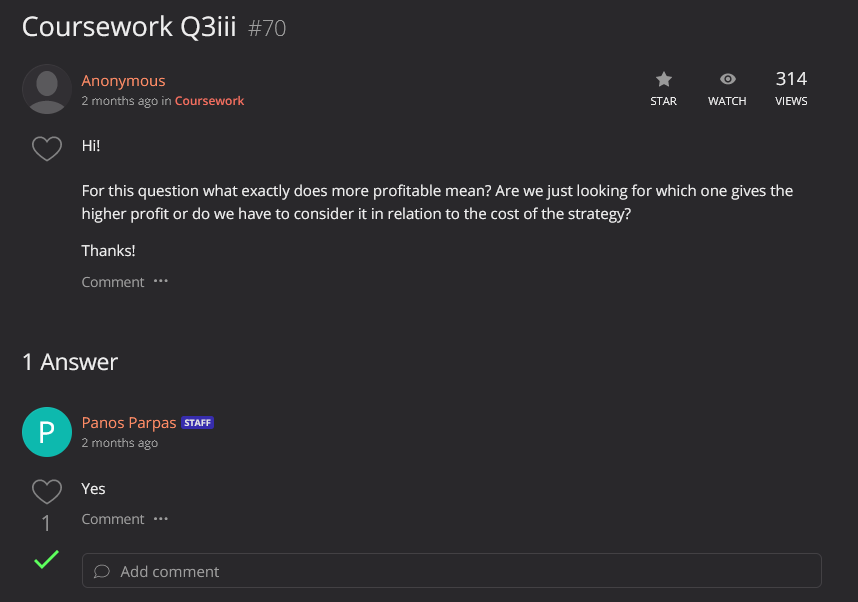
Overall, keeping the above in mind, this is a module I recommend taking for the interesting content. It should be noted that there’s good use of probability and some multivariate calculus (Lagrange multipliers, with some, mostly optional, use of matrix calculus) which familiarity is assumed for. – Leaderboard, 2021/22

Highly recommend it if you are interested in tipping your toes into finance for the first time. The course is well structured, the coursework is not difficult, and the exam is reasonable. Besides, the textbook helps a lot to understand the slides. (21/22)

---- NEW FORMAT ----

Panos’ lectures are unhelpful, full of typos in both slides and in tutorials so it’s hard to find out whether you are right or not. Content is reasonably interesting. Coursework wasn’t too bad although some things unclear. Exam was hard, bookwork questions even though we were guaranteed there weren’t any, essentially a competition of who prepared the best cheat sheet. Most of the content can just be learned from the text books. Panos was unhelpful on EdStem, didn’t answer in any detail and didn’t correct typos very often. Overall, would recommend just learning the content outside of the module if you are interested (22/23)

The course is interesting but it’s definitely NOT WORTH TAKING if you have to deal with Panos. The exam had absolutely nothing to do with any past exam, any tutorial or any in class exercises. He also teaches important things only during the tutorial and nowhere in the slides and does not answer Ed questions. He also has many typos that he never fixes and it makes everything very confusing all the time. (22/23)



I was advised that this module would be a good fit for JMC students. Well, it was not for me. It was very messy, there were many typos in the notes and the tutorials, the tutorials weren't very helpful for the second part of the module and in general the jump in how difficult the content was somewhere in the middle of the module was huge. As a JMC student, I understood what he did related to PDEs and all, but I was surprised how we jumped into PDEs and ODEs without any preliminary stuff being explained. I was very disappointed that there was almost nothing "computational" about this module, after all that is how the module is called. Overall, there was a lot of ambiguity in the lectures and in the tutorials, and the exam was not similar to anything we had done in the tutorials. - Zlatina, JMC 22/23

## [70068 Scheduling and Resource Allocation](https://www.imperial.ac.uk/computing/current-students/courses/70068/)

This module is new for 2021-22. Despite having no prior experience with either part of the course, I liked both sections of this module. The first section (Giuliano) of the module is basically learning different algorithms for scheduling – quite interesting connecting it with things I’d seen in the past such as operating systems (through there was barely any discussion from an OS perspective) and topological sorting (of which scheduling with precedence graphs are just a subset of). The second section (Dario) is similarly interesting even through his section bears little resemblance to Giuliano’s. He focused on the fundamentals of game theory (taking Operations Research can help with parts of this section, though by no means necessary) with the last part being an application to auctions. This section helps with some of those brain-teaser style puzzles one may see (but isn’t the only one) – it wasn’t obvious before what game theory had to do with some of them! As for the lecturers, both were perfectly competent – while neither “wowed” me, I didn’t see anything concerning either.

The coursework was fine – the backend (which wasn’t to be implemented) was quite interesting. Though the 2-page report limit had no good purpose other than to cause irritation when trying to fit the data to the report. Marking was on the lenient side and it’s not hard to get a 100%.

The exam was fine through lengthy – Giuliano’s section was full of algorithm cranking though not hard overall, while Dario’s section had some “sting-in-the-tail” aspects and had a tricky second section.

Overall, this is a good module, and I would recommend taking it – even if you don’t have prior experience in either scheduling or game theory. – Leaderboard, 2021/22

# Spring

## [CO409 Cryptography Engineering](http://www.imperial.ac.uk/computing/current-students/courses/409)

Apparently quite easy

EIE: coding theory covers similar overlaps and more in depth with encoding (Hamming and co)

* Moderate in difficulty – good self-contained lecture notes. The hardest thing for me was understanding the group/ring theory and how it related to some of the concepts – people who have taken a group theory course will have a significant advantage on that part of the module, and the module doesn’t always explain the ideas well. On the other hand, it isn’t directly tested in the exam.
* Michael Huth is OK – competent but not outstanding. Except for a week where he didn’t answer on EdSTEM (and attributed it to EdSTEM not notifying him which I am inclined to believe), he was fairly active on EdSTEM and responded reasonably fast.
* There were also optional Python notebooks prepared by a TA and made reasonably well. The coursework was not hard (3 exam-styled questions to be done in groups of 4), though interestingly no one managed to get full marks (median = 92, highest mark = 99, so still high scoring).
* The exam was not hard either – there is a significant amount of “cranking” in that one question is always about elementary probability and (for the last couple of years at least) one question about Shamir Secret computation (the latter can be a timesink, so use an online calculator for find the inverse of an element in a group).
* Overall, while this was the least interesting module for me this semester, that doesn’t mean that it is bad (I’m just not good with very pure material). – Leaderboard, 21/22

## [CO416 Machine Learnin for Imaging](http://www.imperial.ac.uk/computing/current-students/courses/416)

Pending completion of module: If I could choose again, I wouldn’t take this module. It’s not bad and the lecturer isn’t bad at all. It’s just that there is so little that is new over Computer Vision. Maybe the last 3rd/quarter is new but I just feel so drained from going over the exact same content. Pending as cw hasn’t been done yet and neither has exam – Sam Trew (online year)

Really well-organised course, which is saying something compared to most 4th year courses lol. First half overlaps with Computer Vision a lot, although it tends to go faster. Later there’s more on image segmentation, object detection, unsupervised learning and inverse problems which are pretty interesting in the medical imaging domain. Small part on privacy-preserving ML at the end.

Coursework is very manageable and doesn’t involve too much writing, exam was fairly straightforward. VERY helpful if you just want to learn more about machine learning/have a machine learning project, even if it’s not in imaging. Ben Glocker is great, very easy to understand (again saying something compared to most 4th year lecturers). - Fawaz (online learning, 2020-21)

-- this course wasn’t taught in 21-22 --

Fairly easy course but overlaps a lot with Computer Vision. If you want to learn something new then don’t take this course. If you just want a good grade and took Computer Vision, then take it (although it can be a bit annoying to cover the same thing again, especially since Deep Learning also covers a lot of the same content again). Overall easy, but repetitive. (22/23)

## [CO417 Advanced Computer Graphics](http://www.imperial.ac.uk/computing/current-students/courses/417)

Avoid, it’s hard.

Agree it isn’t easy, but I do think the content is very interesting... That’s not to say I’d like to be examined on it... But in my opinion a flat-out “avoid” is too harsh – James Langley (online year)

Great course. Content is interesting and fyou will learn a lot. Abhijeet is passionate about this stuff and it shows. Exams are difficult and very tight on time – Jamal (online year)

## CO433 Advanced Robotics

Lorem ipsum

## [CO440H Software Reliability](http://www.imperial.ac.uk/computing/current-students/courses/440H)

Used to be a nice full course, combining PL and system sides of SRE. The half course format is rather meh in my opinion, since Alastair no longer teaches, hence less from PL side. A big CW consisting of 33% of mark takes quite a lot of effort, but it is open-ended so you can do a lot of fancy stuff or do the minimal. Exams are weird and the questions may involve unnecessarily small details. - Fangyi

Generally a decent course, and relevant to quite a lot of FYPs. Content and coursework are both a bit heavy, and knowledge of the C specification is definitely a plus to save you having to learn to recognise and identify a lot of specific undefined behaviour examples. Cristian is very meh as a lecturer, sometimes interesting but mostly slow and boring, not to mention that he feels the need to explain every last detail of every example. - 22/23

Personally, I quite enjoyed this module. It was pretty well-contained. From an EIE perspective if you want something CompArch-ey, then this module may be an interesting flavour for you, leveraging C skills or knowledge is quite helpful here, as above. Provides an interesting flavour combining logical reasoning with low-level prog. Courseworks are quite heavy, but definitely quite interesting and allow you to explore the learnings of the course. Final CW being a presentation allows you to show what you know and explore similar content, but definitely make sure you are in a capable group as the first two are heavy (the first one being quite open-ended, so it is difficult to gauge if you’re doing well or very far off). Classes themselves are pretty cool and lecture slides/pace is pretty decent, tutorials are well-laid out, TAs tend to be somewhat active on the Ed and are helpful, and the solutions are well detailed, however be prepared get fiended by semantics, as minor details have to be juggled to reason about types of questions. If you’re a fan of structured textbook modules this one is probably not for you, but you will have to read a fair few academic journals on the (rather niche) topics covered to get a deeper grasp of what’s going on. Certain chatbots will not be able to reason about the content much.. The exam is pretty calm, and standard year on year, however this year it seemed there was a lot more emphasis on information recall than problem solving, so it was more of a case of who prepared the better cheatsheet, be prepared to dive into niche semantics. - Aryan, 22/23.

## [CO446H Applied Network Security](http://www.imperial.ac.uk/computing/current-students/courses/446H)

Lorem ipsum

## [CO447H Advanced Security in Smartphone and IoT Systems](http://www.imperial.ac.uk/computing/current-students/courses/447H)

Lorem ipsum

## [CO460 Deep Learning](http://www.imperial.ac.uk/computing/current-students/courses/460)

Courseworks are really interesting and relevant

Very useful course, with a helpful set of instructors/TAs. The coursework load is a bit heavy with three courseworks, but they’re pretty much guaranteed marks if you put in (a lot of) effort. The content is all very interesting and approachable, though I imagine the second half would have been fairly difficult without taking Probabilistic Inference or NLP - there’s a big overlap in content. The exam was more difficult than previous years due to its size, but overall the coursework secures a large portion of the grade for the module. - Giovanni P (20-21)

n

The first coursework was fine, easy to finish in about a day (if you don’t want to heavily tune your hyperparameters). But the second coursework... Only half of it took at least a week to somewhat finish the first part ~75h. It is heavily reliant on hyperparameter and model architecture tuning. Also, there is a lack of nations of the use of the content taught in the second part, or how it is implemented in the real world.

The course was pretty fine for me, the material was interesting and hugely relevant to my project, but beware:

* The coursework load is **heavy** (especially the second piece about generative models), so you must be prepared for lots of your time being eaten (this is mostly why I struggled with catching up with Probabilistic Inference, which requires lots of practice as well...).
* When I took this course, there were two parts: one done by Bernhard and one done by Yingzhen. Don’t expect the Yingzhen’s part of the exam to be easy. A 35% average for her part in 2020/21 (see the examiner feedback) speaks for itself really.
* Go through some Probabilistic Inference notes (you don’t have to take that module for credit) such as graphical models, Monte Carlo sampling, and variational inference. They will make the Yingzhen’s part a bit easier to understand.

- Maks (2021/22)

## [CO466H Inductive Programming](http://www.imperial.ac.uk/computing/current-students/courses/466H)

Lorem ipsum

## [CO467 Principles of Distributed Ledgers](http://www.imperial.ac.uk/computing/current-students/courses/467)

Arthur has some extreme opinions about how a course should be run. Just keep him in check and it will be fine.

I would say if you have previous experience in Solidity/Web3 stuff (which seemed like some of the cohort had), this course is a nice bang out, I felt that with the new format however (with DeFi content being merged in as the original lecturer left), it felt like at times the content was quite rushed, or sometimes underexplained or neglected, with a lack of understanding over what was examinable and what was not (tutorial lecture boogaloo), however there are plenty of resources existing online such as that on the ETH website or Binance Academy, or even the whitepapers themselves which provide the sufficient conceptual understanding. Pretty self contained with not too much content, but decently packed slides, this course did not consume a lot of my week and required a lower time commitment than others. The coursework was pretty interesting, there was however a slight tone of ambiguity over what is testable, but writing our own unit tests in Foundry was a valuable exercise nonetheless, and the tutorials were engaging, providing an impetus for what we were learning in the course, particularly as we got on to Solidity based stuff. The questions in the exam were pretty standard and you’re not stretched to gargantuan lengths. Definitely amplified my interest in blockchain protocols, would defo take again – Aryan 22/23

## [CO468H Probabilistic Programming](http://www.imperial.ac.uk/computing/current-students/courses/468H)

Lorem ipsum

## [CO449 Probabilistic Model Checking and Analysis](http://www.imperial.ac.uk/computing/current-students/courses/469)

It’s Herbert time again, so get ready for in-class exams, no panopto/Piazza, and tensors. PMC part is basically markov chains with a LOT of different flavours, you actually need to compute stuff with a calculator. First half of PPA is program analysis, similar to the last part of Paul Kelly Compiler. Second half of PPA is magic, although learn classic able because Herbert always sets the same exams. Probabilistic Program Analysis was indeed interesting, but I couldn’t focus in lecturers and slides are full of magic symbols. (To be fair, all PL-course slides are full of them) - Fangyi

In 22/23 it’s called “Program Analysis”. About half based on formal logic/maths-y analysis of programs and half on probabilistic analysis. Herbert can be dull at times, but the course is well prepared, the slides are fairly complete, there is a lot of tutorial material to go through (with tutorial answers given right after tutorial) and the exam is predictable. If you are into formal logic and program analysis this is a well prepared course (unlike when Herbert taught MOC). - Mike 22/23

## [CO475 Software Engineering for Industry](http://www.imperial.ac.uk/computing/current-students/courses/475)

Do it. One of the most valuable courses you can do

Weekly ~~300-word~~ essay and a big essay in the end. Your mileage may vary depending on whether you like this stuff. - Fangyi

har

Hated this course. Don’t think I learnt anything. Also incredibly boring. - Jamal (online year)

The articles can feel like a massive chore at times (make sure you find a good group!), but it is useful for understanding the jargon and how to communicate ideas in the SW industry. If you put in the effort to research ideas and build up a good knowledge base, it can become a vital asset for your career, especially if you are interested in tech leadership - Wilson (21/22)

An interesting enough module. Really appreciated Eoin’s enthusiasm towards the course and that he got in some high-profile guest speakers, although the in-class discussions were a bit boring (most people usually left before them, a bit rude but yeah I get it). The workload is decent considering there is no exam – I generally didn’t put in more than 5hr per week including the lecture – however for me generally the content felt a bit wishy-washy and the weekly tasks / courseworks lacked clarity (most of my time on them was spent discussing what we thought we were actually meant to be doing). Overall I would say this module is highly useful for industry (I’d already encountered a lot of the general themes on placement) but there might be better modules out there if you’re more into theoretical stuff. - 22/23

Definitely take the module if you enjoy software engineering. Each week has its own topic - some of them were pretty interesting (such as software arch and microservices) and some of them were quite boring. An easy module overall that you can do fairly well in with relatively less effort. - Pranav B. (22/23)

## CO477 Computational Optimisation

Ruth is a good teacher and if you’re comfortable with mathematical analysis, the course isn’t too difficult

An interesting and useful course, nice to add some good maths for a non-JMC student. Some of the exercises can appear pretty difficult at first, but with practice they’re very approachable. Small class size, but very helpful TAs who are great on Piazza. - Giovanni P (20-21)

It’s a nice mathematical course, relevant to all sorts of optimisation problems (including machine learning). Actually, it teaches some of the material covered in infamous MML (e.g. constrained optimisation with Lagrange multipliers and KKT conditions), but in a much better way, so take Computational Optimisation if you still want to learn the stuff that MML failed to deliver. - Maks (2021/22)

Agree with Maks – it doesn’t cover *that* much new material over MML but is simply taught better.

* Ruth is above average and teaches quite well (saw some of the positive hype about her which she did meet) – puts Star Wars references that are surprisingly relevant as well. Panos on the other hand is decent but not to the level of Ruth. The material itself can be freaking interesting and juicy for both halves – won’t be surprised if people fall in love with it.
* The tutorials for Ruth’s half were also well organised – and Ruth cleverly managed to cater for both those in-person/hybrid and those watching the recording. What she did was turn off the recording and allow us (in-person/hybrid) to speak and ask for help without it being recorded. Then, she summarised the questions asked and related discussions in the recording – this I found very nice to see of Ruth.   
  I didn’t like how Panos did it conversely – his style of having a TA run the tutorial *while* he idled online waiting for people to ask for help didn’t appeal to me and was difficult to work with in-person (since I would have to get out of the room).
* The coursework was recycled from past papers. That being said, I found it pretty hard to get beyond 85 – there was a nasty subquestion which I struggled a lot and got a large penalty. The marks were returned quite quickly (~ 5 days).
* On that note, preparing for the exams. There are some relevant tutorials and exercises (better than MML for sure) – they help but I found that working through past papers is more important as they helped in solidifying the concepts. They can be pretty hard (and feel brutal) at times – I’ve felt that especially with Panos’ part of the module where he can assume knowledge of non-obvious concepts in some of the exam questions – and some parts of the module have little in relevant exercises. The material can take some time to understand, and practice is important. Knowing the material in the “review” quiz is not enough – and this was admitted by Ruth as the challenge is in combining this material together. The textbook can also be useful.
* Oddly though, this year’s exam was unusually easy – most of Q1 was completely recycled from Q4 of 2018-19 and Q2 was also reasonable. At least it was significantly easier than the previous year’s paper.
* The TAs do a reasonable job – one of them was quite helpful for the coursework while the other two showed up once in a while (and in the TAs). The lecturers themselves were generally inactive on EdSTEM (other than the first week or so).

Overall, as long as you have a good mathematical background (fluency in linear algebra is important, this is a good module to take. MML does help with this one but doing poorly in MML does not mean that this module shouldn’t be taken.

-- Leaderboard, 2021/22

## [CO490H Natural Language Processing](http://www.imperial.ac.uk/computing/current-students/courses/490H)

Pretty interesting, decent coursework too

Bad teaching?

I was sceptical about this course, but it proved to be extremely useful taken in tandem with other modules such as Deep Learning. The ML / DL side of the content is great, for example RNNs, Transformers and some really cutting edge models. The linguistics side of things bored me to be honest, and a lot of the content seemed redundant, but it was worth it for the other bits. The coursework was fairly interesting, but should have a higher coursework weighting in my opinion. The exam content was easy, but ended up being very rushed as it was about 10 pages long! It seems that they thought we had extra time and added another question. - Giovanni P (20-21)

Really great course content-wise, although you will most likely have to teach yourself some bits as the lecturer teaching quality is not great. The course is based roughly on a free textbook (<https://web.stanford.edu/~jurafsky/slp3/> ) so this wasn’t as painful as I thought it would be. The coursework is long (basically a mini-project), but is good if you want to get your hands dirty building a machine learning pipeline from scratch (i.e. learning how to do data cleaning, preprocessing, etc. before all the glamorous models come intoma play). - Fawaz (online learning, 2020-21)

## [CO493 Probabilistic Inference](http://www.imperial.ac.uk/computing/current-students/courses/493)

Pending: Boi oh boi. If this wasn’t compulsory for me, I wouldn’t be taking it. It’s definitely interesting and taught well. But the only other people taking it are JMC or *very* mathsy people. Very maths heavy module, the cws are actually quite nice and help with understanding but just beware for the computing students. Pending as I haven’t done exam. - Sam Trew (online year)

My favourite course I’ve taken in four years at Imperial! (Perhaps I am biased as this is the focus of my thesis). The content is extremely interesting and very useful and is a refreshing change from straight Computing modules. It’s very mathematical, so requires \*a lot\* of practice and interest to do well, but opens your mind to a very interesting area of machine learning. The courseworks are free 100% as they are just LabTS tests. Mark is an incredible lecturer; he teaches very well, goes into a large amount of detail answering any questions and writes personalized essay responses on piazza - it seems he is very dedicated to helping the students. The one downside with this is that a lot of this content, for example in Q&As and suggested reading, is basically examinable and so the size of the course grows large if you want to keep on top of it. The exercises/tutorials appear very difficult at first, but it seems that he makes the exams much easier. Overall great module, and I would recommend to anyone who was comfortable with the maths from Mathematics for Machine Learning. I cannot stress how useful this course was for understanding certain areas in Deep Learning and NLP courses – there is a huge overlap with generative models. – Giovanni P (19-20).

## [CO498H Logics for Strategic Reasoning in AI](http://www.imperial.ac.uk/computing/current-students/courses/498H)

Some issues in 2018-19 but if David Thomas is doing it then at least go to the lectures, he’s really good.

After 2019: CO499 = CO498H + CO499H

## 

40% modal logic, 60% temporal logic (LTL, CTL, CTL\*, eight flavours of ATL), which matched the exam questions. Fun, relatively easy, theoretical course with some proofs (mostly the modal logic part really). Coursework was tedious but quite easy. Exam was somewhat easy, but longer than the individual half modules. Lectures were quite dry so I ended up not going to any, as Belardinelli just read the slides and so can you. - David Ang

Ended up dropping the module halfway. I am very interested in Logic and all modules related to Logic and Reasoning. However, the module was quite boring for me with a massive amount of content – the slides were filled with theoretical stuff with small fontsize and I just could not read them all so I dropped it. Heard that the exam was okay, though, and I found the first coursework okay, but a bit long. - Zlatina, JMC 22/23

## [70066 Decentralised Finance](https://www.imperial.ac.uk/computing/current-students/courses/70066/)

* The workload of this module was the lowest for me by a significant amount. All you need to do was watch a ~1.2 hour video every week, and come for a “discussion” section which was held online (though I think one can safely skip it). The material isn’t hard to understand either – taking Computational Finance isn’t needed but helps with part of this module (in particular, the Traditional Finance lecture can be fully skipped).
* The coursework is a joke. 50% of it was from quizzes which were graded on completion – so all one needed to do to get full marks was to complete them on time (irrespective of the score obtained) as they don’t show up on CATE – and it seemed to me that quite a few found this challenging and forgot to do them. The other 50% was from an individual MCQ coursework which was mostly straightforward – the mean mark of 98.5% (amongst those who completed it) says it all.
* The exam was also a joke. Arthur said that he would make it harder given the high coursework average, but most of the exam seemed to be plain bookwork, with portions also recycled from the quizzes and a practice exam he gave us (this module is new for 2021/22). I’m not sure how they are going to mark it and suspect it may be heavily scaled down. I finished the exam in 40 minutes (out of 120) which felt so *un*-Imperial given how lengthy the average exam at Imperial is.
* The lecturer does a good job on EdSTEM and Arthur is generally quite responsive and helpful. The TAs didn’t interact at all.

-- Leaderboard, 2021/22

## [70030 Knowledge Representation](https://www.imperial.ac.uk/computing/current-students/courses/70030/)

I very much enjoyed this module! Very interesting module if you like Answer Set Programming, and you can build on what you learned in Logic-based Learning if you took this module in Year 3 (but it is not necessary to have taken Logic-based learning). The courseworks were not long and were very fun to do. The content is very well presented and explained, there are tons of examples on the slides and there are also helpful tutorials with solutions. In my opinion, the exam was a bit tricky, but not too bad, so you will be fine. - Zlatina, JMC 22/23

# **Mathematics**

# Autumn

## MATH70005 Optimisation

So incredibly dull. Dante is a very engaging lecturer, notes are clear and lectures very well presented but content is very boring. Every time anything interesting comes up the module moves on. Easy coursework. - Andrew (JMC 22/23)

## PHYS70009 Quantum Information

N.B.: requires QM1 & QM2 as pre-requisities (I also found it useful to grab a copy of Physics Dept Foundations of Quantum Mechanics notes for density matrix stuff - took the module because DoC Quantum Computing was Term 2 and needed this for my thesis) Heard complaints that Myungshik wasn’t a good lecturer but frankly I disagree. He’s very engaging, explains things well and responds to questions positively. He can be slow at times but if that bothers you just 2x Panopto. Unfortunately, tutorial questions aren’t the most informative and the content of the module can be a bit dull (error correction section ends as things start picking up and getting interesting). **This module is quite heavy on Physics knowledge.** There is quite a lot of *physical realisations of quantum computers* stuff. - Andrew (JMC 22/23)

## M4P32 Number Theory: Elliptic Curves

Easy module due to abundance of computations rather than proofs (honestly, given the right prerequisites, only the last few bits are non-trivial, but their proofs are non-examinable anyway). Interestingly, the “elliptic curves” bit is only the second half of the course, since so much time was spent doing prerequisites, namely the p-adic “number theory” bit (which will be used later), the “algebraic geometry” bit (which was done far more thoroughly in M3P20 Geometry I: Algebraic Curves), and the “conics” bit (which I don’t think really matters). The second half is essentially the proofs of two theorems: the Lutz-Nagell theorem and the Mordell-Weil theorem, both used in heavy computations. Coursework was somewhat okay but may need some tricks. Last year’s exam was apparently quite tough but there will always be two computation-based questions (torsion subgroup and rank). No strict prerequisites, but M3P20 Geometry I: Algebraic Curves is helpful. Official notes are available. - David Ang

## M4P52 Manifolds

(Did not really attend the classes here except for the first few and the last few, but were relatively easy to follow when I joined in) Apparently an easy module full of definitions and examples, with no conceptually difficult proofs. Much of the difficulty comes from book-keeping all the details when computing, say the transition maps. Interesting and important material at the end on differential forms which will be used a lot in M4P51 Riemannian Geometry, M4P54 Differential Topology, and M4P57 Complex Manifolds. Was not a fan of the lecturer’s lecturing style though. Official notes are available. - David Ang

## M4P55 Commutative Algebra

Very easy module due to significant overlap with all previous ring theory related courses. No particularly interesting results, but basic results will be used heavily for any number theory or geometry related future studies, say in M4P33 Algebraic Geometry. There was an introduction of topology (Spec R for algebraic geometry) and topological rings (completions of rings for number theory) this year, which spiced it up slightly. Skorobogatov explains the material quite clearly and quite slowly, so it was very easy to follow. Tests were given rather than coursework (I might have convinced him to switch to coursework next year), which were somewhat of a surprise initially. Last year’s exam was apparently relatively doable given his style of questions. The only prerequisite is M3P8 Algebra III, without the last non-commutative bit. Official notes are available. - David Ang

## M4P58 Modular Forms

Personally a hard module, and probably the hardest I have done in four years, but extremely rewarding at the end. Essentially a very superficial introduction to the topic due to its massive depth and breadth, so no important results are proven but some are stated for motivation (modularity, Weil conjectures, Eichler-Shimura theory). Covers the final chapter of Serre’s “a course in arithmetic” for the basic theory and a bit more from other sources. The only strict prerequisites are first/second year linear algebra (eigenspace decompositions, spectral theorem, etc), group theory (orbits, stabilisers, etc), complex analysis (residue theorem, argument principle, etc), but Helm sometimes used results from Fourier analysis (some result idk idc), Galois theory (the main correspondence), representation theory (of characters), etc (which are non-examinable, of course). Interestingly, the courseworks are extremely easy (almost trivial manipulations or first year group theory stuff), and apparently the exams would be similar in style. Official notes are available. - David Ang

## M4P61 Infinite Groups

Relatively easy module, which had a content decrease as compared to last year. Probably more suitably named “geometric group theory”, as it covers the applications of graph theory to group theory and vice versa. Several interesting results were presented, including the Nielsen-Schreier theorem and the fundamental theorem of Bass-Serre theory. The lecturer is very engaging, very enthusiastic, and explains things clearly with lots of pictures. The coursework takes some time to do and was not straightforward. The only prerequisite is basic group theory (orbits and stabilisers), but having done M3P21 Geometry II: Algebraic Topology may help motivate some of the constructions (fundamental groups, free products, amalgamated free products, etc). An interesting construction from M3P65 Mathematical Logic also shows up here (HNN extensions). Official notes are available. - David Ang

## M4P72 Modular Representation Theory

(Only attended the first lecture, but speaking from general consensus and facts) Very hard module, given that M3P12 Group Representation Theory was hard already, and this was a strict prerequisite from spring of the previous year. Everyone I knew who took this said it’s very tough and many dropped it, even halfway into the course. Tests rather than coursework. No official notes are available, but would be nice if we started an informal one. - David Ang

## [MATH70024 - 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

## [MATH70023 - 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 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

## M4P33 Algebraic Geometry (variety theory)

Relatively easy module, even without attending M3P20 Geometry I: Algebraic Curves. Essentially a rehash of that, but for varieties (even more polynomials!), so the entire module was very explicit with many details. Covers affine and projective varieties in detail, very basic algebraic and topological dimension, with singular varieties for mastery material. Buzzard taught the course in 2019-2020, so things went quite weird (e.g. he tried to coerce us to do Lean at the beginning). All lectures were informal, so we had to read the set of notes from 2016-2018 available in Martin Orr’s webpage before each lecture (2018-2019 was a weird year for this course, ignore that). We had oral exams rather than coursework or tests, which involved a 15-minute private Q&A session with him and was slightly intimidating at first. He streamed at twitch.tv/kbuzzard and Discord for the last few lectures due to the virus. Only basic topology (definitions, connectedness, compactness) and basic algebra (rings, ideals, algebraic and transcendence degrees) are prerequisites. The module itself is interesting, useful for future algebraic geometry studies, but not strictly necessary to have a working knowledge as you’ll use the language of schemes in the modern world anyway. - David Ang

## M4P46 Lie Algebras

(Only attended the first lecture, but had friends doing it all the way) Somewhat ok module, and relatively short content-wise. Covers the full classification of lie algebras using root systems and lots of diagrams. Lecturer had an interesting style of teaching, which is quite fast, and may or may not hinder understanding of the material. Tests rather than coursework. Prerequisites include a strong grasp of linear algebra, and preferably M3P20 Group Representation Theory but this is not strictly needed. The module is useful for future study of topological groups and Lie theory, but this is a first course and only covers the algebraic theory without requiring any differential geometry. Official notes are available. - David Ang

## M4P51 Riemannian Geometry

(Only attended the first two weeks, but had a friend doing it all the way) Somewhat ok module, with quite some content. Covers the essentials of differential geometry (differential forms, connections, curvature, metrics, geodesics, some classical theorems) with emphasis towards physics, as the lecturer is a physicist. Apparently a lot of explicit computations where you could get lost. A hard prerequisite is M4P52 Manifolds, as they would assume that you are familiar with basic manifold theory, but doing M3P5 Geometry of Curves and Surfaces also helps, as it is essentially the simpler Euclidean version of the course. The module is useful for further study of differential geometry and general relativity. Official notes are not available, but you can find them easily from previous years. - David Ang

## M4P54 Differential Topology

Relatively hard module, but quite manageable if you did M3P21 Geometry II: Algebraic Topology, and especially manageable if you’re doing it concurrently with M4P57 Complex Manifolds. Covers the basics of manifold theory (including differential forms, integration, and Stokes’ theorem, which is the mastery material for M4P52 Manifolds, so you get this for free), de Rham cohomology (many parts are very similar to the ones in algebraic topology, but with general manifolds), Morse theory and homology (new), and singular homology (which is essentially the same as the one in algebraic topology, but covered faster). The two courseworks are doable. A hard prerequisite is the manifolds module, as they will not go through the main material anymore (except the integration bit), and a soft prerequisite is the algebraic topology module. The module is crucial for any future study of differential geometry and topology, and even their algebraic counterparts. Official notes are available. - David Ang

## M4P57 Complex Manifolds

Very hard module, but doing it concurrently with M4P54 Differential Topology makes it ever so slightly easier, as there is some overlap at the beginning when recapping basic manifold theory, as well all the parts with de Rham cohomology whose facts are assumed. Covers the local theory of complex analysis (Euclidean second year version) very quickly, a recap of differential forms (in the complex case), definitions and examples of complex manifolds (assuming you are very familiar with M4P52 Manifolds), vector bundles (definition, examples, Dolbeault cohomology, tangent bundles, complexification), basics of Riemannian geometry (connections, curvature, metrics), Kahler and Hodge theory (definition of Kahler manifolds, a million different operators on them with lots of identities, Hodge decomposition, Bott-Chern cohomology, Lefschetz decomposition), potentially divisor theory and sheaf theory (the algebraic counterparts), and abelian varieties (Picard groups). The courseworks are tough but short. A hard prerequisite is again the manifolds module, with a complex analytic flavour, with many useful other corequisites such as the differential and algebraic topology courses, as they would assume great familiarity with cohomology and homological algebra. The module is crucial (with the introduction of the notion of a bundle) for any future study of differential and algebraic geometry. Official notes are available. - David Ang

## M4P63 Algebra IV (homological algebra)

Relatively easy module, which does not require much or any knowledge of M3P8 Algebra III. Covers different flavours of modules (projective, injective, free, flat, torsion-free, divisible) with many equivalent definitions and criteria to check these properties, as well as the basics of homological algebra (basic category theory, complexes, resolutions, derived functors). Britnell was quite slow and detailed but managed to cover everything in Skorobogatov’s notes in 2017. The courseworks are hilariously easy (not quite sure why). No hard prerequisites other than basic ring and module theory. The module is useful (with the introduction of exact sequences) for any future study in algebra, geometry, or number theory (especially group cohomology for Galois groups, which is part of the mastery material). Official notes are available on Skorobogatov's website. - David Ang

## [MATH70025 – 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 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

## MATH70051 Vortex Dynamics

Darren Crowdy absolute champ. Super cool module that pretty much only requires vector calculus knowledge. All complex analysis and fluid mechanics knowledge needed can be picked up/is covered. Plenty of past papers and problem sheet questions to help with learning. - Andy (22/23 JMC)

## MATH70022 Finite Elements

Good module. 50% coursework should be fine if you’re JMC (mastery part was a bit peak but overall OK). Colin isn’t the greatest lecurer IMO (very ‘point at the notes on the board and read them out’ type of teacher). But, he’s super responsive on EdStem & email/in person. Content is pretty tough, turns into a Functional Analysis course half way through, but definitely manageable (I haven’t done any functional analysis...) - Andy (22/23 JMC)