**A Note to 2022/23 Freshers**

Dear Future Freshers,

Like the year before ours, the exams were all online and open book. Each theory exam was 80 minutes long, so try to aim for completing each section in an equal amount of time. Here are a few summary notes from our papers that you may find useful.

Also, remember to balance answering “high mark questions” with questions you are good at. Don’t spend a long time perfecting a 5% or 10% question when there is a 40% you can get half the marks in.

The resources below are a combination of those from the previous year, and our year, which we think will help you.

Lastly, we are aware that you may not have online TRAs (Timed Remote Assessments), and the department could decide to revert to in-person exams. Please don’t panic because that won’t help you study. Make sure to practice throughout the year (DO NOT LEAVE REVISION UNTIL APRIL (OR MAY), PLEASE), and learn definitions – it was a common theme in relevant exams before things went online...

<<ANY 2021/22 STUDENTS PLEASE FEEL FREE TO UPDATE THIS WITH ADVICE AND GUIDANCE>>

Good luck,

2021-22 Freshers

P.S. all our exams were in the summer term. I cannot explain how 9 exams in 3 weeks are depressing 🙁

ShapeNotes on 2022 papers:

* 40001 - Introduction to Computer Systems
  + Simple. If you have in person exams, you need to memorise every diagram and table in all the lectures. Including the last ones.
  + Read the question properly: the only constraint on drawing circuits is whatever you MUST use out of the parts available in the question. If something is given but not mandatory, chances are you can find a way to solve the question without it.
  + Multiplexers are nice and convenient, but know when to use and when NOT to use them.
* 40005 - Introduction to Computer Architecture
  + This could have been worse.
  + The only issue with the first half of the exam is forgetting quite simple things that had been taught in the first half of the course. Thinking over potential solutions is important when writing instructions. We were taught both ALGORITHMS and FORMULAS for a reason, please learn them.
  + Honestly, I didn’t find the first half of the paper too difficult given past papers, and the notes on Booth’s Algorithm in the slides. It was quite unexpected though, but there was enough time to learn about the algorithm in the exam.
  + Honestly, Maria could have been harsh and wasn’t. Learn the difference between references and actual values and note when a function call does/doesn’t return a value. The last question is like previous years – don't sweat over memory hierarchy, just practice past papers.
  + Was an easy paper. This is coming from someone who absolutely hated the module. Highly recommend University of Washington’s lectures. Also, if you haven’t done the tutorial sheets, at least do all of Maria’s sheets.
* 40007 - Introduction to Databases
  + Standard, but more difficult overall. Do past papers as the past few have generally followed the exact same pattern.
  + Question 1 was annoying because it was so precise and covered almost everything in that part of the course. Practice using subqueries in SQL, and learn to use ALL, SOME, PARTITION, RANK etc., properly.
  + Question 2 was mixed. Marks distributed rat her evenly, and the schema for 2a was easier than previous years. However, the wording of the description is not kind. 2b can be done with the functional dependency calculator but learn how to write working. For 2c, make sure you understand the last lectures in the course and can replicate anomalies.
* 40008 - Graphs and Algorithms
  + Wasn’t too bad. Covered content from the whole course. Had the exact same format as 2021 and 2020 papers combined.
  + Slightly harder than 2020/2021 imo.
  + Hardest graphs paper we’ve had yet imo – yet again, **still pretty standard – just do past papers and you’ll be fine.** Bottom up might be a good bet to practice
* 40016 - Calculus
  + PRE-EXAM THOUGHTS: This course gave people a lot of grief. We also did not have any practice on a new topic, so if we made it through the exam, you can too.
  + POST-EXAM THOUGHTS: Hell on earth! Y’all are lucky we were the test subjects for Paul’s madness, because he really didn’t give us proper revision materials in time (compared to Linear Algebra).
  + Make sure if you’re practising papers for Calculus, you practice this one first. It is not the same as the other past papers, not even close. He really covered the whole course up to metric spaces, so read, practise and understand EVERYTHING in the notes. Seriously, you need THE WHOLE COURSE for this exam.
  + (If you still have online TRAs) Learn how to use online calculators (Wolfram Alpha, Wolfram Mathematica, even Matlab)!
  + A follow-up to the note above... please learn how to solve questions WITHOUT online calculators. They help, but they won’t teach you how to write proofs or use definitions in the notes to answer questions properly. It also won’t help you remember things like the sequence/series converging to e2 (for instance). This goes for both Linear Algebra and Calculus.
  + Question 1. Can be done. Lengthy, but the proofs aren’t that bad. It helps to remember small facts.
  + Question 2. If I speak... heh, IF I SPEAK!
* 40017 – Linear Algebra
  + Coursework level difficulty. Haven’t done a single past exam question because Chiraag’s April exam revision questions were enough.
  + Just as difficult as last year’s - either doing 3-4 past papers or Chiraag's daily problems is ok. Lots of little things here and there you may forget so it would be super useful to write yourself a crib sheet and test yourself on it (or use it if you have TRAs).
  + In case you don’t get anything from Chiraag next year, hopefully someone will upload his April questions from our year in the Linear Algebra folder. If not, bring it up with him or whoever is your lecturer and ask for some Easter questions. We got one a day :)
* 40018 Part 1 - Discrete Mathematics, Logic, and Reasoning
  + This paper covered Discrete Maths, Propositional Logic, and First-Order Logic.
  + Again, Steffen was kind in Discrete Maths. The early parts are simple, and 2 questions were repeated from previous years. The countability question was very solvable (compared to other questions). A good grasp of general rules and concepts should reduce time spent considerably. All the parts are equally weighted.
  + Propositional Logic was standard. Stays very much in line with other papers. Direct argument involves changing the question to a formula, so don’t disregard simple details or your proof would be invalid. Fair for time.
  + Predicate Logic. Truly kind. Completable within 20 minutes. Pay attention to the wording of the question when it comes to L-structures; our year wanted something slightly different to the year before us. Pandora does help for natural deduction, but this was easy to solve without it 😊
* 40018 Part 2 - Discrete Mathematics, Logic, and Reasoning
  + This paper covered Reasoning about Programs
  + The first half of the exam was very painful. In theory, it’s easy to do... if you revise every type of induction. SERIOUSLY, EVERY SINGLE TYPE. Most of us forgot that structural induction on relations was a thing, and induction on functions was more well-known, but the question was a myth. Lengthy one as well, 7 parts is not nice so be prepared to write a lot.
  + The second half was much better, and Mark was nice with the question. It can be completed, but there was a bit of confusion around writing the invariant. Take your time and use anything the question set-up gives you 😉

ShapeHelpful Resources used:

|  |  |  |
| --- | --- | --- |
| **Course** | **Resource** | **Link** |
| 40017 | Free Wolfram Alpha Pro | <https://www.imperial.ac.uk/admin-services/ict/self-service/computers-printing/devices-and-software/get-software/get-software-for-students/wolfram-alpha-pro/> |
| 40017 | Linear Algebra Toolkit | <http://www.math.odu.edu/~bogacki/cgi-bin/lat.cgi>  <https://textbooks.math.gatech.edu/ila/demos/rrinter.html> |
| 40007 | Functional Dependency Calculator | <http://functionaldependencycalculator.ml/> |
| 40005 | Godbolt Online Compiler | <https://godbolt.org/> |
| 40018 | Logic Crib sheet | <https://www.doc.ic.ac.uk/~imh/teaching/140_logic/Cribsheet.pdf> |
| 40008 | Various algorithms implemented in Haskell | <https://github.com/sorrowfulT-Rex/Haskell-Graphs> |
| 40005 | X86-64 reference page | <https://flint.cs.yale.edu/cs421/papers/x86-asm/asm.html> |
| ALL | Jordan Spooner’s IC First Year Notes | This guy is a king! His notes are a few years old but are \*somewhat\* reliable. That said, please rely on your lecture notes.  <https://github.com/jordanspooner/ic-first-year-notes> |



Text Box