DEPARTMENT OF
COMPUTER SCIENCE

UNDERGRADUATE COURSE HANDBOOK

PRELIMINARY EXAMINATIONS

For students starting in 2023/24

Computer Science
Computer Science & Philosophy
Mathematics & Computer Science

2023

Version 1

Welcome

This is a supplement to the Computer Science Handbook. It is designed to give you all the course-specific information you will need in your first year, complete with all important deadlines.

Please don't hesitate to get in touch with one of the academic admin staff at academic.administrator@cs.ox.ac.uk if you have any questions.

Contents

Disclaimer

This handbook supplement applies to students starting an undergraduate degree in Computer Science, Mathematics & Computer Science or Computer Science & Philosophy in Michaelmas Term 2023. The information in this handbook may be different for students starting in other years.

The Examination Regulations relating to this course are available at

Preliminary Examination in Computer Science

Preliminary Examination in Mathematics and Computer Science

Preliminary Examination in Computer Science and Philosophy

If there is a conflict between information in this handbook and the Examination Regulations then you should follow the Examination Regulations. If you have any concerns please contact the academic admin team at academic.administrator@cs.ox.ac.uk .

The information in this handbook is accurate as at October 2023. It may be necessary for changes to be made in certain circumstances, as explained at www.ox.ac.uk/coursechanges webpage. If such changes are made the department will publish a new version of this handbook, together with a list of the changes, and you will be informed.

Version	Action	Date
Version 1.0	Published start of MT23	

1 Courses

1.1 Computer Science

The Department of Computer Science offers the following degrees in Computer Science at undergraduate level:

- BA Computer Science, 3-year
- MCompSci Computer Science, 4-year

You will initially be entered for the 4-year degree, and will need to decide by early in your third year whether you wish to carry on into the fourth year or leave at the end of the third year with a BA. To proceed into the fourth year (Part C), you will need to have a 2:1 or higher in Parts A and B together.

Course Aims

- To provide a course of the highest academic quality in Computer Science in a challenging and supportive learning environment that attracts the very best students from the UK and elsewhere.

- To provide students with a broad, balanced knowledge of core areas and advanced topics in Computer Science, as defined by the relevant QAA Benchmark Statement.

- To develop in students the ability to evaluate primary evidence critically and the conceptual understanding to marshal and present arguments and solutions based on primary data, relevant theory and the application of sound reasoning.

- To develop transferable skills relating to problem solving and spoken and written communication.

- To bring students to a position on graduation where they can choose confidently from a wide range of careers, both within the Information Technology sector and outside it.

Intended Learning Outcomes

Students will develop a knowledge and understanding of:

- the general theoretical and practical principles of Computer Science;

- a broad range of topics in theoretical Computer Science and the architecture and implementation of information systems;

- relevant mathematical theories and techniques and their application to practical design problems;

- and methods of software development.

The course is in line with the criteria set out in the QAA benchmark statement for Computing. That benchmark statement recognizes the need for diversity of provision in Computing, and the Oxford course remains firmly established at the theoretical end of the spectrum of degree courses.

1.1.1 First year

In the first year of the Computer Science degree, you will take nine lecture courses – eight taught in the Department of Computer Science and one taught in conjunction with Mathematics (with lectures organised by the Mathematical Institute).

Computer Science

- Continuous Mathematics
- Design & Analysis of Algorithms
- Digital Systems
- Discrete Mathematics
- Functional Programming
- Imperative Programming
- Introduction to Proof Systems
- Linear Algebra

Mathematics

- Probability

Tutorials in Computer Science and Mathematics

Tutorials are your main opportunity for developing a deep understanding of a subject – and for sorting out misunderstandings. On average undergraduates can expect to have two or three tutorials or classes per week. Each tutor has particular methods and particular ways of organising tutorials, and will adapt your tutorials to give you the best opportunities for learning. Many tutors will base their tutorials around problem sheets issued by lecturers to accompany their lectures. Your tutor should be the first person you ask for advice on how to study, and later on how to choose optional subjects.

1.2 Mathematics & Computer Science

The Department of Computer Science offers the following joint degrees with the Department of Mathematics:

- BA Mathematics and Computer Science, 3-year
- MMathCompSci Mathematics and Computer Science, 4-year

You will initially be entered for the 4-year degree, and will need to decide by early in your third year whether you wish to carry on into the fourth year or leave at the end of the third year with a BA. To proceed into the fourth year (Part C), you will need to have an average of 2:1 or higher in Parts A and B together.

Course Aims

- To provide a course of the highest academic quality in Mathematics and Computer Science in a challenging and supportive learning environment that attracts the very best students from the UK and elsewhere.

- To provide students with a broad, balanced knowledge of the two subjects of Mathematics and Computer Science, as defined by the relevant QAA Benchmark Statements.

- To develop in students the ability to evaluate primary evidence critically, and the conceptual understanding to marshal and present arguments and solutions based on primary data, relevant theory and the application of sound reasoning.

- To develop transferable skills relating to problem solving and spoken and written communication.

- To bring students to a position on graduation where they can choose confidently from a wide range of careers, both within the Information Technology sector and outside it.

Intended Learning Outcomes

Students will develop a knowledge and understanding of:

- core areas of Mathematics, including the principal areas of Mathematics needed in applications;

- the general theoretical and practical principles of Computer Science;

- the basic ideas of mathematical modelling, particularly as applied to design problems in Computer Science;

- a broad range of topics in theoretical Computer Science and the architecture and implementation of information systems;

- and the basic ideas of a variety of areas of specialisation in Pure and Applied Mathematics and in Computer Science.

The course combines elements from the programmes in Mathematical Sciences and in Computer Science, each of which is in line with the criteria set out in the respective QAA benchmark statements.

1.2.1 First year

In the first year of the Mathematics & Computer Science degree, you will take 11 lecture courses; five in Computer Science and six taught in conjunction with Mathematics (with lectures organised by the Mathematical Institute).

Computer Science

- Continuous Mathematics
- Design & Analysis of Algorithms
- Functional Programming
- Imperative Programming
- Introduction to Proof Systems

Mathematics

- Introduction to University Mathematics
- Introduction to Complex Numbers
- Linear Algebra I and Linear Algebra II
- Analysis I – Sequences and Series; Analysis II – Continuity and Differentiability and Analysis III – Integration
- Probability
- Groups and Group Actions

Course info on Maths courses can be found on the Maths website.

Tutorials in Computer Science and Mathematics

Tutorials are your main opportunity for developing a deep understanding of a subject – and for sorting out misunderstandings. On average undergraduates can expect to have two or three tutorials or classes per week. Each tutor has particular methods and particular ways of organising tutorials, and will adapt your tutorials to give you the best opportunities for learning. Many tutors will base their tutorials around problem sheets issued by lecturers to accompany their lectures. Your tutor should be the first person you ask for advice on how to study, and later on how to choose optional subjects.

1.3 Computer Science and Philosophy

The Department of Computer Science offers the following joint degrees with the Faculty of Philosophy:

- BA Computer Science and Philosophy, 3-year
- MCompSciPhil Computer Science and Philosophy, 4-year

You are initially entered for the 4-year degree, and need to decide by early in your third year whether you wish to carry on into the fourth year or leave at the end of the third year with a BA. To proceed into the fourth year (Part C), you will need to have an average of 2:1 or higher in Parts A and B together.

Course Aims

- To provide a course of the highest academic quality in Computer Science and Philosophy in a challenging and supportive learning environment that attracts the very best students from the UK and elsewhere.

- To provide students with a broad, balanced knowledge of core areas and advanced topics in Computer Science, as defined by the relevant QAA Benchmark Statement, including logic as a natural bridge with Philosophy.

- To enable students to appreciate the interest and importance of philosophical questions on a variety of topics, including links with Computer Science, and to contribute to the discussions of these questions.

- To enhance the understanding of both Computer Science and Philosophy by parallel study of these related disciplines with particular emphasis on the interdisciplinary subjects of logic and philosophy of science.

- To develop in students the ability to evaluate primary evidence critically and the conceptual understanding to marshal and present arguments and solutions based on primary data, relevant theory and the application of sound reasoning.

- To provide a learning environment which draws on the wide-ranging talents and expertise of staff in both Computer Science and Philosophy, and challenges and encourages students, with their differing needs, interests and aspirations, to reach the full potential, personally and academically.

- To develop transferable skills relating to problem solving, as well as promoting the ability to think independently, to develop powers of critical analysis, of sustained argumentation and of clear and effective communication both orally and in writing.

- To bring students to a position on graduation where they can choose confidently from a wide range of careers, both within the Information Technology sector and outside it.

And for students taking the 4-year MCompPhil

- To provide the foundations for graduate study at a leading university, in the UK or abroad, in either Computer Science or Philosophy.

Intended Learning Outcomes

Students will develop a knowledge and understanding of:

- the general theoretical and practical principles of Computer Science;
- a broad range of topics in theoretical Computer Science and the architecture and implementation of information systems;
- relevant mathematical theories and techniques and their application to practical design problems;
- methods of software development;
- selected philosophical texts and central philosophical issues and the concepts needed to discuss those texts and issues in and effective manner;
- and the elements of mathematical logic and philosophy of science.

1.3.1 First year

In the first year of the Computer Science and Philosophy degree you will take ten courses - five in Computer Science, four taught in Philosophy and one taught in conjunction with Mathematics (with lectures organised by the Mathematical Institute).

Computer Science

- Design & Analysis of Algorithms
- Discrete Mathematics
- Functional Programming
- Imperative Programming
- Introduction to Proof Systems

Philosophy

- General Philosophy
- Introduction to Logic
- Elements of Deductive Logic
- Turing on Computability and Intelligence

Mathematics

- Probability

Tutorials in Computer Science and Mathematics

Tutorials are your main opportunity for developing a deep understanding of a subject – and for sorting out misunderstandings. On average undergraduates can expect to have two or three tutorials or classes per week. Each tutor has particular methods and particular ways of organising tutorials, and will adapt your tutorials to give you the best opportunities for learning. Many tutors will base their tutorials around problem sheets issued by lecturers to accompany their lectures. Your tutor should be the first person you ask for advice on how to study, and later on how to choose optional subjects.

Tutorials in Philosophy

For tutorials in Philosophy you will be expected to bring knowledge of the readings which have been set for it (or a variant on your own initiative if some items prove really inaccessible) and almost always an essay in which you address some aspect of the topic covered by the readings.

Work on a tutorial essay in Philosophy involves library searches, reading, thinking, and writing. It should occupy a minimum of three days. Read attentively and thoughtfully, skipping bits that obviously do not bear on your topic: one hour of that is worth many hours of `summarising' paragraph by paragraph. As your reading progresses, think of a structure for your essay (but do not write an elaborate plan which you won't have time to execute). Expect to have to worry out your thoughts, both during and after reading. Use essays to develop an argument, not as places to store information. You can assume that your tutor knows what is in the set reading, and is not interested in a simple re-hash of that. But they will be interested in your critical appraisal of what you have read, and any arguments of your own that bear upon the topic. At the same time, it is important that the relation of what you say in your essay to what you have read can be made explicit if discussion in tutorial turns on it, and for this reason, it is important to include page references to your readings for points you criticise or make use of.

You will learn a lot if you share and discuss ideas with your fellow students, and if you chance your arm in tutorials. Be enterprising, and be prepared to be wrong, for that is how one learns. (Remember that Oxford's system is not one of continuous assessment; it is what you can do at the end that matters, and not the various mistakes that you will inevitably make on the way.) And bear in mind that tutorials are not designed as a substitute for lectures, or for accumulating information, but to develop the capacity to think on the spot and to articulate your thinking clearly in responding to issues raised about ideas in the essay you or a tutorial partner are presenting. This means that note-taking, if it occurs in a tutorial at all, should be incidental to the dialogue. Producing essays for philosophy tutorials gives excellent training in writing, and particularly in writing to a deadline. You will need to equip yourself with a writer's tools, most crucially a dictionary, such as the Concise Oxford Dictionary or the online Oxford English Dictionary, also a thesaurus, and a guide to grammar such as Fowler's Modern English Usage.

2 The Preliminary Examinations

In Preliminary Examination papers for Computer Science and for Mathematics, each question is marked out of 20. Each question will contain some parts of a straightforward nature, and some parts requiring more advanced understanding or an unseen application of techniques or theory from the syllabus. The examination paper will show the marks available for each part.

To pass the Preliminary Examination and progress to Part A, you must attain at least a Pass in each paper.

Please read the Examination Conventions and Notices to Candidates carefully.

2.1 Computer Science

In the Preliminary Examinations for Computer Science you will take the following **four** papers:

- Functional Programming and Design and Analysis of Algorithms
- Imperative Programming and Introduction to Proof Systems
- Discrete Mathematics, Continuous Mathematics and Probability
- Digital Systems and Linear Algebra

Functional Programming and Design and Analysis of Algorithms is of 3 hours' duration and contains eight questions (four on each constituent course); candidates should answer no more than five questions, with no more than three questions from either half of the paper.

Imperative Programming and Introduction to Proof Systems is of 3 hours' duration and contains eight questions (two on Imperative Programming I, three on Imperative Programming 2 and three on Introduction to Proof Systems); candidates should answer no more than five questions, with no more than two questions from any part of the paper.

Discrete Mathematics, Probability, and Continuous Mathematics is of 3 hours' duration and contains nine questions (three on each constituent course); candidates should answer no more than five questions with no more than two from any section.

Digital Systems and Linear Algebra is of 3 hours' duration and contains eight questions (four on Digital Systems, and four on Linear Algebra); candidates should answer no more than five questions with no more than three from any section.

Please see also the course description above and the Examination Regulations here.

2.2 Mathematics & Computer Science

Mathematics & Computer Science candidates take five papers; Functional Programming and Design and Analysis of Algorithms, and Imperative Programming and Introduction to Proof Systems, as described above, and also

- Mathematics I
- Mathematics II
- Continuous Mathematics and Probability

Mathematics I is of 2.5 hours' duration and contains seven questions (four on Part A and three on Part B); you should submit answers to no more than five questions. You should submit answers to no more than three questions from Section A and no more than two questions from Section B.

Mathematics II is of 2.5 hours' duration and contains seven questions (three on Part A, three on Part B and one on Part C); you should submit answers to no more than five questions. You should submit answers to no more than two questions from Section A and to no more than two questions from Section B.

Continuous Mathematics and Probability is of 2.5 hours' duration and contains six questions (3 on each constituent course); candidates should answer no more than four questions with no more than two from each section.

Please also see the course description above and the Examination Regulations here.

2.3 Computer Science and Philosophy

Computer Science and Philosophy candidates take five papers; Functional Programming and Design and Analysis of Algorithms and Imperative Programming and Introduction to Proof Systems as described above, and also:

- Discrete Mathematics and Probability
- Introduction to Philosophy
- Elements of Deductive Logic

Discrete Mathematics and Probability is of 2.5 hours' duration and contains six questions (3 on each constituent course); candidates should answer no more than four questions with no more than two from each section.

Introduction to Philosophy is of 3 hours' duration and contains around fourteen questions (eight in Part A, General Philosophy, and around six in Part B, on Turing on Computability and Intelligence); candidates should answer four questions, including at least one from each section.

Elements of Deductive Logic is of 3 hours' duration and contains typically seven or eight questions; candidates should answer four questions. If you answer more than four questions, your overall mark will be determined by your four best answers.

Please also see the course description above and the Examination Regulations here.

3 Important Dates

3.1 Dates of term 2023-24:

Michaelmas term:	Sunday 8th October 2023 – Saturday 2nd December 2023
Hilary term:	Sunday 14th January 2024 – Saturday 9th March 2024
Trinity term:	Sunday 21st April 2024 – Saturday 15th June 2024

Dates of Full Term for future years are available on the University's website.

3.2 Hand-In Dates – Practicals Reports

Practicals reports:
By noon on Friday of week 5, Trinity term, on Inspera.

4 Recommended Pattern of Teaching

Please compare the list of courses on the Departmental Website. If in doubt, please refer to the website.

4.1 Computer Science

Paper	Term	Faculty Lectures	College Tutorials	Practicals	Comments
Discrete Mathematics (CS3)	MT	16	4		
Functional Programming (CS1)	MT	16	4	Y	
Linear Algebra (CS4)	MT	20	4		
Probability (CS3)	MT	16	4		Taught by the Maths Institute.
Continuous Mathematics (CS3)	HT	16	4		
Design & Analysis of Algorithms (CS1)	HT	16	4	Y	
Digital Systems (CS4)	HT	16	4	Y	
	TT	8	2		
Imperative Programming 1&2 (CS2)	HT	20	4	Y	
Introduction to Proof Systems (CS2)	TT	12	3		

Notes:

- All first year courses are accompanied by tutorials organised by colleges.
- Practicals will start in week 2 of Michaelmas term in your first year. You will usually have an introduction to practicals in week 1.
- There will usually be a number of exercises that you will need to complete for each course. For example, a course with two practical exercises might have a practical timetable as follows:
  - Weeks 2, 4 Classes for first practical exercise
  - Weeks 6, 8 Classes for second practical exercise

4.2 Mathematics and Computer Science

Paper	Term	Faculty Lectures	College Tutorials	Practicals	Comments
Analysis	MT	15	4		Taught by the Maths Institute.
	HT	16	4		
	TT	8	2		
Functional Programming (CS1)	MT	16	4	Y	
Introduction to University-Level Mathematics	MT	16			Taught by the Maths Institute.
Introduction to Complex Numbers	MT	2			Taught by the Maths Institute
Linear Algebra	MT	14	4		Taught by the Maths Institute.
	HT	8	2		
Probability (CS3)	MT	16	4		Taught by the Maths Institute.
Continuous Mathematics (CS3)	HT	16	4		
Design & Analysis of Algorithms (CS1)	HT	16	4	Y	
Groups and Group Action	HT	8	2		Taught by the Maths Institute.
	TT	8	2		
Imperative Programming Parts 1 and 2 (CS2)	HT	20	4	Y	
Introduction to Proof Systems (CS2)	TT	12	3		

Notes:

- All first year courses are accompanied by tutorials organised by colleges.
- Practicals will start in week 2 of Michaelmas term in your first year. You will usually have an introduction to practicals in week 1.
- There will usually be a number of exercises that you will need to complete for each course. For example, a course with two practical exercises might have a practical timetable as follows:
  - Weeks 2, 4 Classes for first practical exercise
  - Weeks 6, 8 Classes for second practical exercise

4.3 Computer Science and Philosophy

Paper	Term	Faculty Lectures	College Tutorials	Practicals	Comments
Discrete Mathematics (CS3)	MT	16	4		
Functional Programming (CS1)	MT	16	4	Y	
General Philosophy	MT	8			
Introduction to Logic	MT	8			
Probability (CS3)	MT	16	4		Taught by the Maths Institute
Design & Analysis of Algorithms (CS1)	HT	16	4	Y	
Elements of Deductive Logic	HT	8			
Imperative Programming 1&2 (CS2)	HT	20	4	Y	
Introduction to Proof Systems (CS2)	TT	12	3		
Turing on Computability and Intelligence	TT	8			

Notes:

- All first year courses are accompanied by tutorials organised by colleges.
- Practicals will start in week 2 of Michaelmas term in your first year. You will usually have an introduction to practicals in week 1.
- There will usually be a number of exercises that you will need to complete for each course. For example, a course with two practical exercises might have a practical timetable as follows:
  - Weeks 2, 4 Classes for first practical exercise
  - Weeks 6, 8 Classes for second practical exercise