WHTC University application subject guides

Computer Sciences

As a Computer Science student at university, you'll study the theory of software, learn how to design it and become a top problem-solver. Our university rankings include Artificial Intelligence, Games, Health Informatics and Software Engineering.

This guide has been written to help support you in your application to university. It contains the following information relevant to your subject to help you decide where to apply and put together the best application that you possibly can:

Contents Page

---

1. Course links

Below are links to the top courses for this subject in the UK (according to The Complete University Guide). Click on the links to find information about what the course is like, what you'll learn, and loads of information about things such as fees and accommodation. However, remember that there are loads of other great universities out there, so check out The Complete University Guide or just google studying your subject at university.

1. University of Cambridge
2. University of Oxford
3. University of St Andrews
4. Imperial College London (BEng computing)
5. University of Durham

6. University of Birmingham
7. King's College London, University of London
8. University of Nottingham
9. University of Sheffield
10. UCL (University College London)
11. Royal Holloway, University of London
12. University of Leicester
13. Queen Mary University of London
14. Brunel University
15. University of Essex

---

2. Entrance requirements & Recommended A levels

Here are the grades that the university suggests you need to get in to that course, and the likely offer that they will give you.

1. Cambridge: A*A*A
2. Oxford: A*AA with A in mathematics
3. St Andrews: AAA with A in mathematics and a science subject or equivalent
4. Imperial College London: A*A*A with A* in mathematics
5. Durham: A*AA including mathematics
6. Birmingham: AAA including mathematics
7. King's College London, University of London: AAA including mathematics or Further Mathematics
8. University of Nottingham: AAB*-AAA *including A in Computer Science
9. University of Sheffield: AAA or AAB including A in mathematics and Computer Science
10. UCL (University College London): A*A*A including A* in Mathematics
11. Royal Holloway, University of London: ABB-AAB including Mathematics or Physics. EPQ will be taken into consideration
12. University of Leicester: ABB
13. Queen Mary University of London: AAB-ABB*-* Grade B or above in Mathematics
14. Brunel University: BBB
15. University of Essex: BBB

3. Admissions tests

What admissions tests are you typically required to sit in addition to submitting your application? This also differs from uni to uni, so if your chosen university isn't on this list, make sure you check out the course page so you know exactly what you need to apply.

1. Cambridge: applicants for Computer Science are required to take the pre-interview Cambridge Test of Mathematics for University Admission (CTMUA) and some colleges require the Computer Science Admissions Test (CSAT) at interview
2. Oxford: Maths Admission Test (MAT)
3. St Andrews: None
4. Imperial College London: None
5. Durham: None
6. Birmingham: None
7. King's College London, University of London: None
8. University of Nottingham: None
9. University of Sheffield: None
10. UCL (University College London): None
11. Royal Holloway, University of London: None
12. University of Leicester: None
13. Queen Mary University of London: None
14. Brunel University: None
15. University of Essex: None

4. Recommended reading

Reading some relevant books or articles is a really great way to demonstrate your passion for your chosen subject in your personal statement, and show how you've gone beyond the curriculum. Plus, if you really want to spend three years or more studying this subject at university, it should be enjoyable! Try taking notes and jotting down your thoughts as you're reading so that you can share some of this in your personal statement.

The Cambridge University site recommends:

The most important preparation is to build up a broad background understanding of issues in computer science. There are a range of useful books. An excellent informal collection of accessible and relevant articles is:

- The new Turing omnibus, A Kee Dewdney, Palgrave Macmillan, 2003, ISBN 978-0805071665.

Remember to try the exercises at the end of each chapter.

- Computational Thinking by Jeannette Wing of Carnegie-Mellon University

Computer science relies heavily on mathematics, not only for formal proofs but also as the language used to describe almost every aspect of the subject. You will need to be fluent in mathematics and familiar with the ideas of formal proof.

An excellent introduction is:

- How to think like a mathematician, Kevin Houston, Cambridge University Press, 2009,ISBN 978-0-521-71978-0

Don't be misled by the title; this book is absolutely relevant for computer scientists. It includes many worked examples and also illustrates common mistakes.

- Computing with Quantum Cats – John Gribbin
- The Pleasures of Counting – T.W. Korner
- Algorithmics- The spirit of Computing – David Harel
- Introduction to Algorithms – T. Cormen
- Code: the Hidden language of Computer Hardware and Software – James Petzold
- The Mythical Man-Month: Essays on Software Engineering - Frederick P Brooks Jnr
- The Emperor's New Mind: Concerning Computers, Minds, and the Laws of Physics (Oxford Landmark Science) Paperback – Sir Roger Penrose
- Artificial Intelligence: A Modern Approach: International Edition Paperback – Stuart Russell.

It is worth keeping up with advances in science more generally. Online publications such as ScienceDaily, PopSci, QuantaMagazine and recently published articles from Elsevier ,

Scientific American and New Scientist cover a wide range of topics in an accessible style, and often have articles relating to computer science. Of course, there are also many web sites that carry technical news.

Finally, you might like to look at some recommended text books for first-year lecture courses.

Databases:

- Lemahieu, W., Broucke, S. van den & Baesens, B. (2018) Principles of database management.

Digital Electronics:

- Harris, D.M. & Harris, S.L. (2013). Digital design and computer architecture. Morgan Kaufmann (2nd ed.). The first edition is still relevant.
- Katz, R.H. (2004). Contemporary logic design. Benjamin/Cummings. The 1994 edition is more than sufficient.
- Hayes, J.P. (1993). Introduction to digital logic design. Addison-Wesley.

Discreet Mathematics:

- Biggs, N.L. (2002). Discrete mathematics. Oxford University Press (Second Edition).
- Davenport, H. (2008). The higher arithmetic: an introduction to the theory of numbers. Cambridge University Press.
- Hammack, R. (2013). Book of proof. Privately published (Second edition). Available at: http://www.people.vcu.edu/ rhammack/BookOfProof/index.html

Foundations of Computer Science:

- Okasaki, C. (1998). Purely functional data structures. Cambridge University Press.

Introduction to Graphics:

- Foley, J.D., van Dam, A., Feiner, S.K. & Hughes, J.F. (1990).
- Computer graphics: principles and practice. Addison-Wesley (2nd ed.).
- Shirley, P. & Marschner, S. (2009). Fundamentals of Computer Graphics. CRC Press (3rd ed.).

Object Orientated Programming:

- Deitel, H.M. & Deitel, P.J. (2009). Java: How to Program. Prentice Hall (8th ed.).

Algorithms:

- Cormen, T.H., Leiserson, C.D., Rivest, R.L. & Stein, C. (2009). Introduction to Algorithms. MIT Press (3rd ed.). ISBN 978-0-262-53305-8

Interaction design:

- Preece, J., Rogers, Y. & Sharp, H. (2015). Interaction design. Wiley (4th ed.).

Software and Security Engineering:

- Anderson, R. (2008). Security engineering (Part 1 and Chapters 25-26). Wiley. Available at: http://www.cl.cam.ac.uk/users/rja14/book.html

For more information, please follow the link to Part IA and click on the Syllabus tab for any of the lecture courses listed there: www.cl.cam.ac.uk/teaching/current/.

Cryptography I

5. Interesting MOOCs

Another great way of learning more about your chosen subject and demonstrating your interest is to take a MOOC, or Massive Open Online Course. These are free courses delivered by universities that you can take online. Try looking at Class Central - they have a huge list of different courses and they're all free!

CS50's Introduction to Computer Science:

This is an entry-level course taught to teach you how to think algorithmically and solve problems efficiently.

Topics include abstraction, algorithms, data structures, encapsulation, resource management, security, software engineering, and web development. Languages include C, Python, SQL, and JavaScript plus CSS and HTML.

Elements of AI

This course will enable you to be able to:

- Understand some of the major implications of AI
- Think critically about AI news and claims
- Define and discuss what AI is
- Explain the methods that make AI possible

Mathematics for Computer Science

- The course begins with a detailed discussion of how two parties who have a shared secret key can communicate securely when a powerful adversary eavesdrops and tampers with traffic.
- You will be exposed to many exciting open problems in the field and work on fun (optional) programming projects.

Introduction to Computer Science and Programming using Python – edX
This course will enable you to learn and understand:

- A Notion of computation
- The Python programming language
- Some simple algorithms
- Testing and debugging

- An informal introduction to algorithmic complexity
- Data structures

Click here to access more courses related to Computer Science.

6. Related courses

Generally, computer science is taken as a single subject but some Universities allow a combination with other subjects.

1. University of Oxford
    - Computer Science and Philosophy
    - Mathematics and Computer Science
2. University of St Andrews
    - Computer Science and Biology
    - Computer Science and Economics
    - Computer Science and Mathematics
    - Computer Science and Philosophy
    - Computer Science and Psychology
    - Computer Science and Statistics
3. Imperial College London (BEng computing)
    - Computer Science and Mathematics
4. University of Birmingham
    - Computer Science and Artificial Intelligence
    - Computer Science and Mathematics
    - Computer Science with Digital Technology Partnership (PwC/Vodafone)
5. King's College London, University of London
    - Computer Science and Management
6. University of Nottingham
    - Computer Science and Artificial Intelligence
    - Computer Science and Cyber Physical Systems
7. University of Sheffield
    - Computer Science and Artificial Intelligence
8. Royal Holloway, University of London

- Computer Science and Artificial Intelligence
- Computer Science and Information Security
- Computer Science and Software Engineering
- Computer Science and Mathematics

9. University of Leicester
    - Computer Science with Enterprise
10. Queen Mary University of London
    - Computer Science and Mathematics
    - Computer Science with Management (ITMB)
11. Brunel University
    - Computer Science and Artificial Intelligence
    - Computer Science and Digital media and Games
    - Computer Science and Network Computing
    - Computer Science and Software Engineering
    - Computer Science with Mathematics

---

7. Oxbridge example interview questions

As you will know, applicants to Oxford and Cambridge have to take an interview in order to get a place. It is normal to get open-ended questions, as well as being given charts or pieces of writing to analyse. Here is a sample of the kind of questions you might get asked. Remember, you're supposed to not know the answer! They often deliberately choose topics that they think no one will have studied in order to make the questions fair. What they're looking for is to see how you think under pressure, and how you can present your ideas and your logic.

Some general hints before reading the interview questions:

- Mathematics, philosophy, computer science, psychology and many other subjects require a persistent and precise mode of reasoning. In these interviews, one

question could take a significant amount of time and energy to work through, perhaps even requiring repeated iterations of the same method.

- If the problem contains specific numbers (like 10, 100, 23, 34), does it become easier if we replace those numbers with smaller ones, or even by 0 or 1 or 2? If there are no specific numbers, can you solve the problem in small examples, such as a 2 x 1 bar of chocolate?

- Are there other simplifying assumptions that you can try? What if the bar of chocolate consists of just one row of squares? What if the green square is in one corner?
- Is there a way of reducing the problem as given to a smaller one? Is there a way of filling the first box of blocks that eliminates a colour, leaving us with 9 boxes and 9 colours?
- Some of these problems have definitive answers, other do not – or not answers that can be reached during a half-hour conversation, anyway. Most of them can be solved in several stages, beginning with easy cases and getting more general; some problems can be generalised beyond what is asked in the question.

The Oxford admissions office publish an excellent booklet that explains in general terms what happens when you come for interview at Oxford. Most of that booklet applies just as well to Computer Science as it does to other subjects.

Please see Cambridge Interview general related questions below:

"How can I prepare when the interviewer could ask me absolutely *anything* about Computer Science?" *By understanding how the interview works and, crucially, what it is that the interviewer is looking for. The interviewer is not looking to catch you out, but rather for you to demonstrate your curiosity, knowledge and passion for Computer Science.*

"How am I able to do that?" *Demonstrate your enthusiasm and personality*

You might be asked general interview questions so that the interviewer can learn more about you – review the list of General Interview questions to prepare.

Show that you enjoy studying Computer Science independently

The easiest way to demonstrate your enthusiasm for Computer Science is to show that you are self-motivated and have studied the subject in your free-time for enjoyment – for example through online lectures and independent reading. If you don't know where to start, review our suggested reading list below.

Demonstrate your subject knowledge about and passion for Computer Science

The key to answering these questions is to always demonstrate your thought process aloud.

The interviewer does not expect you to be able to answer every question immediately, but rather wants to determine that you are able to think about and work on unknown topics with confidence, intelligence and clarity – and they won't be able to do that if you sit in silence! Use

the list of questions below to prepare. Perhaps you can have a friend or relative ask you these

questions so that you can develop your skills of thinking under time pressure and speaking out loud.

Please see Oxbridge sample interview questions written in 2019 by Dr Helen Fryer postdoctoral researcher in the Department of Zoology, University of Oxford.

Please see Oxford sample interview problem solving questions.

Please see Cambridge sample interview questions

Sample Question:

Interviewer: Brian Harrington, Keble College

How do pirates divide their treasure?

A group of 7 pirates has 100 gold coins. They have to decide amongst themselves how to divide the treasure, but must abide by pirate rules:

- The most senior pirate proposes the division.
- All of the pirates (including the most senior) vote on the division. If half or more vote for the division, it stands. If less than half vote for it, they throw the most senior pirate overboard and start again.
- The pirates are perfectly logical, and entirely ruthless (only caring about maximizing their own share of the gold).

So, what division should the most senior pirate suggest to the other six?

This is a standard logic problem and is a good example of the type of question that could be asked. I like to see how students can take directions, and if they can break problems into smaller subsets, and work through a complex concept applying a solution in an algorithmic way. If students have any questions, I want them to ask – not to sit in silence feeling stuck!

See the solution to this problem.