

AS and A-Level Computer Science

Entry Requirements:

GCSE Grade 6 in Computer Science/Computing/Computer Studies **and** GCSE 6 in Maths.

For students from schools where Computer Science GCSE is **not delivered**, students may apply with a Grade 6 in GCSE Maths and with a grade 6 in GCSE Physics, or where students completed Combined Science with a Grade 6 in GCSE Maths and a Grade 6-6 in GCSE Combined Science. Tapton students who did not take GCSE Computer Science may also apply with this entry requirement.

Exam Board: OCR H046/ H446

Subject Leader: Mrs S Thomas sthomas@taptonschoool.co.uk

Main Syllabus Area: The OCR Specification has three components

<https://www.ocr.org.uk/qualifications/as-and-a-level/computer-science-h046-h446-from-2015/>

The content of the **AS Level** in Computer Science and the **first year of the course** are divided into two components:

- **AS Level (H046/01) Computing Principles Component (01)** 50% of marks (70) - contains the majority of the content of the specification and is assessed in a written paper recalling knowledge and understanding.
- **AS Level (H046/02) Algorithms and Problem-Solving Component (02)** 50% of marks (70) - relates principally to problem solving skills needed by learners to apply the knowledge and understanding encountered in the Computing principles component.
- This specification has been designed to be co-teachable with the stand-alone A-Level in Computer Science.
- **Mathematical skills** are embedded throughout the content of the two components and assessed in the written papers where appropriate. The quality of extended responses is assessed in the written papers where indicated by an asterisk.

The content of the **A-Level** in Computer Science and the second year of the course is divided into three components:

- **A-Level (H446/01) Computer Systems Component (01)** 40% of marks (140) - contains the majority of the content of the specification and is assessed in a written paper recalling knowledge and understanding.
- **A-Level (H446/02) Algorithms and Programming Component (02)** 40% of marks (140) - relates principally to problem solving skills needed by learners to apply the knowledge and understanding encountered in Component 01.
- **A-Level (H446/03 or 04) Programming Project Component (03 or 04)** 20% of marks (70) - is a practical portfolio-based assessment with a task that is marked.
using levels of response style mark schemes and in the Evaluation section of the Programming project component.

Method of Assessment

- Throughout the course students will take part in peer-group and self-assessment in order to familiarise themselves with the Assessment Objectives and become more objective in their skills of critique.
- Each half term students have a formal assessment, which is marked and then fed back on in the lesson.

- Staff review work with students on a one-to-one basis regularly, giving support, feedback, and set targets for improvement and progression.
- Students who take the AS exams at the end of the first year are externally assessed.
- Students who take the end of year exam in June have 2 internally marked and moderated AS exams in Component 1 and Component 2. These are taken in full exam conditions, (see above)
- All A-Level exams are externally assessed.
- The A-Level coursework unit is marked by the teacher and marks are moderated by an assessor from OCR.

Qualities Required

You are enthusiastic, open-minded and willing to learn new skills and techniques. You are inquisitive and enjoy computer programming from designing and creating websites to creating apps and programs. You enjoy working as part of a group but can work independently. You are determined and will work to ensure you succeed. You understand the need for resilience and always wanting to learn more by putting in the time and effort, coaching lower school students and getting involved in Computer Science projects within the school.

Links with other subjects

Computer Science involves many **transferable skills**, not least problem solving, communicating ideas and concepts, developing creative ideas, refining and testing ideas and realising those ideas. In addition, an A-Level in Computer Science ties in well with all the STEM subjects but can also be combined with Arts, Social Sciences and the Humanities.

Career Prospects

As more and more companies and organisations undergo digital transformation, as automation and machine learning develop at an exponential rate, the demand for tech workers continues to grow at a pace unmatched in other industries. An A-Level in Computer Science gives students essential knowledge, problem solving skills and skills in demand across all sectors. Specific computer-based roles that students can aim towards are Applications Developer, Cyber Security Analyst, Data Analyst, Database Administrator, Forensic computer analyst, Game designer/developer, Information systems manager, IT Consultant, Machine Learning Engineer, Multimedia Programmer, Penetration Tester, SEO Specialist, Software Engineer, Systems Analyst, UX Designer and IOT Engineer and

Designer and let's not forget research and development in universities and of course teaching.

Computer programming is quickly becoming an expected 21st Century literacy, but coding is no longer limited to the realms of computer and information sciences. Technology can be used to solve problems across a range of fields.

Extension and Enrichment Opportunities: Our key enrichment activity is to connect our students with employers and industry, through work experience opportunities, talks, digital events, hackathons, trips and workshops. We are also keen for students to support learning in Lower School Computer Science Classes, help at Computing Clubs and taking part in any Computer Science events or projects.

Reading List Year 11 - Preparing for Sixth Form Computer Science

By far the best way to prepare for Year 12 Computer Science is to develop your confidence as a programmer in Python or JavaScript using an online resource such as Code Academy or W3C see below. A great understanding of computer architecture will also help and "How Do It Know" is a clear and accessible way to develop your knowledge in this area.

A great book for practical coding in python, from beginner to intermediate level, is Learning to Program in Python – it takes you right through the basics. We have some copies in the department so get in touch with me and it may be possible to post one out.

Learning to Program in Python by PM Heathcote. PG Online

<https://www.pgonline.co.uk/resources/computer-science/gcse-aqa/learning-to-program-in-python/>

I have also looked for some books that are fun and inspirational. Hopefully, they will prompt you to read around the subject and remind you why you have chosen to study this fascinating, beautiful subject that impacts every area of our lives, especially at the moment.

- **Computational Fairy Tales** by Jeremy Kubica. ISBN: 978-1477550298:
"A romp through the principles of computational thinking, illustrating high-level computer science concepts, the motivation behind them, and their application via the medium of a fairy tale. Aimed at secondary school students. "Bonkers, but very enjoyable."
- **Python Crash Course, 2nd Edition:** A Hands-On, Project-Based Introduction to Programming Paperback – May 3, 2019, by Eric Matthes (Author) ISBN-10: 1593279280 *The best-selling Python book in the world. A fast-paced, no-nonsense guide to programming in Python. This book teaches beginners the basics of programming in Python with a focus on real projects.*
- **But How Do It Know? - The Basic Principles of Computers for Everyone** Clark Scott (Author) ISBN- 10: 0615303765 *But How Do It Know? really explains how computers work. They are far simpler than anyone has ever permitted you to believe. It contains everything you need to know, and nothing you don't need to know. No technical background of any kind is required.*
<http://www.buthowdoitknow.com/>
- **The Code Book: The Secrets Behind Codebreaking**, By Simon Singh, Category: Teen & Young Adult Nonfiction ISBN-10: 0385730624 In his first book since the bestselling Fermat's Enigma, Simon Singh offers the first sweeping history of encryption, tracing its evolution and revealing the dramatic effects codes have had on wars, nations, and individual lives. From Mary, Queen of Scots, trapped by her own code, to the Navajo Code Talkers who helped the Allies win World War II, to the incredible (and incredibly simple) logistical breakthrough that made Internet commerce secure, The Code Book tells the story
- **How to Think Like a Computer Scientist** by Peter Wentworth, Jeffrey Elkner, Allen B. Downey, and Chris Meyers
<http://openbookproject.net/thinkcs/python/english3e/>
- **Learn Python or a SECOND LANGUAGE using a structured online course or book**
Languages: Code Academy <https://www.codecademy.com/>

C++, Java script etc.

Independent Study

1. **CODE:** Develop programming techniques by independent coding using any of the numerous coding websites such as time2code and w3c.org and the <https://www.raspberrypi.org/>
Even better if you start a coding project such as a game or APP.
2. **CORNELL NOTES:** Consolidating Cornell notes using the Course Textbook and Isaac Computing
3. **PAST PAPERS:** Complete and mark past papers using resources from the exam board websites at OCR.org.uk
4. **Read:** BBC technology, Wired,
5. **Listen:**

- a. MIT Open podcasts: <https://news.mit.edu/podcasts>
- b. Wired Podcasts www.wired.com › podcasts WIRED Podcasts
- c. The Machine: A computer science education podcast
<https://creators.spotify.com/pod/profile/the-machine/>