

Computer Science as a school subject Seizing the opportunity

Computing at School Working Group
<http://www.computingschool.org.uk>
March 2012

© Copyright 2012 Computing At School

This work is licensed under the Creative Commons Attribution-Non Commercial license;
see <http://creativecommons.org/licenses/by-nc/3.0/> for details.

*"I was flabbergasted to learn that today computer science isn't even taught as standard in UK schools," he said. "Your IT curriculum focuses on teaching how to use software, but **gives no insight into how it's made.**"*

- Eric Schmidt, Chairman Google, McTaggart Lecture, August 2011

*"I think Eric Schmidt is right... we're not doing enough to teach the next generation of programmers. One of the things you hear from the businesses here in Tech City is **"I don't just want people who are literate in technology, I want people who want to create programs"**, and I think that's a real wake up call for us in terms of our education system."*

- David Cameron, Tech City interview, November 2011

"We're encouraging rigorous Computer Science courses

"The new Computer Science courses will reflect what you all know: that Computer Science is a rigorous, fascinating and intellectually challenging subject. Computer Science requires a thorough grounding in logic and set theory, and is merging with other scientific fields into new hybrid research subjects like computational biology.

"So I am also announcing today that, if new Computer Science GCSEs are developed that meet high standards of intellectual depth and practical value, we will certainly consider including Computer Science as an option in the English Baccalaureate.

"Although individual technologies change day by day, they are underpinned by foundational concepts and principles that have endured for decades. Long after today's pupils leave school and enter the workplace – long after the technologies they used at school are obsolete – the principles learnt in Computer Science will still hold true."

- Michael Gove's speech at BETT January 2012

Sometime in the early 1980's computers appeared in UK schools and a generation of children were taught how to program them. That generation grew up to make the UK a world leader in computer related technologies. Today the picture is very different: with the best of intentions, we have lost the "how it works" part, in favour of "how to use it".

There is a growing recognition that this loss is to the detriment both of our young people's education, and of our nation's prosperity. Eric Schmidt's remarks, and Michael Gove's subsequent speech (both quoted above) call for the re-establishment of Computer Science in the school curriculum.

This briefing note, intended for Governors and Senior Management Teams, fleshes out the background; it intended as a basis for a strategic debate about reform of the ICT curriculum. It has been written mainly with schools in mind, but the broad outline of the argument applies equally to Sixth Form and Further Education Colleges.

Executive summary

Computer Science is the fourth science. The Government is now encouraging every good school to offer Computer Science as part of their curriculum, from primary school onwards. Michael Gove has indicated that, in future, it may well become a contributor to the English Baccalaureate. What is the thinking behind this announcement? What has brought it about? And how can schools respond?

- **Computer Science is a discipline**, like Maths, Physics, or History. It has a body of knowledge, established techniques, and thinking skills, that will last students a lifetime. The core skill-set of Computer Science is independent of new technologies and programming techniques.
- **Computer Science is a school subject**, not just a university-level discipline. We have ample evidence that, with the appropriate tools, it is accessible to children from primary school onwards. Many of the current leaders in the field, first encountered the subject when it was taught in schools using BBC micros.
- **Computer Science is educationally important**. Just as we give every student the opportunity to learn the workings of physics, chemistry, and biology, because they live in a physical, chemical, biological world, so we should offer every student the opportunity to learn the workings of the digital systems that pervade their world. This knowledge is empowering, enriching, and inspiring; the skills involved readily transferable. Writing a computer program, while seemingly esoteric, is the closest a child can come to thinking about thinking. Likewise, debugging a program is the closest one can come to learning learning. Amongst other things, Computer Science embodies logic, rigour and problem solving. Some commentators have dubbed it ‘the new Latin’.
- **Computer Science is economically important**, because digital systems have become a critical component of our entire society. It is also a terrific path to a good job; the 2011 IDC Microsoft Economic Impact study found over 110,000 IT vacancies in the UK, and expects the IT workforce to grow by a further 113,000 by 2015.
- **Innovative ICT teachers are in the vanguard of the movement for change**. Often they feel themselves to be under-qualified, but that is a problem that can be fixed: many are taking CPD courses in their own time to develop their knowledge.
- Offering Computer Science **directly addresses the problem that ICT has gradually become a low-status, low-achievement subject**. To quote the DfE, evidence indicates that recent curriculum and qualifications reforms have not led to significant improvements in ICT education in schools¹; that the number of students progressing to further study in ICT-related subjects is in decline; and that the ICT curriculum in its current form, particularly beyond KS3 is viewed as dull, repetitive and de-motivating for pupils².

Appendix 1 and 2 articulate in more detail the case for Computer Science as a discipline (Appendix 1) and one that every child should meet at school (Appendix 2). The Royal Society’s report³ on “[The way forward for Computing in schools](#)” gives authoritative backing.

¹ The importance of ICT: information and communication technology in primary and secondary schools, 2005/2008: Ofsted (March 2009); ICT in schools 2008-11: Ofsted (Dec 2011)

² E-skills UK: ‘Raising the bar for ICT: securing KS4 curriculum and relevant ICT pathways’ (2009); National Curriculum Review call for evidence: BCS The Chartered Institute for IT in association with the Computing At School Group (April 2011);

³ <http://royalsociety.org/education/policy/computing-in-schools/report/>

Matters have been brought to a head at national level by several recent developments.

First, several recent major reports have highlighted the issues:

- The [NextGen Livingstone-Hope Report](#)⁴ (2011), was absolutely clear: Recommendation 1 is to establish Computer Science as part of the National Curriculum.
- The Royal Society report [Shut down or restart: the way forward for Computing in UK schools](#)⁵ (Jan 2012) recommended that *“every child should have the opportunity to learn Computing at school, including exposure to Computer Science as a rigorous academic discipline”*.
- The [Ofsted Report on ICT](#)⁶ (Dec 2011) was very critical of the state of ICT education, especially at secondary schools, citing especially lack of challenge, and lack of teacher qualifications.

Second, the Government has responded with some major initiatives.

- The Department for Education is conducting a **Review of the National Curriculum**. This seems likely to remove the current statutory status for ICT in September 2014.
- More concretely, in his [speech at BETT](#)⁷ (Jan 2012), Michael Gove
 - Explicitly recognised Computer Science as a rigorous discipline that should be taught at school (see the quote on the front page of this document)
 - Announced that from September 2012 schools would no longer be required to follow the National Curriculum Programme of Study for ICT.
 - Specifically commended the CAS Curriculum for Computing (mentioned below) as a possible replacement.

Third, there are significant changes to qualifications at KS4:

- The government responded to the [Wolf Report](#)⁸ on vocational education by pledging to remove the “perverse incentives, created by the performance and funding systems, encouraging the teaching of qualifications which attract the most performance points...” The GCSE equivalency of some of the least demanding ICT qualifications may well be removed.
- From the late 1980s until 2010 there was no **GCSE in Computing** (astonishingly), but [OCR launched one](#)⁹ in 2010. In early 2012 both Edexcel and AQA announced new GCSEs in Computer Science to be available from September 2012. We believe that WJEC is also considering following suit.

Fourth, the [Computing at School Working Group](#)¹⁰ (CAS) has grown into a national organisation. CAS is a grass-roots group of concerned individuals, dedicated to establishing Computer Science as a proper school subject. Born in 2009, CAS has grown from zero to nearly 1000 members in three years, and is ramping up fast. School leaders can be assured that there is a vigorous, active support

⁴ http://www.nesta.org.uk/home1/assets/features/next_gen

⁵ <http://royalsociety.org/education/policy/computing-in-schools/report/>

⁶ ICT in Schools, Ofsted Dec 2011. <http://www.ofsted.gov.uk/filedownloading/?file=documents/surveys-and-good-practice/i/ICT%20in%20schools%202008-2011.pdf&refer=1>

⁷ <http://www.education.gov.uk/a00201868/michael-gove-speech-at-the-bett-show-2012>

⁸ <https://www.education.gov.uk/publications/standard/publicationDetail/Page1/DFE-00031-2011>

⁹ http://www.ocr.org.uk/qualifications/type/gcse_2010/ict_tec/computing/

¹⁰ <http://www.comptingschool.org.uk>

network for computing teachers, running an annual conference, CPD courses, local hub meetings, newsletters, and so on.

CAS has also produced a model [Curriculum for Computing](#)¹¹, focused on Key Stages 3 and 4. Developed by teachers and university academics in partnership, this curriculum gives schools a clear model for what Computer Science consists of, while leaving ample freedom for how to teach it.

Lastly, driven in part by these developments, there has been massive coverage in the media of the issue. For example, in January 2012 the Guardian launched [campaign to promote the reform of the ICT curriculum](#)¹², which has much useful coverage linked to it, but it is the tip of an iceberg.

All of this activity is strongly backed by major high-tech employers, including Microsoft and Google, who are finding it increasingly difficult to recruit well-qualified staff. Despite the recession, demand is high for graduates with strong computer science skills. In December, one-third of all jobs listed on a UK nationwide recruitment site were in software development, or closely related areas.¹³

What can you do?

In view of the government's proposed changes schools will likely be reviewing their existing curriculum. Here are some simple suggestions towards the next step:

- (5 minutes) Encourage your ICT teacher to join CAS, to join the local CAS hub, and to go to the CAS Teachers Conference in June.
- (1 hour) Meet with your ICT subject leader to discuss the issue. You may find other teachers too are interested in teaching Computer Science.
- (30 minutes) Review your ICT curriculum at a Governors meeting to understand how well it is serving the needs of your students, using this paper as background to highlight potential issues
- (2 hours) Talk to other heads in your area. Much can be gained by sharing expertise (maybe one school has a particularly well-qualified ICT teacher), and resources.
- (10 hours) Run a study to see how Computer Science could fit into your school's curriculum. For example:
 - Could you include a distinctive Computer Science strand in your ICT provision at Key Stage 2 and 3?
 - Could you offer a GCSE and/or A level in Computing?
 - Assess what is immediately achievable and determine how to build on that success in the future.

Change is inevitable, and the direction of change is clear. Managing the transformation will present challenges for school leaders. Leading edge schools will want to start planning now. The rest of this paper explores the issues in more depth, focusing especially on questions that school leaders have asked. We hope it is useful.

¹¹ <http://www.computingatschool.org.uk/data/uploads/ComputingCurric.pdf>

¹² <http://www.guardian.co.uk/education/series/digital-literacy-campaign>

¹³ <http://jobsite.co.uk>

More information

The easiest way to get more information is to join the Computing at School Working Group: <http://www.computingschool.org.uk>. Alternatively, email the CAS coordinator, Simon Humphreys: simon.humphreys@computingschool.org.uk.

APPENDICES

Contents

1	What is Computer Science?.....	8
1.1	Computer Science is a broad and deep discipline	9
1.2	Computational thinking is transforming all aspects of our society	9
1.3	A curriculum for Computer Science	10
1.4	Computer Science and ICT	11
2	Should Computer Science be taught at school?.....	11
2.1	The educational case for Computer Science at school	12
2.2	The economic case for Computer Science at school	13
3	What qualifications are available in Computer Science?	14
3.1	Key stage 4	14
3.2	A Level	14
3.3	International Baccalaureate	15
4	Questions and answers	15
4.1	Will our school’s ICT teachers be able to teach Computer Science?	15
4.2	Will our students take Computer Science if it’s an option at Key Stage 4 or 5?	15
4.3	Is Computer Science the same as “coding” or “programming”?	16
4.4	Will our school need new hardware/software to offer Computer Science?	17
4.5	Would my teachers have to write entirely new schemes of work?	17
4.6	Does this mean the end of ICT?	17
4.7	Does GCSE in Computer Science replace GCSE in ICT?	18
4.8	What will happen to our school’s 5 A*-C ratings?	18
4.9	Why is Computer Science not in the English Baccalaureate?.....	18
4.10	Do universities value qualifications in Computer Science?	18
4.11	What is the experience of schools that have embraced Computer Science?	19
4.12	What opportunities are there for partnership?.....	19
4.13	What is happening in other countries?.....	19

1 What is Computer Science?

Computer Science¹⁴ is the study of how computers and computer systems work, and how they are constructed and programmed, and the foundations of information and computation, in both artificial and natural information processing systems. Computer science includes (among many other things)

- **Algorithms:** re-usable procedures (often a sequence of steps) for getting something done. For example, plan the shortest delivery route for a lorry, given the required stops on the route.
- **Data structures:** ways to organise data so that a program can operate quickly on it. For example, there are many different ways to represent numbers (twos-complement, floating point, arbitrary precision, etc) with different trade-offs. Another example: a lookup table might be organised as a sorted array or as a hash table, depending on the size of the table and key distribution.
- **Programs** tell a computer exactly what to do. Every program is written in some programming language, each with different strengths. Good languages embody many “abstraction mechanisms”, that allow a piece of code to be written once, and reused repeatedly without reference to its internal structure. These mechanisms are the key to controlling the enormous complexity of real programs (e.g. a web browser), which consists of dozens of layers of such abstractions.
- **Architecture** is the term used to describe the large scale structure of computer systems. At the bottom is real physical hardware. On top of that are layered virtual machines. Compilers translate from a high level programming language to the low-level binary that the hardware or virtual machine executes. Operating systems manage the resources of the machine. Hardware and software interfaces, including device drivers, are required for interactions with other things, e.g. if the system is controlling a chemical plant or interacting with humans.
- **Communication.** Almost all computer systems consist of a collection of sub-computers, each running one or more programs, and communicating with the others by sending messages or modifying shared memory. The internet itself is a large-scale example, and uses protocols (standardised procedures) that keep data flowing smoothly despite all the control being decentralised. Increasingly computers need to be given natural language competences for communicating with humans, along with abilities to understand pictures, drawings and gestures.

Alongside these concepts are a set of Computing ‘methods’ or ways of thinking, including:

- **Modelling:** representing chosen aspects of a real-world situation in a computer. This includes both modelling new engineering designs in order to test them and modelling natural information processing systems in the course of understanding and predicting their behaviours.
- **Decomposing** problems into sub-problems, and decomposing data into its components

¹⁴ We use the term “Computer Science” because that is the term used by universities, and because no one misunderstands what it means. School teachers often use the term “Computing” instead, and we will occasionally do so to where it is less clumsy.

- **Generalising** particular cases of algorithm or data into a more general-purpose, re-useable version. This is often followed by discovering new applications for such generalisations.
- **Designing, writing, testing, explaining, and debugging** programs.

Computer Science provides insights into a broad range of systems — not only ones that include computers — and computational thinking influences fields such as biology, chemistry, linguistics, psychology, economics and statistics. Computing allows pupils to solve problems, design systems, and understand the power and limits of human and machine intelligence. Pupils who can think computationally are better able to conceptualise and understand computer-based technology, and so are better equipped to function in modern society.

Computer Science is also a practical subject, where innovation, creativity, and resourcefulness are encouraged. Pupils must apply underlying principles in order to understand real-world systems, and to create purposeful artefacts for themselves. This combination of principles, practice, and invention makes it an intensely creative subject, suffused with excitement, both visceral (“it works!”) and intellectual (“that is so beautiful”).

1.1 **Computer Science is a broad and deep discipline**

Education enhances pupils’ lives as well as their life skills. It prepares young people for a world that does not yet exist, involving technologies that have not yet been invented, and that will present technical and ethical challenges of which we are not yet aware.

To do this, education aspires primarily to teach *disciplines* with long-term value, rather than *skills* with immediate application. A “discipline” is characterised by:

- **A body of knowledge**, including widely applicable ideas and concepts, and a theoretical framework into which these ideas and concepts fit.
- **A set of rigorous techniques and methods** that may be applied in the solution of problems, and in the advancement of knowledge.
- **A way of thinking and working** that provides a perspective on the world that is distinct from other disciplines.
- **Longevity**: a discipline does not “date” quickly. Although the subject advances, the underlying concepts and processes remain relevant and enlightening.
- **Independence from specific technologies**, especially those that have a short shelf-life.

Computer Science is a discipline with all of these characteristics. It encompasses foundational principles (such as the theory of computation) and widely applicable ideas and concepts (such as the use of relational models to capture structure in data). It incorporates techniques and methods for solving problems and advancing knowledge (such as abstraction and logical reasoning), and a distinct way of thinking (computational thinking) and working that sets it apart from other disciplines. It has longevity (most of the ideas and concepts that were current 50 or more years ago are still applicable today), and every core principle can be taught or illustrated without relying on the use of a specific technology.

Its aspects of design, theory, and experimentation are drawn from, and contribute to, Engineering, Mathematics and Science respectively: Computer Science is a quintessential STEM discipline.

1.2 **Computational thinking is transforming all aspects of our society**

Many subjects teach problem-solving skills to some degree, but Computer Science develops a particularly systematic and deep approach to thinking about complex problems, often called

“computational thinking¹⁵”. Computational thinking is the process of *recognising* aspects of computation in the world that surrounds us, and *applying* tools and techniques from Computer Science to understand and reason about both natural and artificial systems and processes. Pupils learn to think about the same problem at many levels of abstraction, and to recognise that a single solution may apply to many other apparently different problems.

More profoundly, thinking about other disciplines through a computational lens has radically changed the way those subjects are studied, whether physics or biology, psychology or economics. For example, viewing biological processes as computational systems that process information has led to fundamental new insights in understanding disease that would not have been obtained through traditional thinking. Computational thinking has, indeed, led to whole new disciplines such as bioinformatics, and all scientists now need a core understanding of this kind of thinking.

Computational thinking is made concrete in programming. Programming takes computational thinking skills and empowers pupils to take charge of computers and create new software of their own, rather than simply to consume things made by others. This ability unleashes enormous creativity, drives innovation, and opens up completely new horizons and possibilities. (To get some sense of this diversity, look at the [Greenfoot Gallery](#)¹⁶ or the [Scratch Project gallery](#)¹⁷, which contain numerous examples of programs created by students from KS3 upwards.) To take an analogy from mathematics, every child should understand algebra, be capable of abstracting appropriate problems into algebraic expressions, and be able to solve simple algebraic equations. In the same way, in the computational sphere every child should be able to construct elementary algorithms in programmatic form that encapsulate simple ideas and concepts. Programming is a way of expressing creativity, of communicating and sharing ideas, just as mathematics does in a different area of discourse.

1.3 A curriculum for Computer Science

The Computing at School Working Group has developed a curriculum for Computing at school level, the first time such a thing has existed in the UK. Using the structure of the National Curriculum Programmes of Study, it articulates in more detail:

- **The importance of the discipline**
- **Key themes** (languages, machines, and computation; data and representation; communication and coordination; abstraction and design; and the wider context of computers).
- **Key processes** and computational thinking (abstraction, modelling, decomposing, generalising, and programming).
- **Range and content** of the subject (algorithms, programs, data, computers, communication and the internet, and advanced topics)
- **Level descriptors**

This model curriculum can be found at

<http://www.computingatschool.org.uk/data/uploads/ComputingCurric.pdf>

¹⁵ Jeanette Wing, “Computational thinking”, Communications of the ACM, March 2006, <http://www.cs.cmu.edu/afs/cs/usr/wing/www/publications/Wing06.pdf>

¹⁶ <http://greenfootgallery.org>

¹⁷ <http://scratch.mit.edu>

1.4 Computer Science and ICT

Computer Science and ICT are complementary subjects. Computer Science teaches a student how to be an effective *author* of computational tools (i.e. software), while ICT teaches how to be a thoughtful *user* of those tools. This neat juxtaposition is only part of the truth, because it focuses too narrowly on computers as a technology, and computing is much broader than that. More specifically:

- **Computer Science** is a discipline that seeks to understand and explore the world around us, both natural and artificial, in computational terms. Computer Science is particularly, but by no means exclusively, concerned with the study, design, and implementation of computer systems, and the principles underlying these designs.
- **ICT** deals with the purposeful application of computer systems to solve real-world problems, including issues such as the identification of business needs, the specification and installation of hardware and software, and the evaluation of usability. However, within current school curricula, the subset of ICT that focuses on simply using computers for a set task is emphasised.

We want our children to understand and play an active role in the digital world that surrounds them, not to be passive consumers of an opaque and mysterious technology. A sound understanding of computing concepts will help them see how to get the best from the systems they use, and how to solve problems when things go wrong. Moreover, citizens able to think in computational terms would be able to understand and rationally argue about issues involving computation, such as software patents, identity theft, genetic engineering, digital rights management, electronic voting systems for elections, and so on. Some of them can go on to make major contributions in fields attempting to understand and control some of the most complex processes on the planet, including processes in animal and human brains, and other biological processes, such as immune systems and embryo development. In a world suffused by computation, every school-leaver should have an understanding of computing.

2 Should Computer Science be taught at school?

Even if Computer Science is a discipline, it does not necessarily follow that it should be a school subject. But in fact there is an extremely strong case for doing so, with two strands: educational and economic.

The Chairman of Google, Eric Schmidt, said the un-sayable in his November 2011 address at the Edinburgh Television Festival. He paid tribute to Britain's record of innovation, saying the UK had "*invented computers in both concept and practice*" before highlighting that the world's first office computer "*was built in 1951 by the Lyons chain of teashops*". But he went on to say that the country that invented the computer was "*throwing away your great computer heritage*" by failing to teach programming in schools.

"I was flabbergasted to learn that today computer science isn't even taught as standard in UK schools," he said. *"Your IT curriculum focuses on teaching how to use software, but gives no insight into how it's made."*

"If the UK's creative businesses want to thrive in the digital future, you need people who understand all facets of it integrated from the very beginning. Take a lead from the Victorians and ignore Lord Sugar: bring engineers into your company at all levels, including the top."

Moreover, if the [President of the Royal Society](#)¹⁸ is correct in claiming that a major new idea in Biology concerns the central role that information management plays in generating biological organisation, then there is surely a need for many school-leavers *other* than those going into computer science to have a pre-university education with a strong computing component.

2.1 The educational case for Computer Science at school

Every student should encounter elementary Computer Science concepts, from Key Stage 1 onwards, *whether or not they intend to specialise in Computing*. Computer Science is important because:

- Computer Science develops a unique way of thinking about issues, problems and situations, that uses the powers of logic, algorithm, precision and abstraction.
- Computer science is intensely creative; in a real sense we build computer systems from pure “thought-stuff”. It empowers students to bring new things into being, and to move from being consumers of technology to producers and shapers of technology.
- Computer Science lets us understand the natural world in a new way, and is rapidly invading other disciplines, not merely as a way to do calculations, but as a whole new way of thinking. For example, Systems Biologists regard cells as machines controlled by DNA, and are busy programming them, while computational models are illuminating studies of biodiversity and animal populations.
- Computer Science equips students to understand and contribute to understand and argue rationally about societal issues involving computation, such as software patents, identity theft, genetic engineering, electronic voting systems for elections, complex modern financial trading systems, and so on
- Many of the scientific disciplines that are increasingly depending on computational concepts and theories currently have to take in new university students who know little or nothing about computation, even if they are proficient users of some of the technology. This can seriously hamper education, research, and development in those fields. Making computer science available at school could have a major impact on our future advanced teaching and research in those disciplines.

There is a strong analogy with the other sciences. We take it for granted that every student should learn the elementary concepts of (say) Physics at primary school and Key Stage 3, after which some will choose to study further at GCSE. Some will choose to take an A-level, and some of them will go on to study Physics at University. Moreover, there is a range of choices at each level, varying in the level of intellectual demand and vocational emphasis.

Exactly the same pattern should apply to Computer Science. Indeed, in a world in which digital technology and computational concepts play an increasingly crucial role, Computer Science is really the “fourth science”, and Michael Gove suggested (in his BETT 2012 speech) that it might ultimately count as such in the English Baccalaureate.

We want young people to understand and play an active role in the digital world that surrounds them, not to be passive consumers of an opaque and mysterious technology. We want them to be able to take part in well informed discussion of questions like “Am I some sort of computer?” or “Could future robots think like humans?” A sound understanding of Computer Science concepts will equip them to get the best from the systems they use, to recognize poorly designed systems

¹⁸ <http://www.bath.ac.uk/news/2011/03/21/founders-day-lecture/>

rather than blame themselves, and to solve problems when things go wrong. In a world suffused by computation, every school-leaver who lacks a basic appreciation of Computer Science is at a distinct disadvantage.

2.2 The economic case for Computer Science at school

"We don't want the next Intel or Google to be created in China or India. We want those companies and jobs to take root in America" Barack Obama President of the United States of America, May 2011.

Every parent wants their child to get a good job after their education, and there is no surer way to achieve that goal than to study Computer Science.

- The European Commission predicts UK will need an additional 500,000 IT professionals by 2015¹⁹.
- London high-tech start-ups already struggle to recruit high quality graduates as the City currently soaks them up.²⁰ Banks in the City now also recruit from Eastern Europe for IT professionals due to UK shortage of high quality Computer Science graduates.
- The same shortage is now affecting the Computer Games industry²¹, where the UK has now sunk from third in the game development league table to sixth²². This is significant as the Games industry has always been able to take its pick of talented software developers and has never suffered a shortage before.
- 27% of UK jobs are already IT-related - and the demand for IT specialists is predicted to grow at four times the rate of the overall UK workforce²³
- According to a [2011 study for the Royal Academy of Engineering](#)²⁴, those working in technology occupations earn 33% more than those who do not work in science, technology, or engineering.

Many have the mistaken belief that all IT jobs in the UK are being outsourced to China and India. This is false. While approximately 3% of low level IT jobs are outsourced each year, around the same number of higher level, more technically demanding IT jobs are created each year.

Even in a recession, there is a continually increasing demand for IT and tele-communications professionals. According to the 2011 IDC Microsoft Economic Impact study, the UK has an IT workforce of 1.45m, which is expected to grow by 113,000 by 2015. Moreover there are over 110,000 IT vacancies.

This shortage is apparent even in the computer games industry, which has been a flagship sector of our economy. According to NESTA²⁵, *"in just two years, it seems the UK's video games industry has dipped from third to sixth place in the global development rankings"*. This is an especially alarming statistic given that the UK Games industry has always been able to take the pick of the crop. The shortage of developers is, according to the NESTA report *"reinforced by a school curriculum that*

¹⁹ Evaluation of the implementation of the communication of the European Commission E-Skills for the 21st Century, Tobias Hüsing Werner B. Korte, European Commission Oct 2010

²⁰ <http://eu.techcrunch.com/2011/05/30/silicon-milkroundabout-how-london-startups-took-hiring-back-into-their-own-hands/>

²¹ www.nesta.org.uk/library/documents/NextGenV32.pdf

²² <http://www.telegraph.co.uk/technology/video-games/7045378/Dull-teaching-damaging-video-games-industry.html>

²³ New Industry New Jobs, Skills System Case Studies, BIS report, March 2010, URN 10/930

²⁴ [http://www.raeng.org.uk/news/releases/pdf/](http://www.raeng.org.uk/news/releases/pdf/The_Labour_Market_Value_of_STEM_Qualifications_and_Occupations.pdf)

[The_Labour_Market_Value_of_STEM_Qualifications_and_Occupations.pdf](http://www.raeng.org.uk/news/releases/pdf/The_Labour_Market_Value_of_STEM_Qualifications_and_Occupations.pdf)

²⁵ NESTA Review by Ian Livingstone and Alex Hope, Next Gen. Transforming the UK into the world's leading talent hub for the video games and visual effects industries, 2011

focuses in ICT on office skills rather than the more rigorous computer science and programming skills which high-tech industries like video games and visual effects need”.

It is also important to remember that there is a close link between software engineering skills and modern business entrepreneurship and innovation. In California’s Silicon Valley, the two most successful recent start-ups – Facebook and Google – were catalysed by software engineers. Closer to home, the same is true of Silicon Roundabout and Silicon Fen.

The trend in undergraduate recruitment to Computer Science courses reflects the shortage of able young people entering the IT profession. According to HESA data, in 2010 the number of graduates in Computer Science and Software Engineering was 11,095, whilst at its peak in 2002 the number was 22,630 — a drop of slightly more than 50%. Such a critical shortage of applicants is a danger sign that ICT in school does not provide a worthwhile professional progression route.

More data to substantiate these points can be found in the [BCS Computing Fact Sheet](#)²⁶

3 What qualifications are available in Computer Science?

3.1 Key stage 4

It is astonishing to record that at Key Stage 4 there had been no GCSE Computing qualification for two decades until 2010. In that year OCR launched a pilot [GCSE in Computing](#)²⁷, with advice from CAS. It was enthusiastically adopted, and it is now out of pilot.

In January 2012 Edexcel announced that they will introduce a GCSE in Computing Science in September 2012. In February 2012 both AQA and WJEC followed suit. As a result there are now **four** GCSEs in Computer Science, a dramatic change from two years ago.

The University of Cambridge International Examinations runs an [International GCSE in Computer Studies](#)²⁸. This course is taken by around 14,000 students worldwide, with a steady 15% year on year increase. In the UK, around ten private schools offer the IGCSE. The syllabus covers applications of computers, and their social and economic implications; the system life cycle; algorithm design, programming, and logic gates; generic software and the organisation of data; hardware, systems, and communications. One might debate some of the details (there could be a greater focus on principles) but it is clearly a computer science qualification.

In June 2010 the [DfE announced that IGCSEs would be available to UK state schools](#)²⁹. Although the Computer Studies IGCSE is not in the approved list, CAS is working to get this changed.

3.2 A Level

Three awarding bodies offer an A level in Computing:

- [AQA Computing](#)³⁰
- [WJEC Computing](#)³¹
- [OCR Computing](#)³²

In addition:

²⁶ http://www.computingatschool.org.uk/data/uploads/BCS_Computing_Fact_Sheet.pdf

²⁷ http://www.ocr.org.uk/qualifications/type/gcse_2010/ict_tec/computing/

²⁸ http://www.cie.org.uk/qualifications/academic/middlesec/igcse/subject?assdef_id=844

²⁹ http://www.cie.org.uk/qualifications/academic/middlesec/igcse/uk/state_schools/

³⁰ http://web.aqa.org.uk/qual/gce/ict/computing_noticeboard.php

³¹ <http://www.wjec.co.uk/index.php?subject=28&level=15>

³² http://www.ocr.org.uk/qualifications/type/gce/ict_tec/computing/

- **Scotland:** SQA offer Higher Computing
- **Northern Ireland:** CCEA are developing a “Software and Systems Development” A-level, for teaching from 2013

3.3 International Baccalaureate

The International Baccalaureate has a course in Computer Science. The current IB Computer Science course is largely a course in Java programming, and a completely new revision of the course is under way, due for launch in August 2012. It is much less tied to a specific programming language, and instead lays greater emphasis on the principles underlying Computer Science, much along the lines of the [Computer Science Principles course](#)³³ under development in the USA.

In addition the IB has a course called Information Technology in a Global Society (ITGS) is organised around the use and application of computers. The [syllabus](#)³⁴ has a useful table (page 4) contrasting ITGS with Computer Science.

4 Questions and answers

4.1 Will our school’s ICT teachers be able to teach Computer Science?

Yes, they will. The lack of teacher qualifications is often cited as a reason that Computer Science cannot be treated as a school subject, but CAS has ample evidence to the contrary

- **School ICT teachers are in the vanguard.** Many ICT teachers are cordially fed up with their low status, and the low expectations of their subject. They are eager for change. Over 700 of them are members of CAS, simply by word of mouth.
- **Teachers can learn, and are keen to do so.** CAS and Vital, supported by universities, already run CPD courses, and are working with awarding bodies to run more.
- **Fantastic free teaching resources are available.** These include programming environments that can engage and inspire children starting from age 8 or younger (Scratch, Alice, Kodu, Robomind, Logo, Greenfoot, to name but a few). The cs4fn site is a treasure trove of puzzles and stories all with a computer science theme. The CS Unplugged book from New Zealand teaches computer science without even using computers. And so on. Together these resources can already amply support an entire KS3 scheme of work.
- **There is a vigorous support network.** CAS gives isolated ICT teachers the opportunity to work together, and share best practice.

That said, teaching Computer Science does need a subject specialist. ICT is (sadly) often taught by a teacher from another discipline with a spare period (“everyone knows how to use a spreadsheet, right?”), and that will not work for Computer Science any more than it would for Maths. Making time for enthusiastic teachers to learn Computer Science is the main resource implication for schools.

4.2 Will our students take Computer Science if it’s an option at Key Stage 4 or 5?

The experience from schools that have adopted Computing GCSE and/or ‘A’ Level is emphatically “yes”. Here is the content of an email from one UK teacher:

³³ <http://csprinciples.org/>

³⁴ http://www.ibo.org/diploma/assessment/subjectoutlines/documents/d_3_itgsx_gui-out_1001_1_e.pdf

The scene: myself and 20 students:

"OK folks, if we ran A-level Computing and ICT next year, who would do A-level Computing if given the chance?". 12 hands go up, plus a few "thinking about it."

"Who would do ICT?" 1 hand.

"The bad news - after a very lengthy debate, Computing is unlikely to run. Who wants to do ICT?". 8 hands. Some complaints of "Being shafted" in relation to their university choices, and "How stupid is that?" I have a couple of guys with tears coming down their cheeks.

This is not good. I know how they feel.

Computer Science at Key Stage 4 or 5 is not for all students but those who enjoy the subject, particularly if it is introduced in earlier stages, often develop a passion for it and find it hugely rewarding. This is a combination of the intellectual challenge and reward which the inherent problem solving of Computer Science brings and also the result of creating real solutions in the form of programs.

Computer Science has an unhelpfully geeky image, but that is changing:

- Computer science is intensely creative, most obviously in programming where students design, create, test, and share ideas that have never before existed.
- Digital systems are so pervasive in our society, and computing principles are becoming so ubiquitous in other disciplines, that every student should have the opportunity to learn the fundamentals of computer science – the very opposite of a niche subject.
- Computer Science certainly does have hard academic content; but so do Maths and Biology.
- Students and parents also recognise that computing qualifications are valuable in the long term job market which has a strong influence on student's option choices.
- Girls are massively under-represented both in university computing courses and in the IT profession. The causes are complex, but it is clear that the pattern is firmly established by the end of school. This is an area where schools can change the profession.

4.3 Is Computer Science the same as "coding" or "programming"?

Much of the media coverage about the lack of deep content in our current ICT curriculum has focused on introducing **programming** or **coding** instead. (You may have seen the strapline "Coding is the new Latin".)

Programming is absolutely central to Computer Science. Programming animates the discipline, gives ideas concrete form, and makes them dance. Programming is fabulously motivating as a pedagogical device; as well as *designing* and *building* things, students can also *run* and *share* them. Several organisations, such as Coding for Kids, Apps For Good, and Young Rewired State, are organised mainly around programming as a motivator. After-school programming clubs are a popular and effective way of engaging enthusiasts.

So no one could call themselves educated in the subject without the ability to program; and yet *programming is not the same as Computer Science*. Dijkstra famously remarked that "Computer science is no more about computers than astronomy is about telescopes", and that neatly encapsulates the point.

Computer Science fundamentally consists of concepts, ideas, principles, and techniques, such as algorithms, data structures, and abstraction. (The CAS Curriculum gives specifics.) That is the goal. Coding is an essential means towards that goal, and a very employable asset as well. But we must

guard against “We have a course in Java so of course we are teaching Computer Science”. Concepts endure for a lifetime, whilst particular skills (such as a particular programming language) will date.

4.4 Will our school need new hardware/software to offer Computer Science?

No. Most schools have perfectly adequate IT suites. All of the software needed to teach Computer Science is either open-source and hence completely free, or else can be freely licensed for educational use (including by students at home) from manufacturers like Microsoft.

Moreover, students increasingly have mobile technology of their own. Today’s mobile phones are more powerful than yesterday’s computers. Complementing this, new and extremely cheap hardware is becoming available specifically for educational uses, such as [Raspberry Pi](#) (£20 for a complete credit-card-sized computer) and [Gadgeteer](#).

To help teachers expand their understanding of computation and to develop shared resources and tools it may be useful to provide new 'cloud'-based computing facilities. This may require overcoming some resistance to allowing interactive internet access from some school networks.

4.5 Would my teachers have to write entirely new schemes of work?

One of CAS’s major roles is to help teachers of Computer Science get together to share best practice and classroom teaching resources. For example:

- There are active sub-groups of teachers delivering Computing at GCSE and A level, which develop and share resources.
- In collaboration with the National STEM Centre, CAS is developing a resource repository to enable such resources to be shared.
- By way of a “taster” there is already a [six-lesson Key Stage 3 scheme of work](#) based on programming.

4.6 Does this mean the end of ICT?

One of the difficulties (highlighted by the Royal Society) with the current status quo is that the term “ICT” is used to cover a huge range of material linked to computers and digital technology, from basic literacy to hard-core computer science, including at least:

- **The (current) National Curriculum subject** called ICT
- **Digital literacy:** the basic ability to use word processors, the internet, spreadsheets, and so on, safely and effectively.
- **ICT across the curriculum** where the content of the ICT National Curriculum is taught by class teachers or subject teachers within specific areas/departments of the school, under the direction of the ICT coordinator.
- **Using generic information technologies to support teaching and learning** (interactive whiteboards, VLEs, class response systems, etc.).
- **Using specific computer technologies** to support specific aspects of a subject (for example, weather stations in geography, MIDI instruments in music, etc.)
- **Using technologies to support teachers' administrative processes** including registration, record keeping, reporting, communicating, etc.

These multiple meanings lead to misunderstandings, when a speaker (eg a head teacher) intends one thing and a listener (eg an ICT teacher) understands another.

Many of these things remain important:

- **Digital Literacy** is a key skill and, despite the talk of “digital natives”, remains one that needs to be taught; but it is a skill not a subject, and should be treated in much the same way that we treat numeracy, reading, and writing.
- **Information technology is a tremendous enabler of learning in other subjects**, and should be used extensively. But the point there is to use technology to enhance the learning of (say) history, not to use history as a vehicle for teaching ICT skills.
- **The school’s IT infrastructure** is mission critical. But again it is the enabling technology for teaching and learning, like the school heating system.
- **Beyond KS3 schools obviously need a range of pathways and qualifications** for different pupils. Schools are best placed to judge what is appropriate for their cohorts.

4.7 Does GCSE in Computer Science replace GCSE in ICT?

No. A GCSE in Computer Science is different to the GCSE in ICT; it is not a replacement for it. Students with a more applied focus may well want to take the GCSE in ICT, while those more interested in understanding how it all works may take Computer Science.

4.8 What will happen to our school’s 5 A*-C ratings?

Computer Science is a real, high-status, challenging subject, like Maths, Physics, Music, or History. As such it does not offer an easy ride to 100% pass rates, as ICT qualifications have done in the recent past.

However, GCSE equivalency for some ICT qualifications is likely to disappear, or at least be reduced. National benchmarking will continue to change and there is now widespread recognition that mass take-up of some vocational ICT courses had little to do with the development and prospects of the students involved

4.9 Why is Computer Science not in the English Baccalaureate?

Computer Science should be an EBac subject, and CAS is arguing that case vigorously with the DfE. The DfE’s basic position is that the EBac is not a performance benchmark, and schools are free to do as they please. However, in a key part of Michael Gove’s [BETT 2012 speech](#), he said

“So I am also announcing today that, if new Computer Science GCSEs are developed that meet high standards of intellectual depth and practical value, we will certainly consider including Computer Science as an option in the English Baccalaureate.”

Meanwhile, given the clear benefits of learning the subject, its current absence from the EBac does not appear to be sufficient reason to deny pupils this opportunity.

4.10 Do universities value qualifications in Computer Science?

Fundamentally, universities want schools to teach Computer Science. However this message has been muddled by the historical context.

- Universities have no interest whatsoever in ICT qualifications as preparation for further study in Computer Science. This negative message has tainted the perception of university attitudes to a Computing A level.
- University Computer Science departments simply cannot *require* a Computing A level, because so few students currently take it. Since many CS undergraduates will not have A level Computing, a large chunk of a typical first-year course overlaps with the A level. As a

result, some admissions tutors say “if you want to study Computer Science, and you are looking for an A level to drop, then drop Computing”.

- A Computer Science A level would be an excellent foundation for a degree in many subjects, including Physics, Maths, Engineering, Biology, Economics, and so on.
- Until recently there was no GCSE in Computing, so universities admissions will say nothing about it, yet.

As a result of these mutually-reinforcing factors, the message that is received by school teachers and careers advisers can be a negative one: universities don't value A level Computing.

But this is not the message that the universities intend to send. As noted above, they are grappling with a chicken and egg problem of their own. CAS is working with individual universities, and with the Russell Group (whose booklet *Informed Choices* is very influential), so help make this particular communication channel more accurate. For example, Imperial College's admissions page now explicitly welcomes a GCSE in Computing.

We also need to think about the importance of a pre-university education in computing for many other degree courses *apart from computer science*. The importance for biology and biological disciplines such as neuroscience and psychology has already been emphasised. But increasingly computational competence is important for other disciplines including mathematics, physics and engineering. Even philosophy is now addressing computational issues – as illustrated by the new degree in computing and philosophy at Oxford University, and others that have existed for some time.

4.11 **What is the experience of schools that have embraced Computer Science?**

Hundreds of teachers on the CAS mailing list can help to answer this question, but the [Royal Society Report](#) has some useful case studies, in Chapter 4.

4.12 **What opportunities are there for partnership?**

Many IT professionals are deeply concerned about the state of computing education, and are eager to help. Many are members of CAS and/or are STEM ambassadors. There are opportunities here for software developers and game designers to become involved with their local school, if the latter is willing. These relationships might ultimately lead to strategic partnership with local high-tech companies.

4.13 **What is happening in other countries?**

Many other countries, both developed and developing nations, are experiencing much the same epiphany that we have described here. The CAS paper “Computing at school: international comparisons” give a compact summary.