Rationale for a GCSE in Computing

Computing for the Next Generation group

V2.1, August 2008

Abstract: This document argues the case for creating a GCSE in Computing.

1. Summary

Why should there be a new GCSE in Computing? We summarise the arguments as follows:

- Computing is complementary to, but quite different from ICT (Section 2). It appeals to different students, and addresses different needs.
- A GCSE in Computing would serve needs that cannot be met by the existing GCSE in ICT (Section 3).
- There is a looming skills shortage in Computing (Section 4). University applications have fallen by 60% since 2002, but the demand from employers has risen, and continues to rise.
- Computing is a discipline (Section 5), like mathematics or physics, which illuminates and cross-fertilises other subjects.
- The existing GCE level in Computing lacks a "lead in", and when students are making GCE level (i.e. A level) choices they have no experience of Computing. A GCSE in Computing would increase the supply of students taking the GCE.
- Schools are now well equipped with computers, and many students have access to a computer outside school. Centres that already offer the GCE in Computing will have little problem offering a GCSE, and it may provide a development and career path for some ICT teachers and students.

2. Computing and ICT

Computing is complementary to, but quite different from Information and Communication Technology (ICT). ICT is about the **use** of computers and their **applications**. Computing is about their **design** and **implementation**. To use the analogy of a car,

- **ICT** is the equivalent of teaching how to drive a car, and how to navigate it. Once basic skills have been learned (how to use the clutch), the emphasis is on appropriate choice of destination, how to drive safely, how to develop a good route to the destination, how to choose which car is the right vehicle.

  Everyone should be able to drive, and similarly every student should possess basic ICT skills, and some knowledge of how to use them.
- **Computing** is the equivalent of teaching automotive engineering: how the clutch works, how to design new cars, and how to maintain existing ones.

Computing is more than just programming, which in the car analogy would be the equivalent of metalwork.

Not everyone needs to know how to design or maintain a car. Similarly, only a subset of (able) students will want to study Computing, just as only a subset want to work in the automotive or related industry, even though both vehicle and computer based industries are major employers and revenue generators for the UK economy.

3. The need for a GCSE in Computing

There is abundant evidence that, while the current GCSE in ICT serves some needs well, it is actually counter-productive for able students interested in computer technology. The 2008 "IT & Telecoms Insight Report" published by Eskills UK says *"The image of IT-related degrees and careers was that they would be repetitive, boring, and more-of-the-same; for example use of IT office applications such as word processing, spreadsheets, and databases"*. The next bullet says *"The ICT GCSE had a major part to play in creating their (negative) impressions"*.

The 2007 report "Developing the future", sponsored by Microsoft, City University, the British Computer Society, and Intellect says *"With no GCSE in Computing or Computer Science (only the GCSE in ICT which is not about the subject of computing) learning to use a computer and learning Computer Science become indistinguishable as far as students are concerned. The skew in emphasis has a direct bearing on a student's view of the IT industry; one that results in many negative perceptions".*

While there are doubtless improvements that could be made to the ICT GCSE, it is a fine qualification in principle. The problem is just that it has a different purpose than the one advocated here: a hammer is not a saw, and treating it as one leads to disappointment. For able students who are interested in how computers work, the ICT GCSE is simply inappropriate, for a number of reasons:

1. Because ICT is rightly regarded as a generic skill that should be possessed by *all* students, regardless of ability or motivation, the curriculum is necessarily basic.
2. Because ICT is a practical subject, qualifications in ICT carry a particularly heavy coursework load, which able students find de-motivating.
3. Students have increasing access to computing technology both at home and at school and are developing their IT skills at a much younger age, and often at home. The ICT curriculum has, for some, become a repetition of material already learned.
4. As remarked in Section 2, ICT addresses quite different questions from Computing. Students who are interested in the design and implementation of computer systems are unlikely to find ICT rewarding.

By the time students are making GCE level (i.e. A level) choices, they are already profoundly de-motivated about computing. It is too late to change their minds!

A GCSE in Computing would form a natural foundation for the GCE level, and would offer a much earlier opportunity to challenge and motivate students interested in computational thinking.

For those students not progressing to Computing GCE level, the GCSE gives a set of transferable skills, and an appreciation of the technology central to so much of today.

4. Challenge: the looming skills shortage

Computing has an immense impact on modern life. The job prospects are excellent and the field is rigorous, intellectually vibrant, and multi-faceted. Yet, computing is in danger of disappearing from schools, with a critical skills shortage developing.

A recent study by the UK Council of Professors and Heads of Computing illustrates the problem: it predicts that demand for IT professionals will increase by up to 15% in the next eight years, while the number of students aiming for jobs in the industry has fallen by 50% since 2001.

They further identified:

- The number of pupils studying Computing at GCE Level has fallen every year since an all time high in 2003
    - GCE computing students fell by 48.6% from 2001 to 2007 (10,913 down to 5,610)
    - GCE computing students fell by 33.9% from 2004 to 2007 (8,488 down to 5,610)
- The IT labour market is set to grow by 163,000 from 2007 to 2016 (from 1,069,000 up to 1,232,000).
- 179,800 appointments are made each year in the IT labour market, the majority (78.5%; 141,300) of which will go to "new entrants" (people who are not currently in the IT labour market).
    - Of this annual requirement of 141,300, 26,800 will be joining direct from education
- In 2005, an IT "Skill-Shortage Vacancy" was experienced by 5% of all employers. This equates to 28% of all employers with a vacancy
    - 38% of IT Managers have a technical skill gap, as do 12% of Networking Staff, 10% of Programmers, and 10% of PC Support Staff.
    - It is crucial to understand that technical staff and managers are the two areas where the largest employment growth will take place over the coming years.
    - A technical skill gap amongst such people is a serious problem.
- UK university applications to read Computer Science are down over 60% since 2000.

5. Computing is a discipline

A fundamental understanding of computing enables students to be not just educated users of technology, but the innovators capable of designing new computers and programs to improve the quality of life for everyone. It is not an exaggeration to say that our lives depend upon computer systems and the people who maintain them to keep us safe on the road and in air.

Without an understanding of computing we are but users, dependent on others, and the country will become a second class society in thrall to those who can develop the new technology. Without understanding of computing principles, computer and software projects go wrong. Without understanding of computing we are likely to misuse or use inappropriate technology with dangers for the fabric of society and civil liberty.

5.1 Computing is Important Intellectually

The invention of the computer in the 20th century is a "once in a millennium" event, comparable in importance to the development of writing or the printing press. Computers are fundamentally different from other technological inventions in the past in that they directly augment human thought, rather than, say, the functions of our muscles or our senses. Computers have already had enormous impact on the way we live, think, and act. It is hard to overestimate their importance in the future.

So why is it important to study computing? We live in a digitized, computerized, programmable world, and to make sense of it, we need computing. An engineer using a computer to design a bridge must understand the limitations of the numerical methods used, how the maximum capacity estimates were computed and how reliable they are. An educated citizen using a government database or bidding in an eBay auction should have a basic understanding of the underlying algorithms of such conveniences, as well as the security and privacy issues that arise when information is transmitted and stored digitally. These are computing, not ICT issues.

Computing students learn logical reasoning, algorithmic thinking, design and structured problem solving—all concepts and skills that are valuable well beyond the computing classroom. Students gain awareness of the resources required to implement and deploy a solution and how to deal with real-world and business constraints. These skills are applicable in many contexts, from science and engineering to the humanities and business, and have already led to deeper understanding in many areas. Computer simulations are essential to the discovery and understanding of the fundamental rules that govern a wide variety of systems from how ants gather food to how stock markets behave. Computing is also one of the leading disciplines helping us understand how the human mind works, one of the great intellectual questions of all time. There is much exciting work that lies ahead of us.

5.2 Computing Leads to Multiple Career Paths

The vast majority of careers in the 21st century will require an understanding of computing. Many jobs that today's students will have in 10 to 20 years haven't been invented yet. Professionals in every discipline—from art and entertainment, to communications and health care, to factory workers, small business owners, and retail store staff— need to understand computing to be globally competitive in their fields.

Movies like *The Incredibles* and *Lord of the Rings* required the development of new computing techniques. Progress on understanding the genetics of disease or of creating an AIDS vaccine requires professionals to think in terms of computing—because the problems are unsolvable without it. Those who understand the technology can make the new movies and invent the new techniques, and they are the professionals who will go beyond simply using what others have invented. Studying computing will prepare a student to become a professional software developer or to pursue a career in one of many related fields. Despite the depressing reports in the media, the reality is that professionals with computing training have never been more in demand in the UK and worldwide than they are today. Network managers need computing expertise to install new kinds of routers. Professional computer scientists rarely spend their days writing program code. More often they are working with experts in many fields, designing and building computer systems for every aspect of our society.

5.3 Computing Teaches Problem Solving

Artists, philosophers, designers, and scientists in all disciplines are united in the intensely creative activity of problem solving. Every painting by Picasso is an attempt to solve the problem of capturing an active, three-dimensional world on a flat canvas. Every TV commercial is an attempt to solve the problem of how to entice people to want, and then purchase, a product. And every well-designed scientific experiment provides data to support or refute a theory. Computing teaches students to think about the problem-solving process itself. In computing, the first step in solving a problem is always to state it clearly and unambiguously. Often a computer scientist works closely with business people, scientists, and other experts to understand the issues, and to define the problem so explicitly that it can be represented in a computer. This co-operative process requires people with different expertise and perspectives to work together to clarify the problems while considering each other's priorities and constraints. Computer programs must be designed, written, and tested. New hardware or devices may need to be made. Existing software systems and packages may be modified and integrated into the final system. Building a system is a creative process. The process requires computational thinking. With each fix of a bug or addition of a new feature, there's a hypothesis that the problem has been solved. Data is collected, results are analyzed, and if the hypothesis is untrue (alas, often!), the cycle repeats. A computer scientist is concerned with the robustness, the user-friendliness, the maintainability, and even the formal correctness of computer solutions to business, scientific, and engineering problems. These issues often require intense analysis and creativity. Computer specialists draw on their training and experience to avoid problems and to create the best possible solutions. Often this involves creating new programs and systems. That takes computing skill.

5.4 Computing Supports and Links to Other Subjects

Progress in science has always been linked with progress in technology and vice versa. For example, bacteria were first discovered not by a biologist but rather by a Dutch merchant who refined the art of making microscope lenses (and enjoyed peering at plaque he scraped off his unbrushed teeth). Nowadays, it's typical for computer scientists to work in other scientific disciplines. To solve the big scientific problems of the 21st century, such as grappling with new diseases and climate change, we will need people with diverse skills, abilities, and perspectives. And although it may seem surprising, computing can also help us learn what it really means to be human. The sequencing of the human genome in 2001 was a landmark achievement of molecular biology, which would not have been possible without computer scientists. After short DNA fragments of the genome were sequenced in biology labs, computers were used to figure out how to piece the fragments together. That required considerable new programming. This knowledge is paving the way for better computational methods of detecting and curing diseases, such as cancer, because we understand the genetic mutations involved.

It doesn't take a neuroscientist to appreciate the fact that the human brain is amazing. We know, for example, that an infant can effortlessly recognize a familiar face from many different viewpoints, and yet, we have a very poor understanding of the computational mechanisms that the brain uses to solve such tasks. Inferring meaning from images is a computational task, and computer scientists and neuroscientists are working together to figure out how to build computers that can process images and, ultimately, how we can better understand intelligence itself.

The use of modelling and simulation, visualization, and management of massive data sets has created a new field—computational science. This field integrates many aspects of computing such as the design of algorithms and graphics. In science classes, students use sophisticated simulation software to make molecules and geological processes come to life. Writing computer programs that model behaviour allows scientists to generate results and test theories that are impossible in the physical world. Advances in weather prediction, for example, are largely due to better computer modelling and simulation. Computational methods have also transformed fields such as statistics and mathematics. Scientists who can understand and contribute to technological innovation have a huge advantage. Good training for future scientists must therefore include a solid basis in computing.

Computing teaches transferable skills of problem solving, team working, logic and logical thinking, as well as linking to topics such as discrete mathematics.

5.5 Computing Can Engage All Students

Computing applies to virtually every aspect of life, so computing can be explicitly tied to a myriad of student interests. Students may be fascinated with specific technologies such as cell phones or have an innate passion for visual design, digital entertainment, or helping society. Computing teaching nurtures students' interests, passions, and sense of engagement with the world around them and offer opportunities for them to find purpose and meaning in their lives. Pedagogically, computer programming has the same relation to studying computing as playing an instrument does to studying music or painting does to studying art. In each case, even a small amount of hands-on experience adds immensely to life-long appreciation and understanding, even if the student does not continue programming, playing, or painting as an adult. Although becoming an expert programmer, a violinist, or an oil painter demands much time and talent, we still want to expose every student to the joys of being creative, for example by having students design and write programs that control their cell phones or robots, create physics and biology simulations, or compose music. Students will want to learn to use conditionals, loops, and parameters and other fundamental concepts just to make these exciting things happen. Examples include:

- **Computing and Digital Media**. Manipulating and creating digital media is a context that engages students and easily integrates with computing learning goals. Instead of iterating over an array to compute an average, students might write a program to iterate over an array of pixels to compute a negative image or a grey-scale image or try new forms of image manipulation. Students can learn that combining two arrays is the technique used to splice and mix digital sounds. Processing pictures and sounds in new ways needs new programs. Similar contexts are robotics and story-telling with digital media.
- **Team work to solve large uncertain problems**. Computer programs are some of the most complex structures built by mankind, and are rarely built in isolation. Software engineering has worked out paradigms for designing and building such structures by a co-operating team, often in the presence of real world uncertainty, for example by using agile engineering methodology. Generally, students much prefer to collaborate than to work alone, and computing can give them the disciplines and methods they need to work successfully on large uncertain projects, and in collaborative teams.
- **Computational Thinking**. How does one prevent a computer from creating many thousands of e-mail accounts that can be used to send spam to

millions of people? How can one design an electronic auction system that fairly represents the interests of all parties involved? How can one accurately simulate a system consisting of millions of objects evolving over billions of steps? How can one be certain a program will perform correctly, in life critical systems such as avionics or medicine? To deal with these problems, and many more similar ones, requires a type of thinking characteristic of computing: computational thinking. Computational thinking involves a clear focus on tangible problems; a large collection of proven techniques such as abstraction, decomposition, iteration, and recursion; an understanding of the capabilities of humans and machines alike; and a keen awareness of the cost of it all. Emphasis on computational thinking rather than just programming has greatly improved introductory courses and is starting to become a motivating principle in other parts of our curriculum.

- **Computers and the Visual Arts**. Computer technology and graphic arts have developed in parallel. Graphic representations can be used to teach fundamental computing concepts while new programs and technologies allow development of visual effects and new media opportunities.. An example is the development, reliant on new computer technology, of the web from static to dynamic images, and the shift 3-D representations and gesture interfaces. Computing skills are needed to maximise the opportunities these developments offer.
- **Computers and Biology.** Computing has become essential for solving biological and biotechnology problems. In molecular biology, for example the fragment assembly problem was a central computational task in sequencing the human genome. This problem is a nice vehicle for introducing the fact that some computational problems seem to have no efficient solution—a deep insight of computing.