BCS, The Chartered Institute for IT

in association with the Computing At School group

Consultation Response to:

New A levels: subject content consultation

Dated: 15 December 2013

BCS
The Chartered Institute for IT
First Floor, Block D
North Star House
North Star Avenue
Swindon SN2 1FA

This page is left deliberately blank.

BCS, The Chartered Institute for IT

The Institute promotes wider social and economic progress through the advancement of information technology science and practice. We bring together industry, academics, practitioners and government to share knowledge, promote new thinking, inform the design of new curricula, shape public policy and inform the public.

As the professional membership and accreditation body for IT, we serve over 70,000 members including practitioners, businesses, academics and students, in the UK and internationally. We deliver a range of professional development tools for practitioners and employees. We also accredit the computing degree courses in ninety eight universities around the UK.

As a leading IT qualification body, we offer a range of widely recognised professional and end-user qualifications.

www.bcs.org

Computing At School

Computing At School (CAS) is a membership association within BCS, The Chartered Institute for IT. The Computing at School Working Group (CAS) is a grass roots organisation focused on developing computer science as a proper, rigorous school subject.

CAS has 7,900+ members, with new members currently joining at a rate of over 400/month. Our members include school teachers, university academics, parents, school governors, members of professional societies, and IT professionals. We are supported by Google, Microsoft, ARM, Morgan Stanley, Ensoft and a range of other IT employers in the UK.

http://www.computingatschool.org.uk/

Table of Contents

This document

This document is the response of BCS and CAS to the Department for Education's consultation on the Subject Content of the new A levels[1].

We ***will*** also be responding to Ofqual's consultation on proposed changes to the assessment objectives and assessment arrangements.

Our response is focused solely on the proposed new A level in Computer Science and specifically refers to these questions in the response form:

- Q1 c) Computing.
- Q2 Will the revised A level content in each of these subjects enable students to progress to undergraduate study
- Q3 Are the new names for the A level subject content in Business studies (changed to 'Business') and Computing (changed to 'Computer science') appropriate?
- Q4 Is the revised AS qualification content in each of these subjects appropriate? Please provide evidence to support your response, specifically 4 c) Computing.
- Q5 Do you think that any of the proposals have the potential to have a disproportionate impact, positive or negative, on specific student groups, in particular the 'protected characteristic' groups?
- Q6 How could any adverse impact be reduced and how could the subject content of A levels be altered to better advance equality of opportunity between persons who share a protected characteristic and those who do not?

The radical changes in the school computing curriculum at Key Stages 1 to 4 now need to be reflected by a similarly radical reform to A-level content, since one should continue from the other. We therefore felt it necessary to structure our response in a way that combines our answers to the above questions, rather than adding essentially the same comments into every question in the response form.

Response to the DfE Consultation on A level subject content

We welcome the change of title, from "Computing" to "Computer Science". This fits with the terminology now in use at KS1-4, where "Computing" refers to the whole curriculum, embracing both computer science and ICT. For the A level, which has always been focused on computer science, the new title is now precise and unambiguous. It also strengthens the link with university degree courses with that name.

[1] https://www.education.gov.uk/consultations/index.cfm?action=consultationDetails&consultationId=1934&external=no&menu=1

1. Overall recommendations

We are happy with the overall structure:

- Aims and objectives
- Knowledge and understanding (what a student should know)
- Skills (what a student should be able to do)

(We find the above parenthesised sub-titles helpful.)

However, **the subject content for A level Computer Science needs significant revision**, for three main reasons:

- The radical changes in the KS1-4 curriculum[2] will mean that students starting A level will be much better prepared than in the past. Moreover, until 2010 there was not a single GCSE in Computer Science. Now every major awarding body is offering one. There can't *not* be a knock-on effect from such a major upheavals at KS1-4, although it will be some years before all of these changes feed through into A level.
- As can be seen from comments below, the current Subject Content really describes a *Software Engineering* A level, not a *Computer Science* one. We regard this as entirely inappropriate; software engineering is hard enough to teach at university level, let alone at school.
- The "knowledge and understanding" section is more appropriate, but is ill-balanced, and needs review in the light of the new Computing curriculum at KS1-4.

  This point and the previous one are discussed in more detail below, informed by the first.

Although we advocate significant revision, the Computer Science A level can still be launched in 2015, as planned. There is no need for the revisions to impose a delay in the timetable.

Firstly, the revisions are (we believe) not controversial, and could be done quite quickly, in a matter of weeks, during February/March 2014. Second, we do not anticipate that such work would require existing A levels to be rewritten, because they already take on board many of the above points; but they do so in spite of, rather than because of, the current specification.

2. The fate of AS level Computer Science

We are concerned about the possible loss of AS Level Computer Science. Many students take Computer Science as a fourth option (alongside Maths, Physics, Chemistry; or Maths, Further Maths, Physics, say). They do not feel committed to a full 2-year A level, but are willing to give it a try for a year. Many get hooked and do in fact continue, so the AS level serves as a very attractive "on-ramp" to the A level.

If there was no AS level, so students had to make an up-front choice whether to commit to a 2-year A level, we fear that they would drop Computer Science in favour of more traditional subjects.

Moreover, the AS level serves as a valuable complement to the traditional 3 'A' levels because of the growing recognition that Computer Science is a 'useful subject' in the latest edition of the Russell Group "Informed Choices" booklet.

We urge that the AS level maintains its vitality.

[2] https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study

3. Section "Aims and objectives"

The new KS1-KS4 Programme of Study[3] begins with a preamble entitled "Purpose of Study", that gives a vision of the richness and excitement of the subject. The A level specification would be stronger if it began with such a positioning statement, to support awarding bodies in making their qualifications express that sense of vision. One possible source of such a vision is the *CAS Curriculum for Computer Science*[4].

Here are some specific suggestions on the items in this section.

- "An understanding of computer systems". This seems unreasonably narrow. It would be better to replace it with the aim of the new KS1-KS4 Programme of Study: "can understand and apply the fundamental principles and concepts of computer science, including abstraction, decomposition, logic, algorithms and data representation"
- "The ability to apply skills, knowledge and understanding to solve problems". Again, it would be good to re-use the POS aim: "can analyse problems in computational terms, and have repeated practical experience of solving such problems, including writing programs to do so".
- "The capacity for thinking creatively…" and "Computational thinking skills…" These bullets are both concerned with thinking skills. Do they have to be separated?
- "Computational thinking skills, as set out in the attached annex". See the section below on the annex
- The last two bullets "an understanding of the consequences…" and "an awareness of … their impact…" are not usefully distinguishable and should be combined. However we strongly support the inclusion of an aim concerning the impact of computing and technology on society. Here is a suggestion: "Can critically articulate the individual (moral), social (ethical), and cultural opportunities and risks of digital technology". Notice "opportunities" as well as "risks", and "critically articulate" rather than just "understand".

4. Section "Knowledge and understanding"

General observations

- The current list of items under "Knowledge and understanding" mostly make some kind of sense individually, but: they are of hugely-varying size and importance; many are so vague that they could be satisfied by a course from the 1970s; others are too heavily based on recall.
- There is no mention of the words that one associates with a subject discipline: theory, model, proof, explanation, science, exploration, experiment, formalism, semantics, mathematical.
- Together they do not give any sense of the joy and beauty of computer science. As one member commented "we always manage to make the syllabus sound deadly dull and old-fashioned".

[3] https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study

[4] http://www.computingatschool.org.uk/index.php?id=cacfs

Some specifics

- Too vague and/or too recall-based:
  - "Possible future developments"
  - "The purpose and characteristics of a range of uses of computing"
  - "The need for and functions of systems software"
  - "The need for and attributes of different types of software"
  - "The data, processing, and communication requirements of computer systems".
- "Data types, data structures and algorithms" is just one bullet among many, but deserves far more emphasis. Computer science is full of elegant and useful algorithms and data structures, and they need explicit articulation. For example:
  - Algorithms: every A level should include study of a few classic algorithms. The choice can be left to awarding bodies, but might include shortest-path, sorting, searching, page-rank algorithms, etc.
  - Data structures: again every A level should study some classic data structures. The choice can be left to awarding bodies, but might include some among arrays (perhaps multi-dimensional), hash tables, stacks, queues, and pointer-based structures like lists and trees.
  - Students should learn to reason about algorithms. Whether or not the mathematical notation of O(n) is introduced, students should understand clearly that different algorithms have very different costs in time and space, and that these differences can become dominant as input sizes increase.
- Data representation (e.g. number representations such as two's complement and floating point, ASCII and Unicode, compound data such as postal addresses, etc.), compression, encryption, even simple information theory.
- Databases and large scale data analysis ("big data") deserve a mention. They were consciously omitted from the KS1-4 CAS Curriculum, but they deserve study at A level.

Overall, this section needs extensive reworking, in the light of the KS1-4 programme of study. We have not yet undertaken this work, but would be delighted to contribute to it.

5. Section "Skills"

The structure of the "Skills" section, which is intended to be practically oriented, is broken in Analysis, Design, Implementation, and Evaluation, a structure that is drawn from the "waterfall model" of software engineering. There are two major problems here:

1. The waterfall model of software development has been widely replaced in professional practice by "agile methods", scrums, test-driven development, continuous integration, and the like. The current content risks asking teachers to teach outdated methods to bored students, missing tremendous opportunities.
2. Software engineering focuses on the *organisational processes* by which software is produced by multi-person teams. It is an important subject, but it is one that is incredibly hard to teach until the student is a member of a large team developing a large project. It is very difficult to deliver a convincing software engineering course at university, because the teams and the projects are far too small, confined as they are by the number of hours of study.

Moreover, university teachers have seldom themselves had practical experience of building software at the scale in which these techniques become useful.

What is true of university is *a fortiori* true at school. It is counter-productive to attempt to teach the skills of managing a million-line multi-person project to students who are only writing small programs, often by themselves; and to ask teachers to do so who have no experience of using these techniques in anger themselves.

Astonishingly, the current Skills section does not even mention the word "programming", which is the practical skill that any computer scientist should have.

Here is an outline of a better (shorter, less prescriptive) skills section.

Students should:

- *Gain repeated experience of writing programs, in more than one language, to solve problems. (An A level student who knew only, say, Java, would be in serious danger of having learned Java (a particular technology) rather than programming (a life skill).)*
- *Develop a range of alternative solutions to a problem and evaluate the most appropriate for a particular context (for example, making appropriate trade-offs between programmer time, execution time, and storage requirements)*
- *Be able to explain to someone else how their program works, including the use of visualisations; and be able to argue for its correctness and efficiency using logical reasoning, test data, and user feedback. (NB not "prove its correctness".)*
- *Make good use of abstraction*
    - *to structure programs into modular parts with clear, well-documented interfaces*
    - *to model selected aspect of the external world in a program*
- *Carry out a substantial project, involving the design, implementation, and evaluation of a software or software/hardware system.*

6. Section "Annex: Computational Thinking Skills"

The emphasis on computational thinking skills is welcome, but the annex seems entirely unrelated. It is headed "Computational thinking skills" but sub-headed "Maths content". The latter appears to be the case; it has little to do with computational thinking skills.

Indeed, the entire Annex seems out of place; we understand it to be a requirement on all A level subject content specifications to express their Maths content. If so, so be it, but please don't mix it up with computational thinking.

7. Background

Computing at School (CAS) is a 7,900 strong working group whose single aim is to establish and nurture computer science as a foundational school subject, from primary school through to A level. CAS works in close partnership with BCS, The Chartered Institute for IT, which is the UK's professional body for our subject discipline.

The members of the working party that developed this response are:

- Professor Elizabeth Bacon (University of Greenwich)
- Dr Pete Bradshaw (Open University)
- Dr Neil Brown (University of Kent)
- Darren Christie (Peterborough Regional College)
- Dr Peter Dickman (Google)
- Dr Tom Crick (Metropolitan University of Cardiff)
- Ian Crosby (Hills Rd 6th form college)
- Clive Hirst (Merchant Taylors School)
- Professor Achim Jung (University of Birmingham)
- Peter Kemp (Teach First)
- Stuart Maher (Sir Christopher Hatton Academy)
- Martin McBride (Axlesoft Ltd)
- Hardip Mothada (Acton High School)
- John Palmer (The Chase School, Malvern)
- Professor Simon Peyton Jones (Microsoft Research Ltd)
- Hannah Robertson (Reigate College)
- Mark Scott (Bourne Grammar School)
- Dr Neil Smith (Open University)
- John Stout (King George V College)
- Jonathan Torbitt (Rendcomb College)
- Dr John Woollard (University of Southampton)