Spotlight on GCSE Computing – University of Warwick, 19th April 2012

Algorithms in the curriculum

Algorithms are specifically mentioned in the OCR curriculum and in the Computing at Schools suggested curriculum. To aid discussion we have extracted the relevant sections from both specification documents and reproduced them here.

OCR Curriculum

Algorithms are mentioned in section "2.1.7 Programming" under the heading "Algorithms" (page 11).

> *Candidates should be able to:*
> *(a) understand algorithms (written in pseudocode or flow diagram), explain what they do, and correct or complete them*
> *(b) produce algorithms in pseudocode or flow diagrams to solve problems.*

Algorithms are also mentioned under the heading "Control flow in imperative languages" and "Handling data in Algorithms" (also on page 11).

> *Candidates should be able to:*
> *(g) understand and use sequence in an algorithm*
> *(h) understand and use selection in an algorithm (IF and CASE statements)*
> *(i) understand and use iteration in an algorithm (FOR, WHILE and REPEAT loops).*

> *Candidates should be able to:*
> *(j) define the terms variable and constant as used in an imperative language*
> *(k) use variables and constants*
> *(l) describe the data types integer, real, Boolean, character and string*
> *(m) select and justify appropriate data types for a given program*
> *(n) perform common operations on numeric and Boolean data*
> *(o) use one-dimensional arrays.*

Algorithms are also discussed in section 2.3 (starts on page 14) with regard to the solution that students must developed in response to a stated problem.

The full OCR Curriculum is available from:

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

Computing at Schools Curriculum

The Computing at Schools curriculum discusses algorithms in some depth and much of the curriculum makes reference to the topic.

One of the key sections can be found on pages 13 and 14 under in section 4 (“Range and content: what a pupil should know”). Here are the relevant sections for key stages 2, 3 and 4.

KEY STAGE 2

- *Algorithms can be represented symbolically [flowcharts] or using instructions in a clearly defined language [turtle graphics].*
- *Algorithms can include selection (if) and repetition (loops).*
- *Algorithms may be decomposed into component parts (procedures), each of which itself contains an algorithm.*
- *Algorithms should be stated without ambiguity and care and precision are necessary to avoid errors.*
- *Algorithms are developed according to a plan and then tested. Algorithms are corrected if they fail these tests.*
- *It can be easier to plan, test and correct parts of an algorithm separately.*

KEY STAGE 3

- *An algorithm is a sequence of precise steps to solve a given problem.*
- *A single problem may be solved by several different algorithms.*
- *The choice of an algorithm to solve a problem is driven by what is required of the solution [such as code complexity, speed, amount of memory used, amount of data,the data source and the outputs required].*
- *The need for accuracy of both algorithm and data [difficulty of data verification;garbage in, garbage out]*

KEY STAGE 4

- *The choice of an algorithm should be influenced by the data structure and data values that need to be manipulated.*
- *Familiarity with several key algorithms [sorting and searching].*
- *The design of algorithms includes the ability to easily re-author, validate, test and correct the resulting code.*
- *Different algorithms may have different performance characteristics for the same task.*

The full Computing at Schools Curriculum is available from:

http://www.**computingatschool**.org.uk/data/uploads/**Computing**Curric.pdf