AS and A LEVEL

Transition Guide

H046, H446

Accredited

COMPUTER SCIENCE

Theme: Data types, data structures and algorithms

September 2015

CAMBRIDGE UNIVERSITY PRESS
www.cambridge.org

OCR
Oxford Cambridge and RSA

Mapping KS4 to KS5

Possible Teaching Activity (KS4 focus)

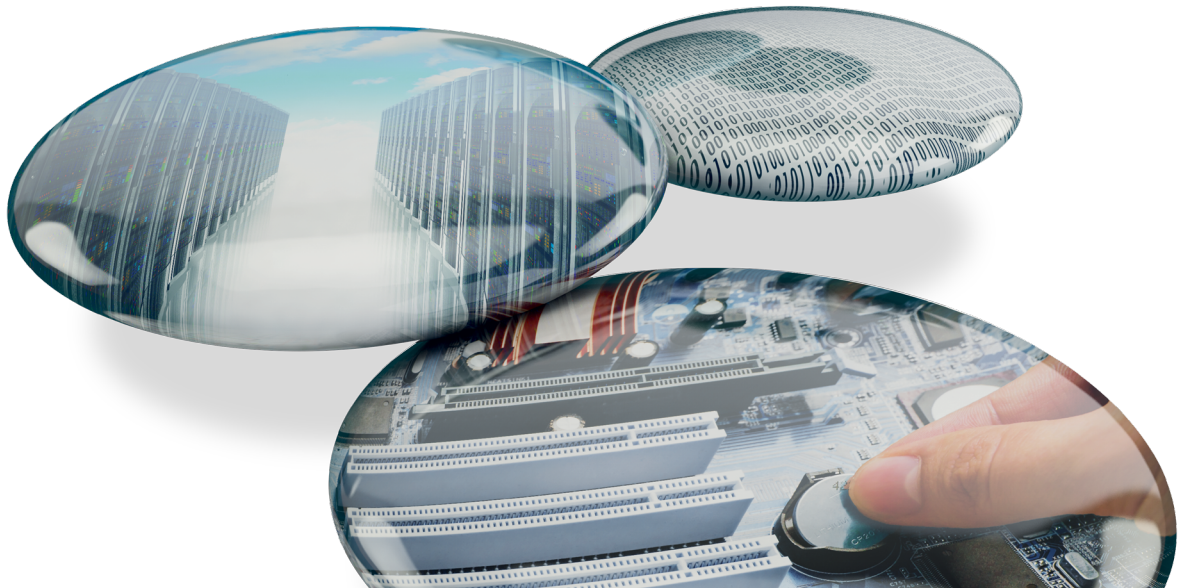
Checkpoint Task

Possible Teaching Activity (KS5 focus)

Challenge and Extension Task

Resources, Links and Support

Contents



Key Stage 4 Content

Key Stage 4 GCSE Content

2.1.2 Binary logic

Candidates should be able to:

d) explain why data is represented in computer systems in binary form
e) understand and produce simple logic diagrams using the operations NOT, AND and OR
f) produce a truth table from a given logic diagram.

2.1.4 Units

Candidates should be able to:

a) define the terms bit, nibble, byte, kilobyte, megabyte, gigabyte, terabyte
b) understand that data needs to be converted into a binary format to be processed by a computer.

2.1.4 Number

Candidates should be able to:

c) convert positive denary whole numbers (0-255) into 8-bit binary numbers and vice versa
d) add two 8-bit binary integers and explain overflow errors which may occur
e) convert positive denary whole numbers (0-255) into 2-digit hexadecimal numbers and vice versa
f) convert between binary and hexadecimal equivalents of the same number
g) explain the use of hexadecimal numbers to represent binary numbers.

2.1.4 Character

Candidates should be able to:

h) explain the use of binary codes to represent characters
i) explain the term character set
j) describe with examples (e.g. ASCII and Unicode) the relationship between the number of bits per character in a character set and the number of characters which can be represented.

Key Stage 5 Content

Key Stage 5 A Level Content

1.4 Data types, data structures and algorithms
How data is represented and stored within different structures. Different algorithms that can be applied to these structures

1.4.1 Data Types

a) Primitive data types, integer, real/floating point, character, string and Boolean.
b) Represent positive integers in binary.
c) Use of sign and magnitude and two's complement to represent negative numbers in binary.
d) Addition and subtraction of binary integers.
e) Represent positive integers in hexadecimal.
f) Convert positive integers between binary hexadecimal and denary.
g) Representation and normalisation of floating point numbers in binary.
h) Floating point arithmetic, positive and negative numbers, addition and subtraction.
i) Bitwise manipulation and masks: shifts, combining with AND, OR, and XOR.
j) How character sets (ASCII and UNICODE) are used to represent text.

1.4.2 Data Structures

a) Arrays (of up to 3 dimensions), records, lists, tuples.
b) The following structures to store data: linked-list, graph (directed and undirected), stack, queue, tree, binary search tree, hash table.
c) How to create, traverse, add data to and remove data from the data structures mentioned above. *(NB: this can be **either** using arrays and procedural programming **or** an object-oriented approach.)*

Key Stage 4 Content

2.1.7 Algorithms

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.

2.1.7 Control flow in imperative languages

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).

2.1.7 Handling data in algorithms

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.

2.3.1 Programming techniques

Candidates should be able to:

a) identify and use variables, operators, inputs, outputs and assignments
b) understand and use the three basic programming constructs used to control the flow of a program: Sequence; Conditionals; Iteration
c) understand and use suitable loops including count and condition controlled loops
d) use different types of data including Boolean, string, integer and real appropriately in solutions to problems
e) understand and use basic string manipulation
f) understand and use basic file handling operations: open, read, write and close
g) define and use arrays as appropriate when solving problems.

Key Stage 5 Content

1.4.3 Boolean Algebra

a) Define problems using Boolean logic. See Appendix 5d.
b) Manipulate Boolean expressions. Including the use of Karnaugh maps to simplify Boolean expressions.
c) Use the following rules to derive or simplify statements in Boolean algebra: De Morgan's Laws, distribution, association, commutation, double negation.
d) Using logic gate diagrams and truth tables. See Appendix 5d.
e) The logic associated with D type flip flops, half and full adders.

COMMENT

The main differences in content for the sections in the A-Level related to data types, data structures and algorithms are that of complexity and depth that students have to study the subject matter. Both qualifications require this knowledge for the written part of the assessment and certainly lends itself to the programming element of the controlled assessments of both qualifications though most that is covered in this section at A Level would not be used in this context, that is Boolean algebra is unlikely to make it into a project at A Level.

It is important to make the point that many students who take Computer Science for A Level may not have studied Computer Science at GCSE and so this may be the first time that students gain exposure to all of these topics. Therefore it could be a good consideration to be prepared to have a refresher on these topics, as well as some harder activities for those who understand them already.

Data types (A-Level section 1.4.1) need to be understood to some extent by GCSE students so they can complete their programming project (A453), particularly primitive data types.

The GCSE written paper (A451 sections entitled Binary logic, Units, Number, Character) also requires students to have knowledge of converting denary to binary and hexadecimal binary addition and how binary can be used to represent characters (ASCII and Unicode for example).

Students at GCSE can find this quite abstract at first as it is very unlike the maths that they are used to, but should find that given a few examples and exercises to go to and from each number representation it should become second nature to them.

Data types in A Level go more into depth such as use of Sign and Two's complement to represent negative numbers and representation of floating point numbers in binary as well as arithmetic of floating point values. Students can find this quite confusing at first, especially if they do not have a sound mathematical background. It is again advised that clear examples are given and a lot of practice.

Data structures in A Level cover arrays of up to three dimensions, records, lists and tuples, as well as linked-lists, graphs, trees and hash tables. Students are required to understand how to create, add/remove items and traverse these data structures. Students at GCSE do not need to understand data structures in quite so much depth, and may only ever encounter arrays/lists in limited capacities.

Boolean algebra (A Level section 1.4.3) is not studied in much depth at GCSE other than having to explain why data is represented in binary in computer systems and being able to construct truth tables for a given logic diagram. Students at GCSE need to understand OR, AND and NOT gates. Students at GCSE generally find this easy to grasp after a few examples.

Students at A Level however have to be able to define problems using Boolean logic, manipulate these expressions and use Karnaugh maps.

Many students at A Level find this way of thinking very confusing and need to be given lots of examples to follow and solve. Whereas the GCSE spec does not require a sound mathematic ability so much, the A Level spec requires students to think with a degree of mathematical logic, which some students may find particularly difficult and may require extra tuition.

Teaching Activity (KS4 focus)

Name: Terms & Resources

Code club

Various activities to support the teaching of Scratch and Python to young people
The activities include:
Scratch
– A painting program
Python
– Hangman
– Quiz
– Compliment generator
– Turtle graphics

Resource: http://projects.codeclub.org.uk/en-GB/index.html

Checkpoint task

Converting between denary, binary and hex

Task to introduce the idea of regular expressions

Teacher Instructions

http://www.ocr.org.uk/Images/253514-data-types-data-structures-and-algorithms-checkpoint-task-teacher-instructions.pdf

Learner Activity

http://www.ocr.org.uk/Images/253513-data-types-data-structures-and-algorithms-checkpoint-task-activity.doc

Teaching Activity (KS5 focus)

Codecademy – Python track

Codecademy

For students that are not confident with programming before the start of the course they could go away and work through the exercises at a website such as Codecademy

Codecademy covers topics such as:
- Variables and string manipulation
- Conditionals and functions
- Lists and dictionaries
- Looping
- Iterating over data structures
- Bitwise operators
- OOP
- File input/output

Resource: http://www.codecademy.com/en/tracks/python

Challenge and Extension Task

Name: 10 Mini Programming Projects

By 'Code? Boom' (Laura @codeboom)
You could get away with trying to code most of these in any programming language
Activities include:

- Hangman
- Noughts and Crosses
- Blackjack
- Mastermind
- Text adventure game
- Love Calculator
- Random Quote Generator
- Shoutbox
- Magic 8 Ball
- Meme Quiz

Resource: http://codeboom.wordpress.com/2012/07/30/10-mini-programming-projects/

The small print

We'd like to know your view on the resources we produce. By clicking on the 'Like' or 'Dislike' button you can help us to ensure that our resources work for you. When the email template pops up please add additional comments if you wish and then just click 'Send'. Thank you.

If you do not currently offer this OCR qualification but would like to do so, please complete the Expression of Interest Form which can be found here: www.ocr.org.uk/expression-of-interest

OCR Resources: *the small print*
OCR's resources are provided to support the teaching of OCR specifications, but in no way constitute an endorsed teaching method that is required by the Board and the decision to use them lies with the individual teacher. Whilst every effort is made to ensure the accuracy of the content, OCR cannot be held responsible for any errors or omissions within these resources. We update our resources on a regular basis, so please check the OCR website to ensure you have the most up to date version.

Please get in touch if you want to discuss the accessibility of resources we offer to support delivery of our qualifications: resources.feedback@ocr.org.uk

We will inform centres about any changes to the specification. We will also publish changes on our website. The latest version of our specification will always be the one on our website (www.ocr.org.uk) and this may differ from printed versions.

Copyright
OCR retains the copyright on all its publications, including the specifications. However, registered centres for OCR are permitted to copy material from this specification booklet for their own internal use.

ocr.org.uk/alevelreform
OCR customer contact centre

General qualifications
Telephone 01223 553998
Facsimile 01223 552627
Email general.qualifications@ocr.org.uk