Evaluation of the new GCSE Computer Science Specifications for first teaching September 2020

Version 4

Brought to you by www.gcsecs.org, the home of:

- Model answers to Programming Projects (former NEA) for teachers only
- The Ultimate GCSE Computer Science Textbook
- The Ultimate GCSE Computer Science Animated Presentations

The above textbook and presentations will be updated to reflect the new J277 specification in time for September 2020.

Please note that Eduqas has not been analysed due to its smaller market share and time constraints.

Contents

Introduction

This is an evaluation of the new GCSEs in Computer Science for first examination in summer 2022. This document compares the qualifications from AQA, Edexcel and OCR and evaluates the strengths and weaknesses of each. There are a number of side-by-side comparisons including all the topic areas covered.

Brought to you by www.gcsecs.org, the home of:

- Model answers to Programming Projects (former NEA) for teachers only
- The Ultimate GCSE Computer Science Textbook
- The Ultimate GCSE Computer Science Animated Presentations

The above textbook and presentations will be updated to reflect the new J277 specification in time for September 2020.

Specification Code Numbers

This table shows you the new entry code numbers for each exam board and the previous ones.

Spec Codes	AQA	Edexcel	OCR
New	8525	1CP2	J277
Previous	8520	1CP1	J276

The last entry for the previous specifications AQA 8520, Edexcel 1CP1 and OCR J276 is summer 2021.

The first entry for the new specifications AQA 8525, Edexcel (1CP2) and OCR J277 is summer 2022.

If you are currently teaching a 3 year KS4, then your current year 9 group should be following the new specifications 8520, 1CP2 or J276 as they will be assessed in summer 2022 when they are in year 11.

Programming Task

There is no longer a programming project (previously NEA) but all exam boards require students to complete a programming task or tasks that cover the areas of Design, Write, Test and Refine at some point during the course of study. None of the exam boards require students to complete any analysis or evaluation, use a set format or specific time limit. Centres will be required to sign a Practical Programming Statement (PPS) to confirm a Programming Task or tasks have been completed. It is up to the centre how these tasks are set. OCR have clarified that each task could cover any one or more of the Design, Write, Test and Refine skills required by Ofqual as long as each area is covered throughout the course. Edexcel have said the PPS simply needs to confirm that opportunities have been provided to design, write, test and refine during the course of study.

Mini projects that can be used as a Programming Tasks for any examination board will be available from www.gcsecs.org Any programming language can be used but it would be advisable to use the programming language that best matches the exam papers.

Exam Papers

Programming paper

The table below shows a breakdown of the 'programming' paper for each exam board.

Programming Paper	**AQA**	**Edexcel**	**OCR**
Paper Number	Paper 1	Paper 2	Paper 2
Title	Computational thinking and programming skills	Application of Computational Thinking	Computational thinking, algorithms and programming
Length	2 hr	2 hours	1 hr 30 mins
Number of marks	90	75	80 Section A: 50 Section B: 30
Assessment Value	50%	50%	50%
Topic areas	Sections 3.1 and 3.2 only but may draw on knowledge from 3.3 to 3.8	Topic 6 only but may draw on knowledge from topics 1-5	Section 2 only but may draw on knowledge from section 1
Paper or screen	Paper	IDE of choice using Python 3	Paper
Reference Code used in questions in exam papers	AQA's pseudo-code and (C# or Python 3 or VB.net depending on entry choice)	Edexcel's Programming Language Subset (PLS) which is a subset of Python 3	OCR's Exam Reference Language (ERL)
Methods for answering algorithm questions	AQA's Pseudocode Other pseudocode Flowchart	Python 3 (which can go beyond the PLS, but all questions can be answered within the PLS)	Pseudocode Flowcharts Bullet points ERL Any high-level programming language
Code for answering programming questions	C# or Python 3 or VB.net depending on entry choice		Any high-level programming language or OCR's ERL
Approach to syntax in answers	Emphasis on logical flow. Case of code is ignored.	Syntax should be correct for program to work as it's real programming.	Won't penalise "Low syntax errors"
Reference code booklet provided for exam	No	Yes – both printed and electronic	No
Flowchart symbols in questions	No defined set	Defined set	Defined set
Flowchart symbols in answers	Expected to be flexible as no defined set	Any understandable symbols in answers	No indication given by OCR
Other information		Pre-existing Python code will be available for refining, testing and correcting. May include text files including CSV	Sections A and B B: design, write, test, refine

One of the biggest choices centres will make will be based on whether the programming exam is on paper or on-screen. If a centre cannot possibly accommodate all students within the school day then AQA or OCR are the options to choose. Do note though that if all students can't be accommodated in a single session with Edexcel then it's still possible to let them take the exam during a different session on the same day as long as they are kept in supervised conditions during the scheduled time of the exam.

The main benefit of an on-screen exam is that students have a realistic programming experience (albeit without web access, access to code repositories or teams) and can debug their program as they go along and will know whether their programs will work as intended. There are also issues to watch out for with on-screen tests such as reliability of hardware and software. Similarly, if a candidate forgets the syntax or has an issue debugging then they may get stuck on a question. However, this wouldn't stop them from attempting the question like they would in a written paper. Edexcel allow the use of off-line help that is an integral part of the IDE. Edexcel have also published their ICE (Instructions for Conducting Exams) document which includes a section on contingency methods.

In previous exams, exam boards used their own versions of pseudo-code when presenting a question in an exam paper. This will now change to be a reference code for OCR but for Edexcel will be a subset of Python 3. For AQA, it will remain as pseudo-code. This means that for Edexcel, there is only a need for students to study Python 3 and no additional reference code.

Edexcel are the only exam board to have stated that any questions asked will be able to be answered within the scope of their reference code (subset of Python 3) which makes it easier for teachers to know the scope of their teaching. It doesn't stop students from using other acceptable answers.

OCR's approach to questions related to programming appears to be over complicated. Although they have said questions where students need to design, write, test or refine will be in section B of Paper 2, the variety of possible ways students can answer the question is confusing. OCR have said they will make it clear in each question how a student should answer, but this may still cause problems. Added to this, in section A of Paper 2, students will be able to respond using an algorithm method of their choice including pseudocode, flowcharts, bullet points, ERL or a high-level programming language. OCR's viewpoint is that in section A, "anything goes" but in section B the questions will be "clearly marked" to state the type of response needed. It would be helpful if OCR made it clear in section A the types of responses that would be acceptable for an algorithm as "write an algorithm" may put some students off drawing a flowchart, even though it would be an acceptable response.

Non-programming paper

The table below shows a breakdown of the 'theoretical' paper (non-programming) for each exam board:

'Theoretical' Paper	AQA	Edexcel	OCR
Paper Number	Paper 2	Paper 1	Paper 1
Title	Computing concepts	Principles of Computer Science	Computer Systems
Length	1 hr 45 mins	1 hr 30 mins	1 hr 30 mins
Number of marks	90	75	80
Assessment Value	50%	50%	50%
Topic areas	Sections 3.3 to 3.8	Topics 1-5 only	Section 1 only
Extended response questions	One 9-mark question	One 6-mark question	One 8-mark question
Other information		5 compulsory questions, one for each topic	

Assessment Objectives

This table shows how each exam board has approached assessment objectives for their exams.

Non-programming paper	AQA	Edexcel	OCR
AO1 Knowledge & Understanding	25.6%	30%	21%
AO2 Application	20%	20%	29%
AO3 Problem solving	4.4%	0%	0%

Programming paper	AQA	Edexcel	OCR
AO1 Knowledge & Understanding	4.4 %	0%	9%
AO2 Application	20%	20%	11%
AO3 Problem solving	25.6%	30%	30%

Both OCR and Edexcel have made it clear in their assessment objectives that algorithms and programming will only be tested in their programming paper whereas AQA have left 4.4% within the non-programming paper to cover SQL questions, but it does mean that all 3 exam boards have completely separated programming from other topics, apart from synoptic questions in the programming paper which may draw upon knowledge from the non-programming topics.

Command words

OCR have stated they will only use command words from their defined list of 29 command words for GCSE Computer Science, most of which differ significantly from Ofqual's Access by Fair Design definitions of command words produced for the Welsh Assembly. Edexcel have also produced a command word taxonomy which is limited to 13 command words, although there are some differences with Ofqual's Access by Fair Design definitions. AQA have not yet produced any guidance on command words but their previous 8520 specification had 15 command words which, out of the 3 exam boards, most closely followed Ofqual's Access by Fair Design. Although Ofqual's Access by Fair Design was produced for the Welsh Assembly (and updated in 2019), it's still the most up-to-date guidance from Ofqual in relation to command words and designing exam papers consistently and fairly.

I am personally of the opinion that there is a big problem in the exam system with different exam boards using different meanings for command words and even different subjects within the same exam boards using different meanings. This means students have to learn a different exam technique for every exam they take, and remember what each one is requiring. Ofqual produced a document called Access by Fair Design which had guidance on how command words should be used which I believe should be followed by all subjects for all exam boards at all levels, even though it was commissioned by the Welsh Assembly.

Accessibility and clarity

OCR have committed to their papers being accessible and have set out 13 guidelines they will follow following "The Cambridge Approach" (https://www.cambridgeassessment.org.uk/Images/cambridge-approach-to-assessment.pdf).

In my experience, AQA's questions have traditionally been the most clear and concise of the 3 exam boards with very little ambiguity.

In my experience, AQA have always been the most flexible when it comes to mark schemes and assessment by allowing unexpected but correct answers throughout the whole marking process. OCR have previously been very rigid with their mark schemes and once a mark scheme is locked in, it is unusual for unexpected correct answers to be awarded marks. OCR's approach is consistent, but it does mean it can be consistently wrong. Edexcel have confirmed their approach to mark schemes does allow unexpected but correct answers.

Subject Content

Each exam board has taken the opportunity to remove, add and clarify content within their specification. OCR have stuck most closely to their original specification which helps with continuity of teaching. Edexcel have made the most structural changes to their specification, but this was necessary as their previous specification was disjointed and the new one is far better structured, although there is a lot of repetition between sections 1 (computational thinking) and 6 (application of computational thinking). The reason for the repetition is so questions can be asked in the non-programming paper about programming concepts whereas the programming paper is about using those concepts.

AQA Key Changes

The key changes that AQA have made from 8520 to 8525 are:

New areas to teach include (numbers relate to new 8525 specification):

- 3.2 a programming language that can be tested in the exam – choice of C#, Python or VB.net
- 3.2.2.b count controlled and condition controlled terminology
- 3.2.6.c specific pseudo-code for RECORD definition
- 3.2.11.f syntax and logic errors
- 3.2.11.g identify and categorise errors
- 3.3.3.b compare quantities of bytes
- 3.4.2.a XOR gates, including in 3.4.2c circuit diagrams
- 3.4.2.d/e use of Boolean expressions including creating logic circuits from a Boolean expression and vice versa
- 3.4.4.a differences between high-level and low-level languages
- 3.4.4.c understanding machine code is specific to a family of processors
- 3.4.4.e understand that assembly language is not translated
- 3.4.5.a role of registers in the CPU (knowledge of individual registers not required)
- 3.4.5.e understand different types of memory and their role
- 3.5.0.a disadvantages, not just risks, of networks
- 3.5.0.b disadvantages, not just risks, of wireless networks
- 3.6.2.a pharming as a cyber attack rather than a social engineering method
- 3.7.1 relational databases and SQL including reading, inserting, updating and deleting
- 3.8.0.a issues related to autonomous vehicles

Areas you no longer need to teach include (numbers relate to old 8520 specification):

- 3.2.7.c read/write from/to a text file
- 3.3.6.a VDUs
- 3.3.8.b **create** Huffman trees
- 3.4.4.a Von Neumann architecture (although still the concept)
- 3.4.4.c effect of cache type on performance of CPU
- 3.6.1.e pharming as a social engineering method (but it is now in 8525 specification as a cyber attack method)
- 3.6.1.hadware
- 3.7.0.a ethical issues related to:
    - theft of computer code
    - copyright of algorithms
    - cracking

The main additions for AQA are XOR gates (not difficult), databases and SQL (including inserting, updating and deleting) and Boolean expressions, including creating logic circuits from a Boolean expression and vice versa. The main content areas that have been removed are file handling in programming and creating Huffman trees, although students still need to be able to interpret a Huffman tree.

Edexcel Key Changes

The key changes that Edexcel have made from 1CP1 to 1CP2 are:

New areas to teach include:

- Iteration over every item in data structures
- Efficiency of algorithms in terms of number of compares, number of passes, use of memory
- Determine maximum number of states represented by a binary bit pattern of a given length
- Convert 8 bit two's complement binary to decimal and vice versa
- Kibibyte, mebibyte, gibibyte and tebibyte
- Be able to construct expressions to calculate file sizes and data capacity requirements
- Impact of wired and wireless connectivity on performance
- Construct expressions involving file size, transmission rate and time
- Ethical and legal issues associated with artificial intelligence, machine learning and robotics
- Malware
- Methods of protecting digital systems
- Convert algorithms to programs and vice versa
- Single entry/exit points from code blocks
- Read/write to comma separated value text files (for exam purposes)
- New flowchart symbol for "pre-defined subprogram":

Areas you no longer need to teach include:

- Investigating requirements
- Test plans and test data
- Interpreting error messages
- Sign and magnitude version of signed integers
- JPG and MP3 examples of lossy algorithms
- Run-length encoding
- Caesar cipher
- Structured and unstructured data
- Databases
- Input-process-output model
- Input devices
- ROM and cache
- Storing date in the 'cloud'
- Produce logic statements
- Simulation and modelling
- Assemblers
- Client-server and peer-to-peer
- Ring topology
- Commercial analysis tools and review of network and user policies
- world wide web and its components

OCR Key Changes

The key changes that OCR have made from J276 to J277 are:

New areas to teach include:

- The need for primary storage
- Bluetooth
- Encryption (not just in relation to Wi-Fi)
- Network standards
- Physical security
- Identifying inputs, processes and outputs of a problem
- Structure diagrams for a problem
- Trace tables
- Random number generation
- Refining algorithms
- Data Protection Act 2018 (rather than 1998)
- Additional levels (more than 2) of combining Boolean operators in a logic diagram

Areas you no longer need to teach include:

- Virtual networks
- Wi-Fi frequency and channels
- Packet switching
- Network policy
- Network forensics
- Full and incremental backups
- Creative Commons Licensing
- Freedom of Information Act 2000
- Input sanitisation
- Planning for contingencies
- Characteristics of an assembler
- Check digits

You can download for free a full side-by-side analysis for each exam board's transition from their old specification to the new specification at the top of the page at www.gcsecs.org

Side by side comparison of topics between exam boards

This table shows how each exam board has covered topic areas within their specifications. The original spreadsheet used to create this table and an expanded worksheet that includes the specification content alongside the codes can be found at the top of the page at www.gcsecs.org Note that in addition to Edexcel's specification, their Getting Started Guide has been used to identify where content is covered.

Number	Topic	AQA	Edexcel	OCR
1	Hardware vs software	3.4.1.a	-	-
2	Purpose of CPU	3.4.5.a	3.1.1	1.1.1.a
3	Fetch-execute cycle	3.4.5.c	3.1.1	1.1.1.a
4	ALU	3.4.5.a	3.1.1	1.1.1.b
5	Control Unit	3.4.5.a	3.1.1	1.1.1.b
6	Clock	3.4.5.a	3.1.1	-
7	Cache	3.4.5.d	-	1.1.1.b
8	Registers	3.4.5.a/d	3.1.1	1.1.1.b
9	Bus	3.4.5.a	3.1.1	-
10	MAR	-	-	1.1.1.c
11	MDR	-	-	1.1.1.c
12	Program Counter	-	-	1.1.1.c
13	Accumulator	-	-	1.1.1.c
14	Von Neumann stored program concept	-	2.1.1	-
15	Effect of clock speed, cache size, number of cores	3.4.5.b	-	1.1.1.d
16	How data is passed through registers	-	3.1.2	-
17	Purpose of embedded systems	3.4.5.j	3.1.3	1.1.1.e
18	Examples of embedded systems	3.4.5.j	3.1.3	1.1.1.e
19	Purpose of main memory	3.4.5.a	3.1.1	1.2.1.a
20	Difference between RAM and ROM	3.4.5.e	-	1.2.1.b
21	Purpose of RAM	3.4.5.d	3.1.1	1.2.1.d
22	Purpose of ROM	3.4.5.d	-	1.2.1.c
23	How virtual memory works	-	-	1.2.1.e
24	Purpose of secondary storage	3.4.5.f	3.1.2	1.2.2.a
25	Difference between primary and secondary storage	3.4.5.e	-	-
26	Optical storage	3.4.5.g	3.1.2	1.2.2.b
27	Magnetic storage	3.4.5.g	3.1.2	1.2.2.b
28	Solid state storage	3.4.5.g	3.1.2	1.2.2.b
29	How storage methods work	3.4.5.g	3.1.2	-
30	Cloud storage	3.4.5.h	-	*(1.3.1.e)*
31	Cloud storage vs local storage	3.4.5.i	-	*(1.3.1.e)*
32	Selecting a suitable storage device	-	-	1.2.2.c
33	Advantages and disadvantages of storage devices	3.4.5.g	-	1.2.2.d
34	Bit	3.3.3.a	2.3.1	1.2.3.a
35	Nibble	-	2.3.1	1.2.3.a
36	Byte	3.3.3.a	2.3.1	1.2.3.a
37	kB – TB	3.3.3.b	-	1.2.3.a
38	PB	-	-	1.2.3.a

Number	Topic	AQA	Edexcel	OCR
39	Kibibyte (KiB) – tebibyte (TiB)	-	2.3.1	-
40	Number bases	3.3.1.a	*(2.1.3)*	*(1.2.4.a)*
41	Binary representation	3.3.1.b	*(2.1.3)*	*(1.2.4.a)*
42	Hexadecimal representation	3.3.2.b	*(2.1.6)*	*(1.2.4.c)*
43	Purpose of binary format	3.3.1.b	2.1.1	1.2.3.b
44	Calculate states represented by a binary bit pattern	-	2.1.1	-
45	Purpose of hexadecimal	3.3.1.c	2.1.6	-
46	Calculate text file size	3.3.8.c	2.3.1	1.2.3.c
47	Unsigned integers using two's complement	-	2.1.2	-
48	Denary to binary conversion	3.3.2.c	2.1.3	1.2.4.a
49	Binary to denary conversion	3.3.2.c	2.1.3	1.2.4.a
50	Signed binary/denary conversion	-	2.1.3	-
51	Denary to hexadecimal conversion	3.3.2.c	-	1.2.4.c
52	Hexadecimal to denary conversion	3.3.2.c	-	1.2.4.c
53	Binary to hexadecimal conversion	3.3.2.c	2.1.6	1.2.4.d
54	Hexadecimal to binary conversion	3.3.2.c	2.1.6	1.2.4.d
55	Binary addition	3.3.4.a	2.1.4	1.2.4.b
56	Logical binary shifts	3.3.4.b	2.1.4	1.2.4.e
57	Need for binary shifts	3.3.4.c	-	-
58	Arithmetic binary shifts	-	2.1.4	-
59	Overflow	-	2.1.5	1.2.4.b
60	Character codes are binary	-	2.3.1	1.2.4.f
61	Character set	3.3.5.a	*(2.2.1)*	1.2.4.g
62	7-bit ASCII	3.3.5.a	2.3.1	1.2.4.h
63	Extended ASCII	-	-	*(1.2.4.h)*
64	Unicode	3.3.5.a	-	1.2.4.h
65	Character sets are a sequence	3.3.5.b	2.2.1	1.2.4.g
66	ASCII vs Unicode	3.3.5.c	2.2.1	1.2.4.h
67	Understanding of pixels	3.3.6.a	2.2.2	1.2.4.i
68	Image size / resolution	3.3.6.b	2.2.2	*(1.2.4.k)*
69	Colour depth	3.3.6.b	2.2.2	*(1.2.4.k)*
70	How a bitmap is represented	3.3.6.c	2.2.2	1.2.4.i
71	Metadata	-	-	1.2.4.j
72	Effect of resolution and colour depth	3.3.6.d	2.2.4	1.2.4.k
73	Calculate image file size	3.3.6.e	2.3.1	1.2.3.c
74	Convert binary to image	3.3.6.f	2.2.2	-
75	Convert image to binary	3.3.6.g	-	-
76	Need for analogue to digital conversion	3.3.7.a	*(2.2.3)*	1.2.4.l
77	Sampling	3.3.7.b	2.2.3	1.2.4.l
78	Effect of sample rate, duration, bit depth	3.3.7.c	2.2.4	1.2.4.m
79	Calculate sound file size	3.3.7.d	2.3.1	1.2.3.c
80	Purpose of compression	3.3.8.a	2.3.2	1.2.5.a
81	Lossy compression	*(3.3.8.a)*	2.3.2	1.2.5.b

Number	Topic	AQA	Edexcel	OCR
82	Lossless compression	*(3.3.8.a)*	2.3.2	1.2.5.b
83	Interpret Huffman trees	3.3.8.b	-	-
84	Calculate Huffman storage	3.3.8.c	-	-
85	Understand run length encoding (RLE)	3.3.8.d	-	-
86	Represent data using RLE	3.3.8.e	-	-
87	Truth Tables	3.4.2.a	1.3.1	2.4.1.b
88	Construct truth tables for logic circuits	3.4.2.b	1.3.1	2.4.1.d
89	Interpret results of truth tables	3.4.2.b	1.3.1	-
90	Create/modify/interpret logic circuit diagrams	3.4.2.c	-	2.4.1.a
91	Create / interpret Boolean expressions	3.4.2.d	-	2.4.1.c
92	Create Boolean expression for a logic circuit	3.4.2.e	-	-
93	Create logic circuit from a Boolean expression	3.4.2.e	-	2.4.1.d
94	Systems software	3.4.3.a	*(3.2.2)*	*(1.5.2.b)*
95	Applications software	3.4.3.a	-	-
96	Operating system functions	3.4.3.b	3.2.1	1.5.1.a
97	Utility programs	3.4.3.a	3.2.2	1.5.2.a/b
98	High-level vs low-level languages	3.4.4.a	3.3.1	2.5.1.a
99	Machine code and assembly code	3.4.4.b	*(3.3.1)*	-
100	Purpose of translation	3.4.4.c	*(3.3.2)*	2.5.1.b
101	Advantages and disadvantages of low-level / high-level	3.4.4.d	*(3.3.1)*	-
102	Compilers & interpreters	3.4.4.e	3.3.1	2.5.1.c
103	Assemblers	3.4.4.e	-	-
104	Integrated Development Environment (IDE)	-	-	2.5.2.a
105	Network purpose	3.5.1.a	4.1.1	-
106	Advantages and disadvantages of networks	3.5.1.a	-	-
107	Personal Area Network	3.5.1.b	-	-
108	LAN and WAN	3.5.1.b	4.1.2	1.3.1.a
109	Wired and wireless advantages & disadvantages	3.5.1.c	4.1.4	1.3.2.a
110	Star topology	3.5.1.d	4.1.8	1.3.1.f
111	Bus topology	3.5.1.d	4.1.8	-
112	Mesh topology	-	4.1.8	1.3.1.f
113	Factors affecting network performance	3.5.1.c	4.1.4	1.3.1.b
114	Network hardware	-	-	1.3.1.d
115	Client-server and peer-to-peer	-	-	1.3.1.c
116	Internet structure	-	4.1.3	1.3.1.e
117	Purpose of protocols / standards	3.5.1.e	4.1.6	1.3.2.d
118	Protocols	3.5.1.f	4.1.6	1.3.2.e
119	Measuring network speeds	-	4.1.5	-
120	Need for network security	3.5.1.g	4.2.1	-
121	TCP/IP model	3.5.1.i	4.1.7	-
122	Concept of layers	-	-	1.3.2.f
123	IP and MAC addressing	*(3.5.1.f)*	*(4.1.6)*	1.3.2.c
124	MAC address filtering	3.5.1.h	-	-

Number	Topic	AQA	Edexcel	OCR
125	Purpose of cyber security	3.6.1.a	4.2.1	-
126	Encryption	3.5.1.h	5.3.2	1.3.2.b, 1.4.2.a
127	Access control / passwords / authentication	3.5.1.h, *(3.6.2.a)*	4.2.1	1.4.2.a
128	Firewall	3.5.1.h	4.2.1	1.4.2.a
129	Anti-malware	3.6.2.2	5.3.1/2	1.4.2.a
130	Physical security	-	4.2.1	1.4.2.a
131	Penetration testing / ethical hacking	3.6.2.b	4.2.1	1.4.2.a
132	Audit trails	-	3.2.3	-
133	Code reviews	-	3.2.3	-
134	Biometric security	3.6.3.a	*(3.2.1)*	-
135	CAPTCHA	3.6.3.a	-	-
136	2FA by email	3.6.3.a	-	-
137	Automatic software updates	3.6.3.a	-	-
138	Acceptable use policy	-	5.3.2	-
139	Backup and recovery	-	5.3.2	-
140	Social engineering	3.6.2.1	5.3.1	1.4.1.a
141	Pharming	3.6.2.a	-	-
142	Malware	3.6.2.2	5.3.1	1.4.1.a
143	Brute force attacks	-	-	1.4.1.a
144	Denial of service attacks	-	-	1.4.1.a
145	Data interception and theft (hacking)	3.8.1.a	-	1.4.1.a
146	SQL injection	-	-	1.4.1.a
147	Weak passwords	3.6.2.a	-	-
148	Misconfigured access rights	3.6.2.a	-	-
149	Removable media	3.6.2.a	-	-
150	Unpatched or out of date software	3.6.2.a	5.3.1	-
151	Concept of a database	3.7.1.a	-	*(2.2.3.d)*
152	Concept of a relational database	3.7.1.b	-	-
153	Relational database structure	3.7.1.c	-	-
154	Data redundancy and inconsistency	3.7.1.c	-	-
155	SQL Select	3.7.2.a	-	2.2.3.d
156	SQL Insert	3.7.2.b	-	-
157	SQL Update	3.7.2.c	-	-
158	SQL Delete	3.7.2.c	-	-
159	Environmental issues	3.8.1.a	5.1.1	1.6.1.a
160	Ethical issues 1	3.8.1.a	5.2.1/2/3	1.6.1.a
161	Legal issues	3.8.1.a	5.2.1/2/3	1.6.1.a
162	Cultural issues	-	-	1.6.1.a
163	Privacy issues	3.8.1.a	5.2.1	1.6.1.a
164	Data Protection Act	*(3.8.1.a)*	*(5.2.1)*	1.6.1.b
165	Computer Misuse Act	*(3.8.1.a)*	*(5.2.1)*	1.6.1.b
166	Copyright, Designs and Patents Act	-	*(5.2.3)*	1.6.1.b

Number	Topic	AQA	Edexcel	OCR
167	Software licences	-	5.2.3	1.6.1.b
168	Abstraction	3.1.1.c	1.1.1, 6.1.1	2.1.1.a
169	Decomposition	3.1.1.b	1.1.1, 6.1.1	2.1.1.a
170	Concept of an algorithm	3.1.1.a	*(1.2.1)*	*(2.1.2.c)*
171	Algorithmic thinking	3.1.1.d	1.2.1	2.1.1.a
172	Identify inputs, processes, outputs for algorithms	3.1.1.e	1.2.1	2.1.2.a
173	Structure diagrams	-	-	2.1.2.b
174	Pseudocode / flowcharts / high level programming	3.1.1.d	1.2.1	2.1.2.c
175	Multiple algorithms to solve same problem	3.1.2.a	-	-
176	Compare efficiency of algorithms	3.1.2.b	1.2.7, 6.1.6	-
177	Trace tables	3.1.1.f	1.2.4	2.1.2.e
178	Linear search	3.1.3.a	1.2.6	2.1.3.a
179	Binary search	3.1.3.b	1.2.6	2.1.3.a
180	Compare and contrast linear & binary search	3.1.3.c	1.2.6, *(1.2.7), (6.1.6)*	-
181	Merge sort	3.1.4.a	1.2.6	2.1.3.b
182	Bubble sort	3.1.4.b	1.2.6	2.1.3.b
183	Insertion sort	-	-	2.1.3.b
184	Compare and contrast merge & bubble sort	3.1.4.c	1.2.6, *(1.2.7), (6.1.6)*	-
185	Concept of data types	3.2.1.a	6.3.1	2.2.2.a
186	Use of data types	3.2.1.a	6.3.1	2.2.2.a
187	Variable declaration and assignment & constants	3.2.2.a	1.2.2, 6.2.1, 6.3.2	2.2.1.a
188	Sequence, selection & iteration	3.2.2.a	1.2.1 6.2.1/2	2.2.1.b
189	Iteration through data structures	*(synoptic)*	1.2.1, 6.2.2	*(synoptic)*
190	Single entry/exit points for code blocks	-	6.2.2	-
191	Count-controlled and condition-controlled loops	3.2.2.b	1.2.1 6.2.2	2.2.1.b
192	Nested selection and iteration	3.2.2.c	-	-
193	Arithmetic operators	3.2.3.a	1.2.3, 6.5.1	2.2.1.c
194	Boolean operators	3.2.5.a	1.2.3, 6.5.3	2.2.1.d
195	Relational operators	3.2.4.a	1.2.3, 6.5.2	*(2.2.1.b)*
196	One and two-dimensional arrays	3.2.6.a/b	1.2.2, 6.3.1	2.2.3.e
197	Records as a data structure	3.2.6.c	1.2.2, 6.3.1	2.2.3.c
198	String manipulation	3.2.8.a	6.3.3	2.2.3.a
199	Obtain input from the user	3.2.7.a	6.4.1	2.2.1.a
200	Output data to a display	3.2.7.b	*(6.4.1)*	2.2.1.a
201	File handling	-	6.4.2	2.2.3.b
202	Comma separated files	-	6.4.2	-
203	Programming for validation	3.2.11.a	6.4.3	2.3.1.b
204	Programming for authentication	3.2.11.b	6.4.4	2.3.1.a
205	Anticipating misuse	-	-	2.3.1.a
206	Random number generation	3.2.9.a	-	2.2.3.g
207	Concept of subroutines	3.2.10.a	6.6.1	2.2.3.f

Number	Topic	AQA	Edexcel	OCR
208	Advantages of subroutines	3.2.10.b	1.1.2	*(2.2.3.f)*
209	Use of parameters	3.2.10.c	6.6.2	*(2.2.3.f)*
210	Functions with return values	3.2.10.d	6.6.2	*(2.2.3.f)*
211	Local /global variables	3.2.10.e	6.6.3	*(ERL)*
212	Advantages of local variables	3.2.10.f	-	-
213	Describe structured approach to programming	3.2.10.g	-	-
214	Advantages of the structured approach	3.2.10.h	-	-
215	Meaningful identifier names	3.2.2.d	6.1.4	2.3.1.c
216	Other methods for maintainable code	-	6.1.4	2.3.1.c
217	Purpose of testing	3.2.11.c	*(1.2.7), (6.1.6)*	2.3.2.a
218	Correct errors	3.2.11.c	1.2.5, 6.1.5	2.3.2.e
219	Types of test data	3.2.11.d	*(6.1.6)*	2.3.2.d
220	Choose suitable test data	3.2.11.e	*(1.2.7), (6.1.6)*	2.3.2.d
221	Syntax errors	3.2.11.f	1.2.5, 6.1.5	2.3.2.c
222	Logic errors	3.2.11.f	1.2.5, 6.1.5	2.3.2.c
223	Runtime errors	-	1.2.5, 6.1.5	-
224	Identify and categorise errors 1	3.2.11.g	1.2.5	2.1.2.c/d
225	Iterative vs final/terminal testing	-	-	2.3.2.b
226	Read, write, analyse & refine programs	*(PT)*	6.1.2	*(PT)*
227	Convert algorithms into programs	-	6.1.3	-

* brackets denote the content is implicitly required rather than explicitly stated

* “synoptic” denotes that the content could be tested by combining different elements of the specification

* “pseudo”, “PLS”, “ERL” denotes that the concept is included within the pseudocode (AQA) / Programming Language Subset (Edexcel – Python) / Exam Reference Language (OCR)

* “PT” denotes coverage in the programming task(s)

Review of content

Overall, AQA is heaviest on subject content with additional challenges such as Huffman trees, RLE, SQL and relational databases but it is also the clearest in terms of which aspects of each topic will be covered. Edexcel appears on the surface to be light on content, but this is more because the specification is vague. However, Edexcel have produced a Getting Started Guide which expands on the specification content to make it clear what elements should and shouldn’t be covered. An early preview of this document provided by Edexcel was very clear and they were very responsive to feedback. OCR has a good balance of subject content and has improved its specification over the previous one significantly with the introduction of the “guidance” column which makes the scope of coverage much clearer. With the clarification that the movement of data between CPU registers is not required, this makes OCR’s coverage much better balanced.

All exam boards require students to be able to understand calculations but Edexcel only require students to create expressions whereas OCR and AQA require the actual calculation to be made. OCR and AQA both use numbers that are easy to use without a calculator. Edexcel focus on testing the process of the calculation rather than the end result.

AQA and OCR both offer additional information for each learning objective in the specification which attempts to clarify the scope of learning that is required and clarify any misconceptions. Edexcel are the only exam board not to have done this but they are finalising a Getting Started Guide which offers detailed clarification of each specification point.

Other considerations

Responsiveness of exam boards

When providing feedback to each of the examination boards on their specifications, Edexcel were the most open and responsive. Edexcel published a draft of their specification and invited feedback and then addressed about 60% of the issues I had highlighted. They also gave specific feedback on the comparison document I produced for 1CP1 to 1CP2. Further to this, Edexcel provided an early draft copy of their Getting Started Guide and they took on board most of the feedback I gave. AQA replied to my feedback and gave reasons for their decisions but they did not have a public opportunity for feedback. AQA's copyright team have also been deliberately obstructive in relation to the comparison documents which means teachers will have to do all the cross-referencing themselves as I've had to remove any direct statements from their specification. OCR uses focus groups and market research companies for feedback. While OCR and AQA received my feedback on their specifications, they were unable to act on any of it as the qualification had already been accredited by Ofqual when they made it public, although AQA did acknowledge they were able to change the spelling mistakes and OCR have indicated they may be able to do some 'tidying up'. OCR provided feedback on this document and their comments have been incorporated into the second draft. It should be noted that OCR did correct some of the sound engineering terminology as a response to feedback that was given before their specification was developed.

It was very disappointing that AQA requested that any copies of their subject content be removed from the comparison documents. This makes it much harder to compare AQA with other exam boards and even the transition to AQA's new specification. Although the specification content code references remain (as these are not copyright), it does provide an added burden to teachers when reviewing AQA's specification changes.

Grade boundaries

Some centres choose an exam board because it has low grade boundaries thinking this makes the exam easier to achieve marks. However, this is a misconception. Grade boundaries are set based on a number of factors including comparison with previous series of exams and statistical information. A very low grade boundary usually means the exam was set at too difficult a level or the mark scheme was too strict while a very high grade boundary could indicate the exam was set at too easy a level. Higher grade boundaries usually mean a more accessible exam paper whereas lower grade boundaries can be demotivating for students sitting practice papers that are very difficult to achieve marks on.

It's also important that the exam papers are well differentiated between grades 1 and 9 which means the more 'distance' between grade boundaries, the better differentiated the paper is. Closer grade boundaries indicate a poorly written exam paper that hasn't differentiated students sufficiently and risks students dropping a grade by losing just a few marks.

For the legacy Computer Science qualifications, AQA has the best differentiation at 78% between grades 9 and 1 but Edexcel has a poor 61% and OCR is reasonable at 73%. Similarly AQA's average grade boundary for a grade 9 was 89% with OCR at 85% and Edexcel at 69%. Edexcel have addressed these issues for 1CP2 by streamlining the new specification, reducing the number of command words, having one question per topic and simplifying the technical language used in exams. It's good to see Edexcel responding to this historical problem. The table below shows all the average grade boundaries for each exam board for 2018 and 2019.

	9	8	7	6	5	4	3	2	1	Differentiation
AQA	89%	82%	76%	66%	56%	46%	34%	23%	11%	78%
Edexcel	69%	60%	52%	45%	38%	31%	23%	15%	8%	61%
OCR	85%	78%	70%	61%	53%	44%	33%	23%	13%	73%
Average	81%	73%	66%	57%	49%	40%	30%	20%	11%	70%

Another factor to consider is how many students achieve each grade. However, this doesn't take into account the performance of each cohort at KS2 and KS3 and so isn't a precise like for like comparison. This table shows the average percentage of students achieving each grade in 2018 and 2019.

	9	8	7	6	5	4	3	2	1	Entries
AQA	5%	14%	24%	39%	53%	65%	80%	90%	97%	14459
Edexcel	4%	11%	21%	32%	46%	62%	78%	89%	97%	6311
OCR	4%	11%	21%	34%	48%	62%	76%	88%	97%	52647
Average	4%	12%	22%	35%	49%	63%	78%	89%	97%	

Customer base

From the table above it can be seen that OCR has by far the largest customer base. This is because OCR were first to market with a GCSE in Computing prior to the change to 9-1 GCSE Computer Science. OCR have maintained this dominance in the market which means they are able to invest more into the qualification. Although many centres will maintain 'loyalty' to an exam board, this is also an opportunity to move. Often publishers will focus on exam boards that have the larger market share, but many publishers (especially online publishers and self-publishers) now cater for all 3 English exam boards. The digital textbook and animated presentations from www.gcsecs.org will be available for all 3 English exam boards.

Misconceptions

All 3 specifications have misconceptions within them that have not been addressed. Below is a summary of those errors to help you decide on whether you can cope with teaching some aspects of the subject in this way:

AQA misconceptions:

- AQA still use the confusing term "sample resolution" instead of "bit depth". Sample resolution is known to have evolved over time and there is no precise definition, where as "bit depth" is precisely defined.

Note: AQA responded to say that the specification developers are happy with the term "sample resolution".

Edexcel misconceptions:

- Edexcel suggest that count-controlled and condition-controlled loops are NOT known as "iteration" but "repetition" and that "iteration" is reserved for data structures that are iterable. The term "iteration" actually applies to both situations.

Note: Edexcel responded to claim a difference of interpretation over the understanding of the terminology.

OCR misconceptions:

- The term “erroneous” test data has been included as different to “invalid” test data which is not the case. The guidance column makes references to validation being about data types rather than acceptable data.

Note: OCR responded to claim the reference to “erroneous” was to meet compliance with another agency.

Conclusion

Each centre needs to make its own decision about which exam board to work with. If a realistic programming experience using Python is essential then only Edexcel offer this. If your centre doesn’t want to use Python then Edexcel is not an option. The other big advantage for Edexcel is that there is no need to learn a separate exam reference language or exam pseudocode.

OCR allows the widest variety of languages which may suit some centres where less popular languages are taught, but watch out for how complicated it is to work out how each algorithm and programming question across sections A and B should be answered, although OCR’s viewpoint is that they want to make it easier for candidates to respond in a way that they feel happy with to express their logical/accurate solution” which “should be helping weaker candidates”. .

AQA was the most clearly defined of all the specifications and provides stretch and challenge, but watch out for the breadth of coverage. With Edexcel’s Getting Started Guide, their specification becomes even more clearly defined than AQA’s and OCR’s new specification is much better defined than their previous one.

AQA has the largest breadth and depth of coverage which has benefits in terms of differentiation at the top end and preparation for A Level, but does mean there is more content to teach and learn. Edexcel has the least breadth which may be attractive in terms of the amount of content to teach and learn but it may lead to less differentiation between grade boundaries in the examination. OCR appear to have a good balance of breadth and depth with their new specification.

If I was choosing, I’d be looking for a cross between Edexcel’s approach to programming, OCR’s content coverage and AQA’s non-programming exam paper. However, the perfect solution for me isn’t available and won’t be for many, so it’s a case of choosing which aspect is of highest importance and which weaknesses you can work with. I’m tempted by Edxcel because of only having to learn one language and no reference and their responsiveness to feedback. If OCR had taken the on-screen approach with no ERL, then I’d probably choose OCR. I’ve liked AQA’s approach in the past, but the additional content would now put me off as there is more content for students to learn, and the fact they are still using a ‘formal pseudocode’ (equivalent of OCR’s ERL) that students need to be familiar with when their papers are focused on one of 3 languages doesn’t help.

Here’s a brief summary of some of the key issues:

Area	**AQA**	**Edexcel**	**OCR**
On-screen exam	No	Yes	No
Clearly defined specification	Yes	In Getting Started Guide	Yes
Variety of languages	Choose 1 of 3	Python	Any high-level
Need to learn exam reference language	Yes	No	Yes
Length of exams	Longest		Shortest