# The impact of OCR's change from J276 to J277 specifications

## VERSION 3

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.

### Contents

**Feedback on this draft is welcome to paul@gcsecs.org**

## The impact of OCR's change from J276 to J277 specifications
## VERSION 3

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.

### When do I teach each specification?

J276 is for examination in summer 2020 and summer 2021:

- Year 10 who started in September 2019 (unless taking exams at end of year 10)
- Year 11 who started in Year 10 in September 2018

J277 is for examination from summer 2022:

- Year 9 starting in September 2019 (unless taking exams at end of year 10)
- Year 10 starting in September 2020

### What's happening with the Programming Project (former NEA)?

When the original J276 specification was launched, it included an element of 20% assessment by Non-Examined Assessment (NEA). This was cancelled during the first year due to concerns by Ofqual that the authenticity of the assessment couldn't be guaranteed. It was replaced by a Programming Project (PP) which was not assessed, but centres had to show they had offered students the opportunity to spend 20 timetabled classroom hours on the PP which was a board set task by OCR. The two examinations were changed from 40% each to 50% each. Centres were required to submit a sample of student work to OCR for verification that students had completed the project. Centres also had to complete an authentication form and timesheet confirming that all students had been given the opportunity to spend 20 timetabled hours on the PP.

With J277, the PP is being replaced by a Programming Task (PT). The key changes appear to be:

- No analysis
- No evaluation
- No set format except that students must have the opportunity to design, write, test and refine
- No specific time requirement
- No specific scope for the size of the PT
- Assessment of programming skills will be in Component 2 (exam), specifically in Section B
- Centres will need to sign a Practical Programming Statement to confirm all students had an opportunity to undertake a PT

As with the PP, any high-level language can be used.

It’s still an important part of the course because students will be assessed on their ability to design, write, test and refine in Section B of the Component 2 exam which is worth 15% of the whole qualification (30% of Component 1).

**What’s Exam Reference Language?**

OCR, AQA and EdExcel have all provided a version of their own pseudo-code that is used when writing exam questions to ensure consistency in the exams and that students are able to understand the command set. Pseudo-code wasn’t really the right name for this as pseudo-code doesn’t have any formal structure. OCR have therefore changed this to be Exam Reference Language.

The Exam Reference Language, as detailed in the specification, is the language that will be used when presenting code in in exam questions so that all students are able to interpret it, no matter what programming language they have used at school.

In Section A of the Component 2 exam, students can write their own algorithms in answer to questions using any of the following formats:

- Pseudo-code
- Flowcharts
- Bullet points
- OCR Exam Reference Language
- Any high-level programming language
- Natural English

In Section B of the Component 2 exam, if a question asks students to **write** an algorithm/code, the they must use one of the following formats:

- OCR Exam Reference Language
- Any high-level programming language

**Changes from pseudo-code in J276 to reference language in J277**

Apart from the method used to open files, all of the pseudo-code in J276 has been transferred to the J277 Reference Language. There are however, some additions:

| **Programming principle** | **J276 Pseudo-code** | **J277 Reference Language** |
|---|---|---|
| Declaration of variables | No method for declaring constants | **const** is now available<br>e.g.<br>• `const VAT 0.20` |
| Casting | Casting could be done to string (**str**), integer (**int**) or float (**float**) | Casting can now also be done to real (**real**) or Boolean (**bool**).<br>e.g.<br>• `bool("False")`<br>• `real("35.298")`<br>• `float("35.298")`<br>*It's unclear why OCR have chosen to include real in addition to float. A real data type uses 4 bytes and has 7 digits of precision (decimal places). A float data type uses 8 bytes and has 15 digits of precision (decimal places). It's not stated why students would need to use these differently at this level.* |
| Count-controlled loop | The variable in the FOR loop could only increase by a value of 1 (one). | A **step** option has been added so the loop can increase (or decrease) by different amounts.<br>e.g.<br>• `for counter = 0 to 20 step 2`<br>• `for counter = 100 to 0 step -10` |
| Concatenation | Not explicit | Now defined as using + |
| Substrings | Substrings could be found by specifying the starting position and the number of characters. | The ability to select a set number of characters from the **left** or **right** of a string.<br>e.g.<br>• `fullname = "Paul Long"`<br>• `firstname = fullname.left(4)`<br>• `surname = fullname.right(4)` |
| Changing case | There wasn't a method for converting to upper or lower case. | The case of a value can be changed, e.g.<br>• `fullname.upper` gives `"PAUL LONG"`<br>• `firstname.lower` gives `"paul"` |
| ASCII Conversion | There wasn't a method for converting to or from ASCII. | Conversion to or from ASCII can be achieved using:<br>• `ASC(D)` returns `'68'`<br>• `CHR(99)` returns `"c"` |

| **Programming principle** | **J276 Pseudo-code** | **J277 Reference Language** |
|---|---|---|
| Files | Files had to be opened using either **openRead** or **openWrite**.<br>There was no option to create a new file | Files can now be opened using **open** without specifying whether read or write is required.<br>e.g.<br>• `myFile = open("customers.csv")`<br>Files can now be created using newFile.<br>e.g.<br>• `newFile("orders.txt")` |
| Random numbers | There were no random number functions. | There is now a **random** number function. If integers are used within the function, then a random integer between the starting and ending value will be returned. If real numbers are used within the function, then a random real number between the starting and ending value will be returned.<br>e.g.<br>• `random(1,100)`<br>• `random(5.0, 10.0)` |

There is an error in the Exam Reference Language in that there is a discrepancy between DO … WHILE and DO … UNTIL on page 27 of the specification. I suspect that OCR intended to call it a DO UNTIL loop rather than a DO WHILE loop.

**Have any of the exams changed?**

There are still two components worth 50% each:

1. Computer Systems
2. Computational thinking, algorithms and programming

The key changes are:

- Data Representation has been moved from being assessed in Component 2 to Component 1
- Component 2 will include 2 sections, A and B
- Section B will assess learners' ability to design, write, test and refine
- In section B of Component 2, if a question asks students to **write** an algorithm/code, they must use either OCR's Exam Reference Language or any high-level programming language
- Students need to be able to write code accurately in Section B of the Component 2 exam using. While OCR state they won't penalise Low syntax errors, it's unclear as to the extent to which they will penalise. This section will focus on assessing students' ability to design, write, test and refine programs.

**What's changed from the J276 Content?**

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 and forensics
- Full and incremental backups
- Creative Commons Licensing
- Freedom of Information Act 2000
- Input sanitisation
- Planning for contingencies
- Characteristics of an assembler
- Check digits
- Flash memory
- Extended ASCII

Content in section 3 has been removed and replaced with the Programming Task.

The changes are detailed in two tables below.

The first table looks at the changes from the perspective of the new J277 specification so you can see the new structure and how it relates to the old J276 specification you have already been teaching.

The second looks at the changes from the perspective of the old J276 specification so you can adapt your teaching based on what you are already doing with ready for J277.

The spec codes refer to the numbering system used in each specification but each bullet point (round bullets in J276 and square bullets in J277) has been identified with a letter in sequence.

Green text means that something new has been added.

Red text means that something has been removed.

Orange text is used to highlight a change that isn't a clear addition or removal but some sort of other change, for example a clarification or change of terminology.

**Changes in J277 order**

## Component 1: COMPUTER SYSTEMS

### 1.1 System Architecture

| J277 | | | J276 | | What difference does it make? | | |
|---|---|---|---|---|---|---|---|
| **Spec** | **Learning Objective** | **Guidance** | **Spec** | **Learning Objective** | **Impact Summary** | **What you need to do** | **Qualified opinion** |
| 1.1.1a | The purpose of the CPU:<br>o The fetch-execute cycle | What actions occur at each stage of the fetch-execute cycle | 1.1.a | The purpose of the CPU | Low Clarified | Probably nothing as you were covering this across 1.1b and 1.1d | Actions and purpose are two different things so the additional information contradicts the learning objective., but it's likely this was taught anyway under 1.1.b /d |
| | | | 1.1.d | the function of the CPU as fetch and execute instructions stored in memory | | | |
| 1.1.1b | common CPU components and their function:<br>o ALU (Arithmetic Logic Unit)<br>o CU (Control Unit)<br>o Cache<br>o Registers | The role/purpose of each component and what it manages, stores, or controls during the fetch-execute cycle | 1.1.c | common CPU components and their function:<br>o ALU (Arithmetic Logic Unit)<br>o CU (Control Unit)<br>o Cache | Low Clarified | Registers are already covered in 1.1.1c | |
| 1.1.1c | Von Neumann architecture:<br>o MAR (Memory Address Register)<br>o MDR (Memory Data Register)<br>o Program Counter<br>o Accumulator | The purpose of each register, what it stores (data or address)<br>The difference between storing data and an address<br>*No: Knowledge of passing of data between registers in each stage* | 1.1.b | Von Neumann architecture:<br>o MAR (Memory Address Register)<br>o MDR (Memory Data Register)<br>o Program Counter<br>o Accumulator | Medium Clarified | No need to teach about how data passes between registers during FDE cycle | Although not required, it's still a good stretch and challenge activity and helps understand their purpose but learners wont' be tested on passing of data. |
| 1.1.1d | how common characteristics of CPUs affect their performance:<br>o clock speed<br>o cache size<br>o number of cores | Understanding of each characteristic as listed<br>The effects of changing any of the common characteristics on system performance, either individually or in combination | 1.1.e | how common characteristics of CPUs affect their performance:<br>o clock speed<br>o cache size<br>o number of cores | Medium Clarified | Ensure combination of clock speed, cache size and number of cores is also covered. | |
| 1.1.1e | The purpose and characteristics of embedded systems | What embedded systems are<br>Typical characteristics of embedded systems | 1.1.f | embedded systems:<br>o purpose of embedded systems<br>o examples of embedded systems. | Low Added | Characteristics in addition to purpose and examples | It's likely characteristics were taught anyway |
| 1.1.1f | Examples of embedded systems | Familiarity with a range of different embedded systems | | | | | |

## 1.2 Memory and storage

| J277 | | | J276 | | What difference does it make? | | |
|---|---|---|---|---|---|---|---|
| **Spec** | **Learning Objective** | **Guidance** | **Spec** | **Learning Objective** | **Impact Summary** | **What you need to do** | **Qualified opinion** |
| 1.2.1a | The need for primary storage | Why computers have primary storage<br>o How this usually consists of RAM and ROM | - | - | Low Added | Ensure you cover the reasons primary storage are needed | It's likely the need was taught anyway under 1.2.a |
| 1.2.1b | the difference between RAM and ROM | Key characteristics of RAM and ROM | 1.2.a | the difference between RAM and ROM | None | | |
| 1.2.1c | the purpose of ROM in a computer system | | 1.2.b | the purpose of ROM in a computer system | None | | |
| 1.2.1e | the purpose of RAM in a computer system | | 1.2.c | the purpose of RAM in a computer system | None | | |
| 1.2.1e | the need for virtual memory | Why virtual memory may be needed in a system<br>How virtual memory works<br>o Transfer of data between RAM and HDD when RAM is filled | 1.2.d | the need for virtual memory | Low Added | Ensure you cover how virtual memory works as well as the need for it | |
| 1.2.2a | the need for secondary storage | Why computers have secondary storage | 1.3.a | the need for secondary storage | None | | |
| 1.2.2b | common types of storage:<br>o optical<br>o magnetic<br>o solid state | Recognise a range of secondary storage devices/media<br>Differences between each type of storage device/medium<br>*No: Understanding of the component parts of these types of storage* | 1.3.c | common types of storage:<br>o optical<br>o magnetic<br>o solid state | Medium Added<br>Medium Clarified | Ensure you cover the differences between each type of storage device/medium.<br>You don't need to cover component parts like read/write heads.<br>Flash memory from 1.2.e has been removed but is covered as part of solid state storage | Although component parts aren't needed, it can sometimes aid with teaching the differences between the devices and so you may still want to cover it or include it as a stretch and challenge activity. |
| 1.2.2c | suitable storage devices and storage media for a given application | Be able to apply their knowledge in context within scenarios | 1.3.d | suitable storage devices and storage media for a given application, and the advantages and disadvantages of these, using characteristics:<br>o capacity<br>o speed<br>o portability<br>o durability<br>o reliability<br>o cost. | None | | |
| 1.2.2d | the advantages and disadvantages of these, using characteristics:<br>o capacity<br>o speed<br>o portability<br>o durability<br>o reliability<br>o cost. | Compare advantages/disadvantages for each storage device | 1.3.d | | | | |

| J277 | | | J276 | | What difference does it make? | | |
|---|---|---|---|---|---|---|---|
| Spec | Learning Objective | Guidance | Spec | Learning Objective | Impact Summary | What you need to do | Qualified opinion |
| 1.2.3a | The units of data storage:<br>o Bit<br>o Nibble (4 bits)<br>o Byte (8 bits)<br>o Kilobyte (1,000 bytes or 1 KB)<br>o Megabyte (1,000 KB)<br>o Gigabyte (1,000 MB)<br>o Terabyte (1,000 GB)<br>o Petabyte (1,000 TB) | Familiarity with data units and moving between each | 2.6.a | bit, nibble, byte, kilobyte, megabyte, gigabyte, terabyte, petabyte | Low Clarified | Always teach using multiples of 1,000 and use in exams | Stretch and challenge activities could include understanding kibibytes (KiB), mebibytes (MiB) etc in multiples of 1024, but it won't be assessed and 1,000 is easier to use in an exam. |
| 1.2.3b | how data needs to be converted into a binary format to be processed by a computer. | Why data must be stored in binary format | 2.6.b | how data needs to be converted into a binary format to be processed by a computer. | Low Clarified | The 'how' is covered in converting between number bases. The 'why' was already taught in 2.4.a so continue to teach it | There is a contradiction in the spec between 'how' and 'why'. |
| | | | 2.4.a | why data is represented in computer systems in binary form | | | |
| 1.2.3c | data capacity and calculation of data capacity requirements | Calculate capacity of devices<br>Calculate required capacity for a given set of files<br>Calculate file sizes of sound, images and text files<br>o sound file size = sample rate x duration (s) x bit depth<br>o image file size = colour depth x image height (px) x image width (px)<br>o text file size = bits per character x number of characters<br><br>Allowed but not necessary:<br>Use of 1,024 for conversions and calculations would be acceptable<br>Allowance for metadata in calculations may be used | 1.3.b | data capacity and calculation of data capacity requirements | Medium Added<br><br>Medium Clarified | Ensure students can calculate minimum file sizes (excluding metadata) for plain text files, images and sound files. | Metadata is mentioned but it's not necessary and so it's best just to refer to the minimum file size and ignore metadata.<br><br>Although 1024 is allowed, students should stick to 1000 bytes in a kilobyte as 1024 is the number of bytes in a kibibyte and just complicates calculations when a calculator isn't available. |

| J277 | | | J276 | | What difference does it make? | | |
|---|---|---|---|---|---|---|---|
| Spec | Learning Objective | Guidance | Spec | Learning Objective | Impact Summary | What you need to do | Qualified opinion |
| 1.2.4a | How to convert positive denary whole numbers to binary numbers (up to and including 8 bits) and vice versa | Denary number range 0 – 255<br>Binary number range 00000000 – 11111111 | 2.6.c | how to convert positive denary whole numbers ~~(0–255)~~ into ~~8 bit~~ binary numbers and vice versa | Low<br>Clarified | The maximum number of bits was already 8 due to the limit being 0-255 but the "up to" bit means students don't need leading zeros in their binary answers unless asked for an 8-bit format. | |
| 1.2.4b | how to add two binary integers together (up to and including 8 bits) and explain overflow errors which may occur | Understanding of the terms most significant bit, and least significant bit<br>Ability to deal with binary numbers containing between 1 and 8 bits<br>o e.g. 11010 is the same as 00011010 | 2.6.d | how to add ~~two 8 bit~~ binary integers and explain overflow errors which may occur | Medium<br>Clarified | Exams may ask students to add two binary integers that are less than 8 bits each.<br><br>Students need to be familiar with MSB and LSB. | LSB and MSB may have been terms used when explaining overflow, but students now need to know these terms for the exam. |
| 1.2.4c | How to convert positive denary whole numbers into 2-digit hexadecimal numbers and vice versa | Denary number range 0 – 255<br>Hexadecimal range 00 – FF | 2.6.f | how to convert positive denary whole numbers ~~(0–255)~~ into 2 digit hexadecimal numbers and vice versa | None | | |
| 1.2.4d | How to convert binary integers to their hexadecimal equivalents and vice versa | Hexadecimal range 00 – FF<br>Binary number range 00000000 – 11111111 | 2.6.g | how to convert from binary to hexadecimal equivalents and vice versa | Low<br>SPAG | Nothing | |
| 1.2.4e | binary shifts | Understand the effect of a binary shift (both left or right) on a number | 2.6.e | binary shifts | Low<br>Clarified | This suggests students don't need to be able to carry out a binary shift. | To understand the effect of a binary shift, the best way to explain it is to show how it's done so I would teach this anyway. |
| 1.2.4f | the use of binary codes to represent characters | How characters are represented in binary | 2.6.i | the use of binary codes to represent characters | None | | |
| 1.2.4g | the term 'character-set' | Understand how character sets are logically ordered, e.g. the code for 'B' will be one more than the code for 'A' | 2.6.j | the term 'character-set' | Low<br>Added | Ensure you teach that if one character's code is known, it's often possible to work out another one using sequence. | Most textbooks covered this anyway. |

| J277 | | | J276 | | What difference does it make? | | |
|---|---|---|---|---|---|---|---|
| **Spec** | **Learning Objective** | **Guidance** | **Spec** | **Learning Objective** | **Impact Summary** | **What you need to do** | **Qualified opinion** |
| 1.2.4h | the relationship between the number of bits per character in a character set, and the number of characters which can be represented, e.g.:<br>o ASCII<br>o Unicode | How the number of characters stored is limited by the bits available<br>The differences between and impact of each character set<br>Binary representation of ASCII in the exam will use 8 bits<br>*No: Memorisation of character set codes* | 2.6.k | the relationship between the number of bits per character in a character set and the number of characters which can be represented (for example ASCII, ~~extended ASCII~~ and Unicode). | Medium Clarified | Ensure you cover binary representation of ASCII as well as decimal representation.<br>OCR won't use the term "extended ASCII" in an exam but students can make reference to it in an answer. | This has added a slightly extra layer of complexity but helps in the understanding that the codes are actually binary and so decimal is just an intermediary step for the ASCII code. |
| 1.2.4i | how an image is represented as a series of pixels, represented in binary | Each pixel has a specific colour, represented by a specific code | 2.6.l | how an image is represented as a series of pixels represented in binary | None | | |
| 1.2.4j | metadata | Metadata stores additional image information (e.g. height, width, etc.) | 2.6.m | metadata ~~included in the file~~ | Low Clarified | No action required | |
| 1.2.4k | the effect of colour depth and resolution on:<br>o the quality of the image<br>o the size of an image file | The effect on image size and quality when changing colour depth and resolution | 2.6.n | the effect of colour depth and resolution on the size of an image file. | Medium Added | Ensure you cover the effect on quality (e.g. pixilation, loss of colours) as well as size. | This would probably have been taught anyway. Do be careful though that increasing colour depth doesn't increase quality unless additional colours are also added. Similarly, increasing resolution doesn't increase quality unless smoothing takes place. |
| 1.2.4l | how sound can be sampled and stored in digital form | Analogue sounds must be stored in binary | 2.6.o | how sound can be sampled and stored in digital form | None | | |
| 1.2.4m | The effect of sample rate, duration and bit depth on:<br>o The playback quality<br>o The size of a sound file | Sample rate – measured in Hertz (Hz)<br>Duration – how many seconds of audio the sound file contains<br>Bit depth – number of bits available to store each sample (e.g. 16-bit) | 2.6.p | how ~~sampling intervals and other factors~~ affect the size of a sound file and the quality of its playback:<br>~~o sample size~~<br>~~o bit rate~~<br>~~o sampling frequency.~~ | Medium Terms | Use the term "sample rate" instead of "sampling frequency" and "bit depth" instead of "sample size"<br>No need to cover "bit rate" | OCR are now using the correct industry standard terminology. Removing "bit rate" avoids confusion with "bit depth". |
| 2.2.4n | need for compression | Common scenarios where compression may be needed | 2.6.q | need for compression | None | | |

| J277 | | | J276 | | What difference does it make? | | |
|---|---|---|---|---|---|---|---|
| Spec | Learning Objective | Guidance | Spec | Learning Objective | Impact Summary | What you need to do | Qualified opinion |
| 1.2.4o | types of compression:<br>o lossy<br>o lossless. | Advantages and disadvantages of each type of compression<br>Effects on the file for each type of compression<br>*No: Ability to carry out specific compression algorithms* | 2.6.r | types of compression:<br>o lossy<br>o lossless. | Medium Clarified | Ensure you cover advantages and disadvantages of each as well as the effect on the file (e.g. lossy loses data).<br>No need to teach RLE. | These things were covered in most textbooks anyway, but it's now clear exactly what OCR could ask questions about.<br>Although students don't need to carry out an algorithm like RLE, using RLE as an example helps to explain a lossless technique and is also a useful stretch and challenge activity. |

## 1.3 Computer networks, connections and protocols

| J277 | | | J276 | | What difference does it make? | | |
|---|---|---|---|---|---|---|---|
| **Spec** | **Learning Objective** | **Guidance** | **Spec** | **Learning Objective** | **Impact Summary** | **What you need to do** | **Qualified opinion** |
| 1.3.1a | types of networks:<br>o LAN (Local Area Network)<br>o WAN (Wide Area Network) | The characteristics of LANs and WANs including common examples of each<br>Apply understanding of networks to a given scenario | 1.4.a | types of networks:<br>o LAN (Local Area Network)<br>o WAN (Wide Area Network) | Low<br>Clarified<br><br>High<br>Removed | Include examples of common LANs and WANs<br><br>You no longer need to cover virtual networks (old 1.4.f). | Most teaching would have included this already. |
| 1.3.1b | factors that affect the performance of networks | Understanding of different factors that can affect the performance of a network, e.g.:<br>o Number of devices connected<br>o Bandwidth | 1.4.b | factors that affect the performance of networks | Low<br>Clarified | Ensure at least these 2 examples are covered. Students can use other answers in an exam. | |
| 1.3.1c | the different roles of computers in a client-server and a peer-to-peer network | Concept of clients requesting/using services from a server | 1.4.c | the different roles of computers in a client-server and a peer-to-peer network | None | | |
| 1.3.1d | the hardware needed to connect stand-alone computers into a Local Area Network:<br>o wireless access points<br>o routers<br>o switches<br>o NIC (Network Interface Controller/Card)<br>o transmission media | The tasks performed by each piece of hardware | 1.4.d | the hardware needed to connect stand-alone computers into a Local Area Network:<br>o wireless access points<br>o routers/switches<br>o NIC (Network Interface Controller/Card)<br>o transmission media | Low<br>SPAG | Teach routers and switches as two distinctly separate components. | These are 2 separate components with different functions so it's right they are separated. |
| 1.3.1e | the internet as a worldwide collection of computer networks:<br>o DNS (Domain Name Server)<br>o hosting<br>o the cloud<br>o web servers and clients | The concept of the Internet as a network of computer networks<br>A DNS's role in the conversion of a URL to an IP address<br>Concept of servers providing services (e.g. Web server = Web pages, File server = file storage/retrieval)<br>The Cloud: remote service provision (e.g. storage, software, processing)<br>Advantages and disadvantages of the Cloud | 1.4.e | the internet as a worldwide collection of computer networks:<br>o DNS (Domain Name Server)<br>o hosting<br>o the cloud | Medium<br>Clarified | DNS is clarified that you only need to teach about its role of converting URL to an IP address using a database.<br><br>For hosting, ensure you cover at least web hosting and file hosting including web servers and clients.<br><br>You need to cover advantages and disadvantages of the Cloud | Most centres probably previously focused on web hosting rather than other types of hosting previously so this helps to clarify.<br><br>Although many textbooks would have covered advantages and disadvantages of the Cloud, it wasn't previously specified. |
| 1.3.1f | star and mesh network topologies | Advantages and disadvantages of the Star and Mesh topologies | 1.5.a | star and mesh network topologies | Medium<br>Clarified | You need to include the advantages and disadvantages. | Many centres will have taught this anyway, but it's now clear that it is needed. |

| J277 | | | J276 | | What difference does it make? | | |
|---|---|---|---|---|---|---|---|
| Spec | Learning Objective | Guidance | Spec | Learning Objective | Impact Summary | What you need to do | Qualified opinion |
| 1.3.2a | Modes of connection:<br>o Wired<br>• Ethernet<br>o Wireless<br>• Wi-Fi<br>• Bluetooth | Compare benefits and drawbacks of wired versus wireless connection<br>Recommend one or more connections for a given scenario<br>*No: Understand how Ethernet, Wi-Fi and Bluetooth protocols work* | 1.5.b | Wifi:<br>~~o frequency and channels~~<br>o encryption | High Added<br>High Removed<br>Low Clarified | Benefits and drawbacks of wired and wireless need covering.<br>Students need to be able to justify wired or Wi-Fi or Bluetooth (or combination of) based on a scenario.<br>Need to cover Bluetooth.<br>No need to cover frequency and channels for Wi-Fi.<br>No need to teach Ethernet, Wi-Fi or Bluetooth protocols. | Ethernet was an outlier before in 1.5.c and it was never clear what to teach. This makes it much clearer. Knowing that the protocols aren't needed simplifies the teaching of this topic.<br>Frequency and channels was a challenging topic that has now been removed.<br>Although encryption has been removed from Wi-Fi, it is still covered in a wider sense in 1.3.2b which isn't just restricted to Wi-Fi. |
| | | | 1.5.c | ethernet | | | |
| 1.3.2b | Encryption | The principle of encryption to secure data across network connections | 1.5.b | Wifi:<br>o frequency and channels<br>o encryption | Low Moved | Encryption isn't now restricted to Wi-Fi encryption so cover it in a wider sense. | Most centres would be covering the purpose of encryption anyway and making reference to it in HTTPS so impact should be low. |
| 1.3.2c | IP addressing and MAC addressing | IP addressing and the format of an IP address (IPv4 and IPv6)<br>A MAC address is assigned to devices; its use within a network<br>*No: Understand differences between static and dynamic, or public and private IP addresses* | 1.5.d | the uses of IP addressing, MAC addressing, and protocols including:<br>o TCP/IP (Transmission Control Protocol/Internet Protocol)<br>o HTTP (Hyper Text Transfer Protocol)<br>o HTTPS (Hyper Text Transfer Protocol Secure)<br>o FTP (File Transfer Protocol)<br>o POP (Post Office Protocol)<br>o IMAP (Internet Message Access Protocol)<br>o SMTP (Simple Mail Transfer Protocol) | Low Clarified | Ensure format of IPv6 is covered as well as IPv4.<br>No longer need to teach static, dynamic, public or private IP addresses.<br>See also 1.3.2.e | Some textbooks already covered IPv6.<br>Although the differences between types of IP addresses isn't needed for the exam, it's a useful stretch and challenge activity. |

| J277 | | | J276 | | What difference does it make? | | |
|---|---|---|---|---|---|---|---|
| Spec | Learning Objective | Guidance | Spec | Learning Objective | Impact Summary | What you need to do | Qualified opinion |
| 1.3.2d | Network standards | The principle of a standard to provide rules for areas of computing<br>Standards allows hardware/software to interact across different manufacturers/producers<br>*No: Knowledge of individual standards* | - | - | High<br>Added | This is a new topic that needs covering. | |
| 1.3.2e | Common protocols including:<br>o TCP/IP (Transmission Control Protocol/Internet Protocol)<br>o HTTP (Hyper Text Transfer Protocol)<br>o HTTPS (Hyper Text Transfer Protocol Secure)<br>o FTP (File Transfer Protocol)<br>o POP (Post Office Protocol)<br>o IMAP (Internet Message Access Protocol)<br>o SMTP (Simple Mail Transfer Protocol) | The principle of a (communication) protocol as a set of rules for transferring data<br>That different types of protocols are used for different purposes<br>The basic principles of each protocol i.e. its purpose and key features | 1.5.d | the uses of IP addressing, MAC addressing, and protocols including:<br>o TCP/IP (Transmission Control Protocol/Internet Protocol)<br>o HTTP (Hyper Text Transfer Protocol)<br>o HTTPS (Hyper Text Transfer Protocol Secure)<br>o FTP (File Transfer Protocol)<br>o POP (Post Office Protocol)<br>o IMAP (Internet Message Access Protocol)<br>o SMTP (Simple Mail Transfer Protocol) | High<br>Removed | No longer need to cover the old 1.5.f packet switching<br>See also 1.3.2.c | |
| 1.3.2f | the concept of layers | How layers are used in protocols, and the benefits of using layers;<br>for a teaching example, please refer to the 4-layer TCP/IP model<br>*No: Knowledge of the names and function of each TCP/IP layer* | 1.5.e | the concept of layers | Low<br>Added<br>Low<br>Clarified | Ensure you include the benefits of using layers.<br>Students no longer need to know the names and function of each TCP/IP layer. | The benefits were probably covered in most teaching anyway.<br>Although names and function of TCP/IP layer aren't needed, it's a useful teaching tool, but students won't be tested on it in the exam. They can use it as an example in their answers though. |

## 1.4 Network security

| J277 | | | J276 | | What difference does it make? | | |
|---|---|---|---|---|---|---|---|
| **Spec** | **Learning Objective** | **Guidance** | **Spec** | **Learning Objective** | **Impact Summary** | **What you need to do** | **Qualified opinion** |
| 1.4.1a | Forms of attack:<br>o Malware<br>o Social engineering, e.g. phishing, people as the 'weak point'<br>o Brute-force attacks<br>o Denial of service attacks<br>o Data interception and theft<br>o The concept of SQL injection | Threats posed to devices/systems<br>Knowledge/principles of each form of attack including:<br>o How the attack is used<br>o The purpose of the attack | 1.6.a | forms of attack | Medium Added<br>Medium Removed | You may want to teach about other social engineering techniques like pharming and shouldering but these won't be specifically tested in an exam.<br>For each attack, ensure you cover its purpose (what it's intended to do) and how it is used<br>You no longer need to cover poor network policy. | |
| | | | 1.6.b | threats posed to networks:<br>o malware<br>o phishing<br>o people as the 'weak point' in secure systems (social engineering)<br>o brute force attacks<br>o denial of service attacks<br>o data interception and theft<br>o the concept of SQL injection<br>~~o poor network policy~~ | | | |
| 1.4.2a | Common prevention methods:<br>o Penetration testing<br>o Anti-malware software<br>o Firewalls<br>o User access levels<br>o Passwords<br>o Encryption<br>o Physical security | Understanding of how to limit the threats posed in 1.4.1<br>Understanding of methods to remove vulnerabilities<br>Knowledge/principles of each prevention method:<br>o What each prevention method may limit/prevent<br>o How it limits the attack | 1.6.c | identifying and preventing vulnerabilities:<br>o penetration testing<br>~~o network forensics~~<br>~~o network policies~~<br>o anti-malware software<br>o firewalls<br>o user access levels<br>o passwords<br>o encryption. | Medium Added<br>Medium Removed | You need to include physical security in your teaching.<br>You no longer need to cover network forensics or network policies. | |

## 1.5 Systems software

| J277 | | | J276 | | What difference does it make? | | |
|---|---|---|---|---|---|---|---|
| Spec | Learning Objective | Guidance | Spec | Learning Objective | Impact Summary | What you need to do | Qualified opinion |
| 1.5.1a | The purpose and functionality of operating systems:<br>o user interface<br>o memory management and multitasking<br>o peripheral management and drivers<br>o user management<br>o file management | What each function of an operating system does<br>o Features of a user interface<br>o Memory management, e.g. the transfer of data between memory, and how this allows for multitasking<br>Understand that:<br>o Data is transferred between devices and the processor<br>o This process needs to be managed and what this entails (e.g. the use of buffers when transferring data to a printer)<br>User management functions, e.g.:<br>o Allocation of an account<br>o Access rights<br>o Security, etc.<br>File management, and the key features, e.g.:<br>o Naming<br>o Allocating to folders<br>o Moving files<br>o Saving, etc.<br>*No: Understanding of paging or segmentation* | 1.7.b | operating systems:<br>o user interface<br>o memory management/multitasking<br>o peripheral management and drivers<br>o user management<br>o file management | High<br>Clarified | Students need to understand how memory management enables multitasking.<br>Students need to be aware of data transfer between peripherals and the processor including the use of buffers.<br>Some examples of user and file management are given but students can use other examples in their answers.<br>There's no need to cover paging or segmentation. | There's quite a lot added here and a bit unclear on the depth required such as buffering and the transfer of data between devices and the processor. |
| 1.5.2a | the purpose and functionality of utility software | Understand that computers often come with utility software, and how this performs housekeeping tasks | 1.7.a | the purpose and functionality of ~~systems~~ software | Low<br>Changed<br>Medium<br>Removed | Cover what you will teach in 1.5.1b but also cover the purpose of utility software in general.<br>It's no longer necessary to cover systems software as the set of utility software and operating system, just utility software. | Personally, I'd have included this as the introduction to 1.5.1b<br>There's no harm in mentioning to students that system software is the set that includes utility software and operating systems |
| 1.5.2b | utility system software:<br>o encryption software<br>o defragmentation<br>o data compression | Purpose of the identified utility software and why it is required | 1.7.c | utility system software:<br>o encryption software<br>o defragmentation<br>o data compression<br>~~o the role and methods of backup:~~<br>~~full~~<br>~~incremental.~~ | Medium<br>Removed | No longer need to teach about full and incremental backups. | Still remind students regularly about the importance of backing up their own work. |

## 1.6 Ethical, legal, cultural and environmental impacts of digital technology

| J277 | | | J276 | | What difference does it make? | | |
|---|---|---|---|---|---|---|---|
| **Spec** | **Learning Objective** | **Guidance** | **Spec** | **Learning Objective** | **Impact Summary** | **What you need to do** | **Qualified opinion** |
| 1.6.1a | Impacts of digital technology on wider society including:<br>o Ethical issues<br>o Legal issues<br>o Cultural issues<br>o Environmental issues<br>o Privacy issues | Technology introduces ethical, legal, cultural, environmental and privacy issues<br>Knowledge of a variety of examples of digital technology and how this impacts on society<br>An ability to discuss the impact of technology based around the issues listed | 1.8.a | how to investigate and discuss Computer Science technologies while considering:<br>o ethical issues<br>o legal issues<br>o cultural issues<br>o environmental issues.<br>o privacy issues. | Low Clarified | Students need to understand a variety of examples of digital technology and their impact. This can be achieved through regular wider reading such as the articles Tweeted at @paullongnet | This has tidied up the whole area of impacts because it was spread across several bullet points. The old 1.8.a/b/c/d have been merged into a single learning objective. |
| 1.6.1b | Legislation relevant to Computer Science:<br>o The Data Protection Act 2018<br>o Computer Misuse Act 1990<br>o Copyright Designs and Patents Act 1988<br>o Software licences (i.e. open source and proprietary) | The purpose of each piece of legislation and the specific actions it allows or prohibits<br>The need to license software and the purpose of a software licence<br>Features of open source (providing access to the source code and the ability to change the software)<br>Features of proprietary (no access to the source code, purchased commonly as off-the-shelf)<br>Recommend a type of licence for a given scenario including benefits and drawbacks | 1.8.f | legislation relevant to Computer Science:<br>o The Data Protection Act ~~1998~~<br>o Computer Misuse Act 1990<br>o Copyright Designs and Patents Act 1988<br>~~o Creative Commons Licensing~~<br>~~o Freedom of Information Act 2000.~~ | High Removed | No longer need to teach about Creative Commons Licensing or Freedom of Information Act<br><br>Ensure new aspects introduced to the DPA in 2018 such as right to be forgotten and implicit consent are included..<br><br>Open source and proprietary were already covered in 1.8.e but you don't need to teach about other types of software licences. | The use of i.e. instead of e.g. makes it clear that OCR don't need students to know about single-user, multi-user, site licences etc. |
| | | | 1.8.e | open source vs proprietary software | High Updated | | |

## Component 2: COMPUTATIONAL THINKING, ALGORITHMS AND PROGRAMMING

### 2.1 Algorithms

| J277 | | | J276 | | What difference does it make? | | |
|---|---|---|---|---|---|---|---|
| **Spec** | **Learning Objective** | **Guidance** | **Spec** | **Learning Objective** | **Impact Summary** | **What you need to do** | **Qualified opinion** |
| 2.1.1a | Principals of computational thinking:<br>o abstraction<br>o decomposition<br>o algorithmic thinking | Understanding of these principles and how they are used to define and refine problems | 2.1.a | computational thinking:<br>o abstraction<br>o decomposition<br>o algorithmic thinking | None | | |
| 2.1.2a | Identify the inputs, processes, and outputs for a problem | | - | - | Medium Added | Give students algorithms or problems and ask them to identify the inputs, processes and outputs separately. | This is covered in AQA and is quite straightforward. |
| 2.1.2b | Structure diagrams | Produce simple diagrams to show:<br>o The structure of a problem<br>o Subsections and their links to other subsections | - | - | Low Added | Adapt the teaching of decomposition to ensure students are able to create structure diagrams in an exam environment. | This is what students should be doing during decomposition anyway and can have practical experience of with the design part of the Programming Activity. |
| 2.1.2c | create, interpret, correct, complete, and refine algorithms using:<br>o pseudocode<br>o flowcharts<br>o reference language / high-level programming language | Complete, write or refine an algorithm using the techniques listed<br>*See Flowchart Symbols on page 15 of specification* | 2.1.d | how to produce algorithms using:<br>o pseudocode<br>o using flow ~~diagrams~~ | High Added<br>Low SPAG | Students need to use the flowchart symbols on page 15 of the specification.<br>Students will need to be able to refine algorithms as well as create, interpret, correct and complete them.<br>In Section B of the Component 2 exam, students will need to use either OCR's Exam Reference Language or a high-level language.<br>The term flowcharts is used instead of flow diagrams. | There will be a lot of opinions about whether students should use a language or pseudo-code. Section B is 15% of the whole qualification and there will be marks available for structure, logic and syntax. What is unknown is how forgiving examiners will be with syntax (see page 24 of the spec).<br>Refine has been added to enable assessment of the refine process in the Programming Task.<br>The term flowcharts is used correctly in the new spec. |
| | | | 2.1.e | interpret, correct or complete algorithms. | | | |

| J277 | | | J276 | | What difference does it make? | | |
|---|---|---|---|---|---|---|---|
| **Spec** | **Learning Objective** | **Guidance** | **Spec** | **Learning Objective** | **Impact Summary** | **What you need to do** | **Qualified opinion** |
| 2.1.2d | Identify common errors | Identify syntax/logic errors in code and suggest fixes | 2.3.e | how to identify syntax and logic errors | Low Added | Students now need to be able to suggest fixes to errors.<br>*Note: also in 2.3.2c* | |
| 2.1.2e | Trace tables | Create and use trace tables to follow an algorithm | 2.1.e | interpret, correct or complete algorithms. | High Added | Students need to be able to create and use trace tables. | Some teachers thought this was implied by being able to interpret algorithms or would have taught it as a skill anyway, but now it is mandatory. |
| 2.1.3a | standard searching algorithms:<br>o binary search<br>o linear search | Understand the main steps of each algorithm<br>Understand any pre-requisites of an algorithm<br>Apply the algorithm to a data set<br>Identify an algorithm if given the code for it<br>*No: To remember the code for these algorithms* | 2.1.b | standard searching algorithms:<br>o binary search<br>o linear search | Low Clarified | Students don't need to be able to rewrite the code for these algorithms but if shown the code, they need to be able to identify the algorithm by name and its purpose. | This is what most teachers have expected has been the case with J276 but now it is specified. |
| 2.1.3b | standard sorting algorithms:<br>o bubble sort<br>o merge sort<br>o insertion sort | | 2.1.c | standard sorting algorithms:<br>o bubble sort<br>o merge sort<br>o insertion sort | | | |

## 2.2 Programming fundamentals

| J277 | | | J276 | | What difference does it make? | | |
|---|---|---|---|---|---|---|---|
| Spec | Learning Objective | Guidance | Spec | Learning Objective | Impact Summary | What you need to do | Qualified opinion |
| 2.2.1a | the use of variables, constants, operators, inputs, outputs and assignments | Practical use of the techniques in a high-level language within the classroom | 2.2.a | the use of variables, constants, operators, inputs, outputs and assignments | None | | |
| 2.2.1b | the use of the three basic programming constructs used to control the flow of a program:<br>o sequence<br>o selection<br>o iteration (count- and condition-controlled loops) | == Equal to<br>!= Not equal to<br>< Less than<br><= Less than or equal to<br>> Greater than<br>>= Greater than or equal to | 2.2.b | the use of the three basic programming constructs used to control the flow of a program:<br>o sequence<br>o selection<br>o iteration (count and condition controlled loops) | Terms | Just a hyphen added to count-controlled and condition-controlled loops | |
| 2.2.1c | the common arithmetic operators | + Addition<br>– Subtraction<br>* Multiplication<br>/ Division<br>MOD Modulus<br>DIV Quotient<br>^ Exponentiation (to the power) | 2.2.j | the common arithmetic operators | None | | |
| | | | 2.4f | the common arithmetic operators"applying computing-related mathematics:<br>o +<br>o –<br>o /<br>o *<br>o Exponentiation (^)<br>o MOD<br>o DIV" | | | |
| 2.2.1d | the common Boolean operators AND, OR and NOT | | 2.2.k | the common Boolean operators. | Low Clarified | You only need to teach AND, OR and NOT. | Most teachers were doing this anyway but it helps to clarify. |
| 2.2.2a | the use of data types:<br>o integer<br>o real<br>o Boolean<br>o character and string<br>o casting | Practical use of the data types in a high-level language within the classroom<br>Ability to choose suitable data types for data in a given scenario<br>Understand that data types may be temporarily changed through casting, and where this may be useful | 2.2.i | the use of data types:<br>o integer<br>o real<br>o Boolean<br>o character and string<br>o casting | None | | |

| J277 | | | J276 | | What difference does it make? | | |
|---|---|---|---|---|---|---|---|
| Spec | Learning Objective | Guidance | Spec | Learning Objective | Impact Summary | What you need to do | Qualified opinion |
| 2.2.3a | the use of basic string manipulation | Ability to manipulate strings, including:<br>o Concatenation<br>o Slicing | 2.2.c | the use of basic string manipulation | Low Clarified | String manipulation must include concatenation and slicing. | This would have been standard practice at schools but it now narrows down what aspects of string manipulation are likely to be covered. |
| 2.2.3b | the use of basic file handling operations:<br>o open<br>o read<br>o write<br>o close | | 2.2.d | the use of basic file handling operations:<br>o open<br>o read<br>o write<br>o close | None | | |
| 2.2.3c | the use of records to store data | | 2.2.e | the use of records to store data | None | | |
| 2.2.3d | the use of SQL to search for data | SQL commands:<br>o SELECT<br>o FROM<br>o WHERE | 2.2.f | the use of SQL to search for data | Low Clarified | The only SQL commands to teach are SELECT, FROM, WHERE | This clarifies that INNER or OUTER JOINS aren't required or sorting into order.<br>OCR have stated they have removed LIKE, AND, OR and wildcards although this is not clear in the specification. |
| 2.2.3e | the use of arrays (or equivalent) when solving problems, including both one-dimensional and two-dimensional arrays | Arrays as fixed length static structures | 2.2.g | the use of arrays (or equivalent) when solving problems, including both one and two dimensional arrays | Low SPAG | No action required | |
| 2.2.3f | how to use sub programs (functions and procedures) to produce structured code | The use of functions<br>The use of procedures<br>Where to use functions and procedures effectively | 2.2.h | how to use sub programs (functions and procedures) to produce structured code | None | | |
| 2.2.3g | Random number generation | | | | High Added | Include the use of random numbers in programming tasks so that students get an experience of using them. | Would be nice to have some additional information. |

## 2.3 Producing robust programs

| J277 | | | J276 | | What difference does it make? | | |
|---|---|---|---|---|---|---|---|
| **Spec** | **Learning Objective** | **Guidance** | **Spec** | **Learning Objective** | **Impact Summary** | **What you need to do** | **Qualified opinion** |
| 2.3.1a | defensive design considerations:<br>o anticipating misuse<br>o authentication | Understanding of the issues a programmer should consider to ensure that a program caters for all likely input values<br>Authentication to confirm the identity of a user<br>Practical experience of designing simple authentication (e.g. username and password) | 2.3.a | defensive design considerations:<br>~~o input sanitisation~~/validation<br>~~o planning for contingencies~~<br>o anticipating misuse<br>o authentication | Low Moved<br>Low Removed | No longer need to cover input sanitisation or planning for contingencies.<br>Validation has moved to 2.3.1b<br>Students must have experience of designing username and password authentication routines | This suggests exam questions would require students to be able to design and maybe even write a simple username and password routine. |
| 2.3.1b | Input validation | Understanding of how to deal with invalid data in a program<br>Practical experience of designing input validation | 2.3.a | | Low Moved<br>Low Clarified<br>High Removed | Input validation was covered in 2.3.a previously so it's still the same teaching process.<br>Students must have experience of designing validation routines | This suggests exam questions would require students to be able to design and maybe even write a simple validation routine. |
| 2.3.1c | Maintainability:<br>o Use of sub programs<br>o Naming conventions<br>o Indentation<br>o Commenting | Understand why commenting is useful and apply this appropriately | 2.3.b | maintainability:<br>o comments<br>o indentation | Low Added | Students need to understand how sub programs and naming conventions help make programs maintainable. | |

| J277 | | | J276 | | What difference does it make? | | |
|---|---|---|---|---|---|---|---|
| Spec | Learning Objective | Guidance | Spec | Learning Objective | Impact Summary | What you need to do | Qualified opinion |
| 2.3.2a | the purpose of testing | | 2.3.c | the purpose of testing | None | | |
| 2.3.2b | types of testing:<br>o iterative<br>o final/terminal | The difference between testing modules of a program during development and testing the program at the end of production | 2.3.d | types of testing:<br>o iterative<br>o final/terminal | Low<br>Clarified | As well as understanding iterative and final/terminal testing, students need to understand the difference between these with iterative focused on testing modules during development and final focused on testing the final program. | It's likely this was taught by most teachers already. |
| 2.3.2c | identify syntax and logic errors | Syntax errors as errors which break the grammatical rules of the programming language and stop it from being run/translated<br>Logic errors as errors which produce unexpected output | 2.3.e | how to identify syntax and logic errors | Low<br>SPAG | No action required.<br>*Note: this is also in 2.1.2d* | |
| 2.3.2d | selecting and using suitable test data:<br>o Normal<br>o Boundary<br>o Invalid<br>o Erroneous | Normal test data as data which should be accepted by a program without causing errors<br>Boundary test data as data of the correct type which is on the very edge of being valid<br>Invalid test data as data of the correct type but outside accepted validation limit<br>Erroneous test data as data of the incorrect type which should be rejected by a computer system<br>Ability to identify suitable test data for a given scenario<br>Ability to create/complete a test plan | 2.3.f | selecting and using suitable test data. | Low<br>Clarified | Students don't need to know about invalid boundary data, just valid boundary data.<br>The definition used for erroneous data here is wrong. Erroneous data is actually the same as Invalid data. If OCR don't update the specification, you'll have to teach it and make sure students are clear they have to use terms incorrectly, but how OCR want them, in the exam. | I've asked OCR if they will remove the term erroneous and its incorrect definition. The only mention I can find of them being different is on BBC Bytesize which is clearly wrong. There are no other sources that suggest they are different and plenty of reliable sources that identify them as being the same. |
| 2.3.2e | Refining algorithms | | - | - | Medium<br>Added | Through the programming task and other tasks, give students experience of refining algorithms so they can refine an unfamiliar algorithm in an exam. | |

## 2.4 Boolean logic

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 2.4.1a | simple logic diagrams using the operators AND, OR and NOT | Recognition of each gate symbol<br>Understanding of how to create, complete or edit logic diagrams and truth tables for given scenarios | 2.4.b | simple logic diagrams using the operations AND, OR and NOT | Low<br>Clarified | In order to achieve 2.4.c/d/e it's likely all of this is already covered.<br><br>*See page 20 of specification for logic gate symbols* | |
| 2.4.1b | truth tables | Knowledge of the truth tables for each logic gate | 2.4.c | truth tables | Low<br>Clarified | Carry on teaching the NOT, AND, OR gate truth tables. | |
| 2.4.1c | combining Boolean operators using AND, OR and NOT | Ability to work with more than one gate in a logic diagram | 2.4.d | combining Boolean operators using AND, OR and NOT ~~to two levels~~ | High<br>Added | You will need to teach students how to create more complex diagrams. Use this as an opportunity for differentiation and stretch and challenge. | There is now no limit on how complex a problem could be combining Boolean operators. |
| 2.4.1d | applying logical operators in truth tables to solve problems | Understanding of how to create, complete or edit logic diagrams and truth tables for given scenarios<br><br>***Alternatives***<br>*Use of other valid notation will be accepted within the examination, e.g. Using T/F for 1/0, or V for OR, etc.* | 2.4.e | applying logical operators in ~~appropriate~~ truth tables to solve problems | Low<br>SPAG | No action required | |

## 2.5 Programming languages and Integrated Development Environments

| J277 | | | J276 | | What difference does it make? | | |
|---|---|---|---|---|---|---|---|
| Spec | Learning Objective | Guidance | Spec | Learning Objective | Impact Summary | What you need to do | Qualified opinion |
| 2.5.1a | Characteristics and purpose of different levels of programming language:<br>o High-level languages<br>o Low-level languages | The differences between high- and low-level programming languages | 2.5.a | characteristics and purpose of different levels of programming language, ~~including~~ low level languages | Low<br>Clarified | It's most likely you were already including high-level languages so probably no action required.<br>If you weren't already including the differences then cover these explicitly. | |
| 2.5.1b | the purpose of translators | The need for translators | 2.5.b | the purpose of translators | None | | |
| 2.5.1c | the characteristics of a compiler and an interpreter | The differences, benefits and drawbacks of using a compiler or an interpreter<br>*No: Understanding of assemblers* | 2.5.c | the characteristics of an ~~assembler~~, a compiler and an interpreter | Low<br>Added<br>High<br>Deleted | Differences, benefits and drawbacks between compiler and assembler need to be covered.<br>Assemblers can be removed from your teaching. | Most teachers were probably covering differences, benefits and drawbacks as a natural part of explaining the characteristics so shouldn't be too difficult to implement. |
| 2.5.2a | common tools and facilities available in an integrated development environment (IDE):<br>o editors<br>o error diagnostics<br>o run-time environment<br>o translators. | Knowledge of the tools that an IDE provides<br>How each of the tools and facilities listed can be used to help a programmer develop a program<br>Practical experience of using a range of these tools within at least one IDE | 2.5.d | common tools and facilities available in an integrated development environment (IDE):<br>o editors<br>o error diagnostics<br>o run-time environment<br>o translators. | Low<br>Clarification | As well as the characteristics of these features, consider how they help a programmer to develop a program. | This is probably common sense but worth clarifying. |

**Changes in J276 order**

## Component 1: COMPUTER SYSTEMS

### 1.1 Systems architecture

| J276 | | J277 | | | What difference does it make? | | |
|---|---|---|---|---|---|---|---|
| **Spec** | **Learning Objective** | **Spec** | **Learning Objective** | **Guidance** | **Impact Summary** | **What you need to do** | **Qualified opinion** |
| 1.1.a | The purpose of the CPU | 1.1.1a | The purpose of the CPU:<br>○ The fetch-execute cycle | What actions occur at each stage of the fetch-execute cycle | Low<br>Clarified | Probably nothing as you were covering this across 1.1a and 1.1b/d | Actions and purpose are two different things so the additional information contradicts the learning objective., but it's likely this was taught anyway under 1.1.b/d |
| 1.1.b | Von Neumann architecture:<br>○ MAR (Memory Address Register)<br>○ MDR (Memory Data Register)<br>○ Program Counter<br>○ Accumulator | 1.1.1c | Von Neumann architecture:<br>○ MAR (Memory Address Register)<br>○ MDR (Memory Data Register)<br>○ Program Counter<br>○ Accumulator | The purpose of each register, what it stores (data or address)<br>The difference between storing data and an address<br><br>No: Knowledge of passing of data between registers in each stage | Medium<br>Clarified | No need to teach about how data passes between registers during FDE cycle | Although not required, it's still a good stretch and challenge activity and helps understand their purpose but learners wont' be tested on passing of data. |
| 1.1.c | common CPU components and their function:<br>○ ALU (Arithmetic Logic Unit)<br>○ CU (Control Unit)<br>○ Cache | 1.1.1b | common CPU components and their function:<br>○ ALU (Arithmetic Logic Unit)<br>○ CU (Control Unit)<br>○ Cache<br>○ Registers | The role/purpose of each component and what it manages, stores, or controls during the fetch-execute cycle | Low<br>Clarified | Registers are already covered in 1.1.1c | |
| 1.1.d | the function of the CPU as fetch and execute instructions stored in memory | 1.1.1a | The purpose of the CPU:<br>○ The fetch-execute cycle | What actions occur at each stage of the fetch-execute cycle | Low<br>Clarified | Probably nothing as you were covering this across 1.1a and 1.1b/d | Actions and purpose are two different things so the additional information contradicts the learning objective., but it's likely this was taught anyway under 1.1.b/d |
| 1.1.e | how common characteristics of CPUs affect their performance:<br>○ clock speed<br>○ cache size<br>○ number of cores | 1.1.1d | how common characteristics of CPUs affect their performance:<br>○ clock speed<br>○ cache size<br>○ number of cores | Understanding of each characteristic as listed<br>The effects of changing any of the common characteristics on system performance, either individually or in combination | Medium<br>Clarified | Ensure combination of clock speed, cache size and number of cores is also covered. | |
| 1.1.f | embedded systems:<br>○ purpose of embedded systems<br>○ examples of embedded sytems. | 1.1.1e | The purpose and characteristics of embedded systems | What embedded systems are<br>Typical characteristics of embedded systems | Low<br>Added | Characteristics in addition to purpose and examples | It's likely characteristics were taught anyway |
| | | 1.1.1f | Examples of embedded systems | Familiarity with a range of different embedded systems | | | |

## 1.2 Memory

| J276 | | J277 | | | What difference does it make? | | |
|---|---|---|---|---|---|---|---|
| Spec | Learning Objective | Spec | Learning Objective | Guidance | Impact Summary | What you need to do | Qualified opinion |
| 1.2.a | the difference between RAM and ROM | 1.2.1b | the difference between RAM and ROM | Key characteristics of RAM and ROM | None | | |
| 1.2.b | the purpose of ROM in a computer system | 1.2.1c | the purpose of ROM in a computer system | | None | | |
| 1.2.c | the purpose of RAM in a computer system | 1.2.1e | the purpose of RAM in a computer system | | None | | |
| 1.2.d | the need for virtual memory | 1.2.1e | the need for virtual memory | Why virtual memory may be needed in a system<br>How virtual memory works<br>○ Transfer of data between RAM and HDD when RAM is filled | Low<br>Added | Ensure you cover how virtual memory works as well as the need for it | |
| 1.2.e | ~~flash memory~~ | 1.2.2b | common types of storage:<br>○ optical<br>○ magnetic<br>○ solid state | Recognise a range of secondary storage devices/media<br>Differences between each type of storage device/medium<br>No: Understanding of the component parts of these types of storage | Low<br>Moved | Flash memory from 1.2.e has been removed but is covered as part of solid state storage | |

## 1.3 Storage

| J276 | | J277 | | | What difference does it make? | | |
|---|---|---|---|---|---|---|---|
| Spec | Learning Objective | Spec | Learning Objective | Guidance | Impact Summary | What you need to do | Qualified opinion |
| 1.3.a | the need for secondary storage | 1.2.2a | the need for secondary storage | Why computers have secondary storage | None | | |
| 1.3.b | data capacity and calculation of data capacity requirements | 1.2.3c | data capacity and calculation of data capacity requirements | Calculate capacity of devices<br>Calculate required capacity for a given set of files<br>Calculate file sizes of sound, images and text files<br>o sound file size = sample rate x duration (s) x bit depth<br>o image file size = colour depth x image height (px) x image width (px)<br>o text file size = bits per character x number of characters<br><br>Allowed but not necessary:<br>Use of 1,024 for conversions and calculations would be acceptable<br>Allowance for metadata in calculations may be used | Medium Added<br><br>Medium Clarified | Ensure students can calculate minimum file sizes (excluding metadata) for plain text files, images and sound files. | Metadata is mentioned but it's not necessary and so it's best just to refer to the minimum file size and ignore metadata.<br><br>Although 1024 is allowed, students should stick to 1000 bytes in a kilobyte as 1024 is the number of bytes in a kibibyte and just complicates calculations when a calculator isn't available. |
| 1.3.c | common types of storage:<br>○ optical<br>○ magnetic<br>○ solid state | 1.2.2b | common types of storage:<br>○ optical<br>○ magnetic<br>○ solid state | Recognise a range of secondary storage devices/media<br>Differences between each type of storage device/medium<br><br>No: Understanding of the component parts of these types of storage | Medium Added<br><br>Medium Clarified | Ensure you cover the differences between each type of storage device/medium.<br><br>You don't need to cover component parts like read/write heads.<br><br>Flash memory from 1.2.e has been removed but is covered as part of solid state storage | Although component parts aren't needed, it can sometimes aid with teaching the differences between the devices and so you may still want to cover it or include it as a stretch and challenge activity. |
| 1.3.d | suitable storage devices and storage media for a given application, ~~and~~ the advantages and disadvantages of these, using characteristics:<br>○ capacity<br>○ speed<br>○ portability<br>○ durability<br>○ reliability<br>○ cost. | 1.2.2c | suitable storage devices and storage media for a given application | Be able to apply their knowledge in context within scenarios | None | | |
| | | 1.2.2d | the advantages and disadvantages of these, using characteristics:<br>○ capacity<br>○ speed<br>○ portability<br>○ durability<br>○ reliability<br>○ cost. | Compare advantages/disadvantages for each storage device | | | |

## 1.4 Wired and wireless networks

| J276 | | J277 | | | What difference does it make? | | |
|---|---|---|---|---|---|---|---|
| **Spec** | **Learning Objective** | **Spec** | **Learning Objective** | **Guidance** | **Impact Summary** | **What you need to do** | **Qualified opinion** |
| 1.4.a | types of networks:<br>○ LAN (Local Area Network)<br>○ WAN (Wide Area Network) | 1.3.1a | The characteristics of LANs and WANs including common examples of each<br>Apply understanding of networks to a given scenario | The characteristics of LANs and WANs including common examples of each<br>Apply understanding of networks to a given scenario | Low<br>Clarified<br>High<br>Removed | Include examples of common LANs and WANs<br>You no longer need to cover virtual networks (old 1.4.f). | Most teaching would have included this already. |
| 1.4.b | factors that affect the performance of networks | 1.3.1b | Understanding of different factors that can affect the performance<br>of a network, e.g.:<br>o Number of devices connected<br>o Bandwidth | Understanding of different factors that can affect the performance<br>of a network, e.g.:<br>○ Number of devices connected<br>○ Bandwidth | Low<br>Clarified | Ensure at least these 2 examples are covered. Students can use other answers in an exam. | |
| 1.4.c | the different roles of computers in a client-server and a peer-to-peer network | 1.3.1c | the different roles of computers in a client-server and a peer-to-peer network | Concept of clients requesting/using services from a server | None | | |
| 1.4.d | the hardware needed to connect stand-alone computers into a Local Area Network:<br>○ wireless access points<br>○ routers~~/~~switches<br>○ NIC (Network Interface Controller/Card)<br>○ transmission media | 1.3.1d | the hardware needed to connect stand-alone computers into a Local Area Network:<br>○ wireless access points<br>○ routers<br>○ switches<br>○ NIC (Network Interface Controller/Card)<br>○ transmission media | The tasks performed by each piece of hardware | Low<br>SPAG | Teach routers and switches as two distinctly separate components. | These are 2 separate components with different functions so it's right they are separated. |
| 1.4.e | the internet as a worldwide collection of computer networks:<br>○ DNS (Domain Name Server)<br>○ hosting<br>○ the cloud | 1.3.1e | the internet as a worldwide collection of computer networks:<br>○ DNS (Domain Name Server)<br>○ hosting<br>○ the cloud<br>○ web servers and clients | The concept of the Internet as a network of computer networks<br>A DNS's role in the conversion of a URL to an IP address<br>Concept of servers providing services (e.g. Web server = Web pages, File server = file storage/retrieval)<br>The Cloud: remote service provision (e.g. storage, software, processing)<br>Advantages and disadvantages of the Cloud | Medium<br>Clarified | DNS is clarified that you only need to teach about its role of converting URL to an IP address using a database.<br>For hosting, ensure you cover at least web hosting and file hosting including web servers and clients..<br>You need to cover advantages and disadvantages of the Cloud | Most centres probably previously focused on web hosting rather than other types of hosting previously so this helps to clarify.<br>Although many textbooks would have covered advantages and disadvantages of the Cloud, it wasn't previously specified. |
| 1.4.f | ~~the concept of virtual networks.~~ | | | | High<br>Removed | No longer need to teach about virtual networks. | This was always a contentious topic so good to see it's been removed. |

## 1.5 Network topologies, protocols and layers

| J276 | | J277 | | | What difference does it make? | | |
|---|---|---|---|---|---|---|---|
| **Spec** | **Learning Objective** | **Spec** | **Learning Objective** | **Guidance** | **Impact Summary** | **What you need to do** | **Qualified opinion** |
| 1.5.a | star and mesh network topologies | 1.3.1f | star and mesh network topologies | Advantages and disadvantages of the Star and Mesh topologies | Medium Clarified | You need to include the advantages and disadvantages. | Many centres will have taught this anyway, but it's now clear that it is needed. |
| 1.5.b | Wifi:<br>○ ~~frequency and channels~~<br>○ ~~encryption~~ | 1.3.2a | Modes of connection:<br>o Wired<br>• Ethernet<br>o Wireless<br>• Wi-Fi<br>• Bluetooth | Compare benefits and drawbacks of wired versus wireless connection Recommend one or more connections for a given scenario<br>*No: Understand how Ethernet, Wi-Fi and Bluetooth protocols work* | High Added<br>High Removed<br>Low Clarified | Benefits and drawbacks of wired and wireless need covering.<br>Students need to be able to justify wired or Wi-Fi or Bluetooth (or combination of) based on a scenario.<br>Need to cover Bluetooth.<br>No need to cover frequency and channels for Wi-Fi.<br>No need to teach Ethernet, Wi-Fi or Bluetooth protocols.<br>Encryption isn't now restricted to Wi-Fi encryption so cover it in a wider sense. | Ethernet was an outlier before in 1.5.c and it was never clear what to teach. This makes it much clearer. Knowing that the protocols aren't needed simplifies the teaching of this topic.<br>Frequency and channels was a challenging topic that has now been removed.<br>Although encryption has been removed from Wi-Fi, it is still covered in a wider sense in 1.3.2b which isn't just restricted to Wi-Fi.<br>Most centres would be covering the purpose of encryption anyway and making reference to it in HTTPS so impact should be low. |
| | | 1.3.2b | Encryption | The principle of encryption to secure data across network connections | | | |
| 1.5.c | ~~ethernet~~ | 1.3.2a | Modes of connection:<br>o Wired<br>• Ethernet<br>o Wireless<br>• Wi-Fi<br>• Bluetooth | Compare benefits and drawbacks of wired versus wireless connection Recommend one or more connections for a given scenario<br>No: Understand how Ethernet, Wi-Fi and Bluetooth protocols work | Medium Removed | No need to teach Ethernet protocols<br>*See also 1.5.b* | The emphasis here is more on Ethernet as a method of wired networking |

| J276 | | J277 | | | What difference does it make? | | |
|---|---|---|---|---|---|---|---|
| Spec | Learning Objective | Spec | Learning Objective | Guidance | Impact Summary | What you need to do | Qualified opinion |
| 1.5.d | the uses of IP addressing, MAC addressing, and protocols including:<br>○ TCP/IP (Transmission Control Protocol/Internet Protocol)<br>○ HTTP (Hyper Text Transfer Protocol)<br>○ HTTPS (Hyper Text Transfer Protocol Secure)<br>○ FTP (File Transfer Protocol)<br>○ POP (Post Office Protocol)<br>○ IMAP (Internet Message Access Protocol)<br>○ SMTP (Simple Mail Transfer Protocol) | 1.3.2c | IP addressing and MAC addressing | IP addressing and the format of an IP address (IPv4 and IPv6)<br>A MAC address is assigned to devices; its use within a network<br>*No: Understand differences between static and dynamic, or public and private IP addresses* | Low<br>Clarified | Ensure format of IPv6 is covered as well as IPv4.<br>No longer need to teach static, dynamic, public or private IP addresses. | Some textbooks already covered IPv6.<br>Although the differences between types of IP addresses isn't needed for the exam, it's a useful stretch and challenge activity. |
| | | 1.3.2e | Common protocols including:<br>○ TCP/IP (Transmission Control Protocol/Internet Protocol)<br>○ HTTP (Hyper Text Transfer Protocol)<br>○ HTTPS (Hyper Text Transfer Protocol Secure)<br>○ FTP (File Transfer Protocol)<br>○ POP (Post Office Protocol)<br>○ IMAP (Internet Message Access Protocol)<br>○ SMTP (Simple Mail Transfer Protocol) | The principle of a (communication) protocol as a set of rules for transferring data<br>That different types of protocols are used for different purposes<br>The basic principles of each protocol i.e. its purpose and key features | | | |
| 1.5.e | the concept of layers | 1.3.2f | the concept of layers | How layers are used in protocols, and the benefits of using layers;<br>for a teaching example, please refer to the 4-layer TCP/IP model<br>*No: Knowledge of the names and function of each TCP/IP layer* | Low<br>Added<br>Low<br>Clarified | Ensure you include the benefits of using layers.<br>Students no longer need to know the names and function of each TCP/IP layer. | The benefits were probably covered in most teaching anyway.<br>Although names and function of TCP/IP layer aren't needed, it's a useful teaching tool, but students won't be tested on it in the exam. They can use it as an example in their answers though. |
| 1.5.f | ~~packet switching.~~ | | | | High<br>Removed | No longer need to teach packet switching | |

1.6 System security

| J276 | | J277 | | | What difference does it make? | | |
|---|---|---|---|---|---|---|---|
| Spec | Learning Objective | Spec | Learning Objective | Guidance | Impact Summary | What you need to do | Qualified opinion |
| 1.6.a | forms of attack | 1.4.1a | Forms of attack:<br>o Malware<br>o Social engineering, e.g. phishing, people as the 'weak point'<br>o Brute-force attacks<br>o Denial of service attacks<br>o Data interception and theft<br>o The concept of SQL injection | Threats posed to devices/systems<br>Knowledge/principles of each form of attack including:<br>o How the attack is used<br>o The purpose of the attack | Medium Added<br>Medium Removed | You may want to teach about other social engineering techniques like pharming and shouldering but these won't be specifically tested in an exam.<br>For each attack, ensure you cover its purpose (what it's intended to do) and how it is used<br>You no longer need to cover poor network policy. | |
| 1.6.b | threats posed to networks:<br>◯ malware<br>◯ phishing<br>◯ people as the 'weak point' in secure systems (social engineering)<br>◯ brute force attacks<br>◯ denial of service attacks<br>◯ data interception and theft<br>◯ the concept of SQL injection<br>◯ ~~poor network policy~~ | | | | | | |
| 1.6.c | identifying and preventing vulnerabilities:<br>◯ penetration testing<br>◯ ~~network forensics~~<br>~~◯ network policies~~<br>◯ anti-malware software<br>◯ firewalls<br>◯ user access levels<br>◯ passwords<br>◯ encryption. | 1.4.1a | Common prevention methods:<br>o Penetration testing<br>o Anti-malware software<br>o Firewalls<br>o User access levels<br>o Passwords<br>o Encryption<br>o Physical security | Understanding of how to limit the threats posed in 1.4.1<br>Understanding of methods to remove vulnerabilities<br>Knowledge/principles of each prevention method:<br>o What each prevention method may limit/prevent<br>o How it limits the attack | Medium Added<br>Medium Removed | You need to include physical security in your teaching.<br>You no longer need to cover network forensics or network policies. | |

## 1.7 Systems software

| J276 | | J277 | | | What difference does it make? | | |
|---|---|---|---|---|---|---|---|
| **Spec** | **Learning Objective** | **Spec** | **Learning Objective** | **Guidance** | **Impact Summary** | **What you need to do** | **Qualified opinion** |
| 1.7.a | the purpose and functionality of ~~systems~~ software | 1.5.2a | the purpose and functionality of utility software | Understand that computers often come with utility software, and how this performs housekeeping tasks | Low<br>Changed<br>Medium<br>Removed | Cover what you will teach in 1.5.1b but also cover the purpose of utility software in general.<br>It's no longer necessary to cover systems software as the set of utility software and operating system, just utility software. | Personally, I'd have included this as the introduction to 1.5.1b<br>There's no harm in mentioning to students that system software is the set that includes utility software and operating systems |
| 1.7.b | operating systems:<br>○ user interface<br>○ memory management/multitasking<br>○ peripheral management and drivers<br>○ user management<br>○ file management | 1.5.1a | The purpose and functionality of operating systems:<br>o user interface<br>o memory management and multitasking<br>o peripheral management and drivers<br>o user management<br>o file management | What each function of an operating system does<br>o Features of a user interface<br>o Memory management, e.g. the transfer of data between memory, and how this allows for multitasking<br>Understand that:<br>o Data is transferred between devices and the processor<br>o This process needs to be managed and what this entails (e.g. the use of buffers when transferring data to a printer)<br>User management functions, e.g.:<br>o Allocation of an account<br>o Access rights<br>o Security, etc.<br>File management, and the key features, e.g.:<br>o Naming<br>o Allocating to folders<br>o Moving files<br>o Saving, etc.<br>*No: Understanding of paging or segmentation* | High<br>Clarified | Students need to understand how memory management enables multitasking.<br>Students need to be aware of data transfer between peripherals and the processor including the use of buffers.<br>Some examples of user and file management are given but students can use other examples in their answers.<br>There's no need to cover paging or segmentation. | There's quite a lot added here and a bit unclear on the depth required such as buffering and the transfer of data between devices and the processor. |
| 1.7.c | utility system software:<br>○ encryption software<br>○ defragmentation<br>○ data compression<br>○ ~~the role and methods of backup:~~<br>~~full~~<br>~~incremental.~~ | 1.5.2b | utility system software:<br>○ encryption software<br>○ defragmentation<br>○ data compression | Purpose of the identified utility software and why it is required | No longer need to teach about full and incremental backups. | Still remind students regularly about the importance of backing up their own work. | No longer need to teach about full and incremental backups. |

1.8 Ethical, legal, cultural and environmental concerns

| J276 | | J277 | | | What difference does it make? | | |
|---|---|---|---|---|---|---|---|
| **Spec** | **Learning Objective** | **Spec** | **Learning Objective** | **Guidance** | **Impact Summary** | **What you need to do** | **Qualified opinion** |
| 1.8.a | how to investigate and discuss Computer Science technologies while considering:<br>○ ethical issues<br>○ legal issues<br>○ cultural issues<br>○ environmental issues.<br>○ privacy issues. | 1.6.1a | Impacts of digital technology on wider society including:<br>o Ethical issues<br>o Legal issues<br>o Cultural issues<br>o Environmental issues<br>o Privacy issues | Technology introduces ethical, legal, cultural, environmental and privacy issues<br>Knowledge of a variety of examples of digital technology and how this impacts on society<br>An ability to discuss the impact of technology based around the issues listed | Low<br>Clarified | Students need to understand a variety of examples of digital technology and their impact. This can be achieved through regular wider reading such as the articles Tweeted at @paullongnet | This has tidied up the whole area of impacts because it was spread across several bullet points. The old 1.8.a/b/c/d have been merged into a single learning objective. |
| 1.8.b | ~~how key stakeholders are affected by technologies~~ | | | | | | |
| 1.8.c | ~~environmental impact of Computer Science~~ | | | | | | |
| 1.8.d | ~~cultural implications of Computer Science~~ | | | | | | |
| 1.8.e | open source vs proprietary software | 1.6.1b | Legislation relevant to Computer Science:<br>o The Data Protection Act 2018<br>o Computer Misuse Act 1990<br>o Copyright Designs and Patents Act 1988<br>o Software licences (i.e. open source and proprietary) | The purpose of each piece of legislation and the specific actions it allows or prohibits<br>The need to license software and the purpose of a software licence<br>Features of open source (providing access to the source code and the ability to change the software)<br>Features of proprietary (no access to the source code, purchased commonly as off-the-shelf)<br>Recommend a type of licence for a given scenario including benefits and drawbacks | Low<br>Moved | Moved to 1.6.1b | |
| 1.8.f | legislation relevant to Computer Science:<br>○ The Data Protection Act ~~1998~~<br>○ Computer Misuse Act 1990<br>○ Copyright Designs and Patents Act 1988<br>○ ~~Creative Commons Licensing~~<br>~~○ Freedom of Information Act 2000.~~ | | | | High<br>Removed<br><br>High<br>Updated | No longer need to teach about Creative Commons Licensing or Freedom of Information Act<br><br>Ensure new aspects introduced to the Data Protection Act in 2018 such as right to be forgotten and implicit consent are included in your teaching. | Open source and proprietary were already covered in 1.8.e but you don't need to teach about other types of software licences.<br>The use of i.e. instead of e.g. makes it clear that OCR don't need students to know about single-user, multi-user, site licences etc. |

## Component 2: COMPUTATIONAL THINKING, ALGORITHMS AND PROGRAMMING

### 2.1 Algorithms

| J276 | | J277 | | | What difference does it make? | | |
|---|---|---|---|---|---|---|---|
| **Spec** | **Learning Objective** | **Spec** | **Learning Objective** | **Guidance** | **Impact Summary** | **What you need to do** | **Qualified opinion** |
| 2.1.a | computational thinking:<br>◯ abstraction<br>◯ decomposition<br>◯ algorithmic thinking | 2.1.1a | Principals of computational thinking:<br>◯ abstraction<br>◯ decomposition<br>◯ algorithmic thinking | Understanding of these principles and how they are used to define and refine problems | None | | |
| 2.1.b | standard searching algorithms:<br>◯ binary search<br>◯ linear search | 2.1.3a | standard searching algorithms:<br>◯ binary search<br>◯ linear search | Understand the main steps of each algorithm<br>Understand any pre-requisites of an algorithm<br>Apply the algorithm to a data set<br>Identify an algorithm if given the code for it<br>*No: To remember the code for these algorithms* | Low Clarified | Students don't need to be able to rewrite the code for these algorithms but if shown the code, they need to be able to identify the algorithm by name and its purpose. | This is what most teachers have expected has been the case with J276 but now it is specified. |
| 2.1.c | standard sorting algorithms:<br>◯ bubble sort<br>◯ merge sort<br>◯ insertion sort | 2.1.3b | standard sorting algorithms:<br>◯ bubble sort<br>◯ merge sort<br>◯ insertion sort | | | | |
| 2.1.d | how to produce algorithms using:<br>◯ pseudocode<br>◯ using flow ~~diagrams~~ | 2.1.2c | create, interpret, correct, complete, and refine algorithms using:<br>o pseudocode<br>o flowcharts<br>o reference language / high-level programming language | Complete, write or refine an algorithm using the techniques listed<br>*See Flowchart Symbols on page 15 of specification* | High Added<br>Low SPAG | Students need to use the flowchart symbols on page 15 of the specification.<br>Students will need to be able to refine algorithms as well as create, interpret, correct and complete them.<br>In Section B of the Component 2 exam, students will need to use either OCR's Exam Reference Language or a high-level language.<br>The term flowcharts is used instead of flow diagrams. | There will be a lot of opinions about whether students should use a language or pseudo-code. Section B is 15% of the whole qualification and there will be marks available for structure, logic and syntax. What is unknown is how forgiving examiners will be with syntax (see page 24 of the specification).<br>Refine has been added to enable assessment of the refine process in the Programming Task.<br>The term flowcharts is used correctly in the new spec. |
| 2.1.e | interpret, correct or complete algorithms. | | | | | | |
| | | 2.1.2e | Trace tables | Create and use trace tables to follow an algorithm | High Added | Students need to be able to create and use trace tables. | Some teachers thought this was implied by being able to interpret algorithms or would have taught it as a skill anyway, but now it is mandatory. |

## 2.2 Programming techniques

| J276 | | J277 | | | What difference does it make? | | |
|---|---|---|---|---|---|---|---|
| Spec | Learning Objective | Spec | Learning Objective | Guidance | Impact Summary | What you need to do | Qualified opinion |
| 2.2.a | the use of variables, constants, operators, inputs, outputs and assignments | 2.2.1a | the use of variables, constants, operators, inputs, outputs and assignments | Practical use of the techniques in a high-level language within the classroom | None | | |
| 2.2.b | the use of the three basic programming constructs used to control the flow of a program:<br>○ sequence<br>○ selection<br>○ iteration (count and condition controlled loops) | 2.2.1b | the use of the three basic programming constructs used to control the flow of a program:<br>○ sequence<br>○ selection<br>○ iteration (count- and condition-controlled loops) | == Equal to<br>!= Not equal to<br>< Less than<br><= Less than or equal to<br>> Greater than<br>>= Greater than or equal to | Terms | Just a hyphen added to count-controlled and condition-controlled loops | |
| 2.2.c | the use of basic string manipulation | 2.2.3a | the use of basic string manipulation | Ability to manipulate strings, including:<br>o Concatenation<br>o Slicing | Low<br>Clarified | String manipulation must include concatenation and slicing. | This would have been standard practice at schools but it now narrows down what aspects of string manipulation are likely to be covered. |
| 2.2.d | the use of basic file handling operations:<br>○ open<br>○ read<br>○ write<br>○ close | 2.2.3b | the use of basic file handling operations:<br>○ open<br>○ read<br>○ write<br>○ close | | None | | |
| 2.2.e | the use of records to store data | 2.2.3c | the use of records to store data | | None | | |
| 2.2.f | the use of SQL to search for data | 2.2.3d | the use of SQL to search for data | SQL commands:<br>o SELECT<br>o FROM<br>o WHERE | Low<br>Clarified | The only SQL commands to teach are SELECT, FROM, WHERE | This clarifies that INNER or OUTER JOINS aren't required or sorting into order.<br><br>OCR have stated they have removed LIKE, AND, OR and wildcards although this is not clear in the specification. |
| 2.2.g | the use of arrays (or equivalent) when solving problems, including both one and two dimensional arrays | 2.2.3e | the use of arrays (or equivalent) when solving problems, including both one-dimensional and two-dimensional arrays | Arrays as fixed length static structures | Low<br>SPAG | No action required | |

| J276 | | J277 | | | What difference does it make? | | |
|---|---|---|---|---|---|---|---|
| Spec | Learning Objective | Spec | Learning Objective | Guidance | Impact Summary | What you need to do | Qualified opinion |
| 2.2.h | how to use sub programs (functions and procedures) to produce structured code | 2.2.3f | how to use sub programs (functions and procedures) to produce structured code | The use of functions<br>The use of procedures<br>Where to use functions and procedures effectively | None | | |
| 2.2.i | the use of data types:<br>○ integer<br>○ real<br>○ Boolean<br>○ character and string<br>○ casting | 2.2.2a | the use of data types:<br>○ integer<br>○ real<br>○ Boolean<br>○ character and string<br>○ casting | Practical use of the data types in a high-level language within the classroom<br>Ability to choose suitable data types for data in a given scenario<br>Understand that data types may be temporarily changed through casting, and where this may be useful | None | | |
| 2.2.j | the common arithmetic operators | 2.2.1c | the common arithmetic operators | + Addition<br>– Subtraction<br>* Multiplication<br>/ Division<br>MOD Modulus<br>DIV Quotient<br>^ Exponentiation (to the power) | None | | |
| 2.2.k | the common Boolean operators. | 2.2.1d | the common Boolean operators AND, OR and NOT | | Low<br>Clarified | You only need to teach AND, OR and NOT. | Most teachers were doing this anyway but it helps to clarify. |

## 2.3 Producing robust programs

| J276 | | J277 | | | What difference does it make? | | |
|---|---|---|---|---|---|---|---|
| **Spec** | **Learning Objective** | **Spec** | **Learning Objective** | **Guidance** | **Impact Summary** | **What you need to do** | **Qualified opinion** |
| 2.3.a | defensive design considerations:<br>◯ ~~input sanitisation~~/validation<br>◯ ~~planning for contingencies~~<br>◯ anticipating misuse<br>◯ authentication | 2.3.1a | defensive design considerations:<br>◯ anticipating misuse<br>◯ authentication | Understanding of the issues a programmer should consider to ensure that a program caters for all likely input values<br>Authentication to confirm the identity of a user<br>Practical experience of designing . . . simple authentication (e.g. username and password) | Low<br>Moved<br>Low<br>Removed | No longer need to cover input sanitisation or planning for contingencies.<br>Validation has moved to 2.3.1b<br>Students must have experience of designing username and password authentication routines | This suggests exam questions would require students to be able to design and maybe even write a simple username and password routine. |
| | | 2.3.1b | Input validation | Understanding of how to deal with invalid data in a program<br>Practical experience of designing input validation | Low<br>Moved<br>Low<br>Clarified<br>High<br>Removed | Input validation was covered in 2.3.a previously so it's still the same teaching process.<br>Students must have experience of designing validation routines | This suggests exam questions would require students to be able to design and maybe even write a simple validation routine. |
| 2.3.b | maintainability:<br>◯ comments<br>◯ indentation | 2.3.1c | Maintainability:<br>o Use of sub programs<br>o Naming conventions<br>o Indentation<br>o Commenting | Understand why commenting is useful and apply this appropriately | Low<br>Added | Students need to understand how sub programs and naming conventions help make programs maintainable. | |
| 2.3.c | the purpose of testing | 2.3.2a | the purpose of testing | | None | | |
| 2.3.d | types of testing:<br>◯ iterative<br>◯ final/terminal | 2.3.2b | types of testing:<br>◯ iterative<br>◯ final/terminal | The difference between testing modules of a program during development and testing the program at the end of production | Low<br>Clarified | As well as understanding iterative and final/terminal testing, students need to understand the difference between these with iterative focused on testing modules during development and final focused on testing the final program. | It's likely this was taught by most teachers already. |

| J276 | | J277 | | | What difference does it make? | | |
|---|---|---|---|---|---|---|---|
| Spec | Learning Objective | Spec | Learning Objective | Guidance | Impact Summary | What you need to do | Qualified opinion |
| 2.3.e | ~~how to~~ identify syntax and logic errors | 2.3.2c | identify syntax and logic errors | Syntax errors as errors which break the grammatical rules of the programming language and stop it from being run/translated<br>Logic errors as errors which produce unexpected output | Low<br>SPAG | No action required.<br>*Note: this is also in 2.1.2d* | It's a bit odd these are 2 separate items so they could naturally be taught together rather than separately. |
| | | 2.1.2d | Identify common errors | Identify syntax/logic errors in code and suggest fixes | Low<br>Added | Students now need to be able to suggest fixes to errors.<br>*Note: also in 2.3.2c* | |
| 2.3.f | selecting and using suitable test data. | 2.3.2d | selecting and using suitable test data:<br>o Normal<br>o Boundary<br>o Invalid<br>o Erroneous | Normal test data as data which should be accepted by a program without causing errors<br>Boundary test data as data of the correct type which is on the very edge of being valid<br>Invalid test data as data of the correct type but outside accepted validation limit<br>Erroneous test data as data of the incorrect type which should be rejected by a computer system<br>Ability to identify suitable test data for a given scenario<br>Ability to create/complete a test plan | Low<br>Clarified | Students don't need to know about invalid boundary data, just valid boundary data.<br>The definition used for erroneous data here is wrong. Erroneous data is actually the same as Invalid data. If OCR don't update the specification, you'll have to teach it and make sure students are clear they have to use terms incorrectly, but how OCR want them, in the exam. | I've asked OCR if they will remove the term erroneous and its incorrect definition. The only mention I can find of them being different is on BBC Bytesize which is clearly wrong. There are no other sources that suggest they are different and plenty of reliable sources that identify them as being the same. |

## 2.4 Computational logic

| J276 | | J277 | | | What difference does it make? | | |
|---|---|---|---|---|---|---|---|
| **Spec** | **Learning Objective** | **Spec** | **Learning Objective** | **Guidance** | **Impact Summary** | **What you need to do** | **Qualified opinion** |
| 2.4.a | ~~why~~ data is represented in computer systems in binary form | 1.2.3b | how data needs to be converted into a binary format to be processed by a computer. | Why data must be stored in binary format | Low<br>Clarified | The 'how' is covered in converting between number bases. The 'why' was already taught in 2.4.a so continue to teach it | There is a contradiction in the spec between 'how' and 'why'.<br>This was also in 2.6b |
| 2.4.b | simple logic diagrams using the operations AND, OR and NOT | 2.4.1a | simple logic diagrams using the operators AND, OR and NOT | Recognition of each gate symbol<br>Understanding of how to create, complete or edit logic diagrams and truth tables for given scenarios | Low<br>Clarified | In order to achieve 2.4.c/d/e it's likely all of this is already covered.<br>*See page 20 of specification for logic gate symbols* | |
| 2.4.c | truth tables | 2.4.1b | truth tables | Knowledge of the truth tables for each logic gate | Low<br>Clarified | Carry on teaching the NOT, AND, OR gate truth tables. | |
| 2.4.d | combining Boolean operators using AND, OR and NOT ~~to two levels~~ | 2.4.1c | combining Boolean operators using AND, OR and NOT | Ability to work with more than one gate in a logic diagram | High<br>Added | You may need to teach students how to create more complex diagrams. Use this as an opportunity for differentiation and stretch and challenge. | There is now no limit on how complex a problem could be combining Boolean operators. |
| 2.4.e | applying logical operators in ~~appropriate~~ truth tables to solve problems | 2.4.1d | applying logical operators in truth tables to solve problems | Understanding of how to create, complete or edit logic diagrams and truth tables for given scenarios<br>Alternatives<br>Use of other valid notation will be accepted within the examination, e.g. Using T/F for 1/0, or V for OR, etc. | Low<br>SPAG | No action required | |
| 2.4.f | applying computing-related mathematics:<br>○ +<br>○ –<br>○ /<br>○ *<br>○ Exponentiation (^)<br>○ MOD<br>○ DIV | 2.2.1c | the common arithmetic operators | + Addition<br>– Subtraction<br>* Multiplication<br>/ Division<br>MOD Modulus<br>DIV Quotient<br>^ Exponentiation (to the power) | None | | |

## 2.5 Translators and facilities of languages

| J276 | | J277 | | | What difference does it make? | | |
|---|---|---|---|---|---|---|---|
| **Spec** | **Learning Objective** | **Spec** | **Learning Objective** | **Guidance** | **Impact Summary** | **What you need to do** | **Qualified opinion** |
| 2.5.a | characteristics and purpose of different levels of programming language, ~~including~~ low level languages | 2.5.1a | Characteristics and purpose of different levels of programming language:<br>o High-level languages<br>o Low-level languages | The differences between high- and low-level programming languages | Low<br>Clarified | It's most likely you were already including high-level languages so probably no action required.<br>If you weren't already including the differences then cover these explicitly. | |
| 2.5.b | the purpose of translators | 2.5.1b | the purpose of translators | The need for translators | None | | |
| 2.5.c | the characteristics of an ~~assembler~~, a compiler and an interpreter | 2.5.1c | the characteristics of a compiler and an interpreter | The differences, benefits and drawbacks of using a compiler or an interpreter<br>*No: Understanding of assemblers* | Low<br>Added<br>High<br>Deleted | Differences, benefits and drawbacks between compiler and assembler need to be covered.<br>Assemblers can be removed from your teaching. | Most teachers were probably covering differences, benefits and drawbacks as a natural part of explaining the characteristics so shouldn't be too difficult to implement. |
| 2.5.d | common tools and facilities available in an integrated development environment (IDE):<br>○ editors<br>○ error diagnostics<br>○ run-time environment<br>○ translators. | 2.5.2a | common tools and facilities available in an integrated development environment (IDE):<br>○ editors<br>○ error diagnostics<br>○ run-time environment<br>○ translators. | Knowledge of the tools that an IDE provides<br>How each of the tools and facilities listed can be used to help a programmer develop a program<br>Practical experience of using a range of these tools within at least one IDE | Low<br>Clarification | As well as the characteristics of these features, consider how they help a programmer to develop a program. | This is probably common sense but worth clarifying. |

## 2.6 Data representation

| J276 | | J277 | | | What difference does it make? | | |
|---|---|---|---|---|---|---|---|
| Spec | Learning Objective | Spec | Learning Objective | Guidance | Impact Summary | What you need to do | Qualified opinion |
| 2.6.a | bit, nibble, byte, kilobyte, megabyte, gigabyte, terabyte, petabyte | 1.2.3a | The units of data storage:<br>o Bit<br>o Nibble (4 bits)<br>o Byte (8 bits)<br>o Kilobyte (1,000 bytes or 1 KB)<br>o Megabyte (1,000 KB)<br>o Gigabyte (1,000 MB)<br>o Terabyte (1,000 GB)<br>o Petabyte (1,000 TB) | Familiarity with data units and moving between each | Low<br>Clarified | Always teach using multiples of 1,000 and use in exams | Stretch and challenge activities could include understanding kibibytes (KiB), mebibytes (MiB) etc in multiples of 1024, but it won't be assessed and 1,000 is easier to use in an exam. |
| 2.6.b | how data needs to be converted into a binary format to be processed by a computer. | 1.2.3b | how data needs to be converted into a binary format to be processed by a computer. | Why data must be stored in binary format | Low<br>Clarified | The 'how' is covered in converting between number bases. The 'why' was already taught in 2.4.a so continue to teach it | There is a contradiction in the spec between 'how' and 'why'. |
| 2.6.c | how to convert positive denary whole numbers ~~(0–255)~~ into ~~8 bit~~ binary numbers and vice versa | 1.2.4a | How to convert positive denary whole numbers to binary numbers (up to and including 8 bits) and vice versa | Denary number range 0 – 255<br>Binary number range 00000000 – 11111111 | Low<br>Clarified | The maximum number of bits was already 8 due to the limit being 0-255 but the "up to" bit means students don't need leading zeros in their binary answers unless asked for an 8-bit format. | |
| 2.6.d | how to add ~~two 8 bit~~ binary integers and explain overflow errors which may occur | 1.2.4b | how to add two binary integers together (up to and including 8 bits) and explain overflow errors which may occur | Understanding of the terms most significant bit, and least significant bit<br>Ability to deal with binary numbers containing between 1 and 8 bits<br>o e.g. 11010 is the same as 00011010 | Medium<br>Clarified | Exams may ask students to add two binary integers that are less than 8 bits each.<br>Students need to be familiar with MSB and LSB. | LSB and MSB may have been terms used when explaining overflow, but students now need to know these terms for the exam. |
| 2.6.e | binary shifts | 1.2.4e | binary shifts | Understand the effect of a binary shift (both left or right) on a number | Low<br>Clarified | This suggests students don't need to be able to carry out a binary shift. | To understand the effect of a binary shift, the best way to explain it is to show how it's done so I would teach this anyway. |
| 2.6.f | how to convert positive denary whole numbers ~~(0–255)~~ into 2 digit hexadecimal numbers and vice versa | 1.2.4c | How to convert positive denary whole numbers into 2-digit hexadecimal numbers and vice versa | Denary number range 0 – 255<br>Hexadecimal range 00 – FF | None | | |
| 2.6.g | how to convert from binary to hexadecimal equivalents and vice versa | 1.2.4d | How to convert binary integers to their hexadecimal equivalents and vice versa | Hexadecimal range 00 – FF<br>Binary number range 00000000 – 11111111 | Low<br>SPAG | Nothing | |

| J276 | | J277 | | | What difference does it make? | | |
|---|---|---|---|---|---|---|---|
| **Spec** | **Learning Objective** | **Spec** | **Learning Objective** | **Guidance** | **Impact Summary** | **What you need to do** | **Qualified opinion** |
| 2.6.h | check digits. | | | | High<br>Removed | You no longer need to teach check digits. | There was a lot of confusion as to whether this included check digits in terms of validation like ISBN or parity checks so it's welcome to see this removed. |
| 2.6.i | the use of binary codes to represent characters | 1.2.4f | the use of binary codes to represent characters | How characters are represented in binary | None | | |
| 2.6.j | the term 'character-set' | 1.2.4g | the term 'character-set' | Understand how character sets are logically ordered, e.g. the code for 'B' will be one more than the code for 'A' | Low<br>Added | Ensure you teach that if one character's code is known, it's often possible to work out another one using sequence. | Most textbooks covered this anyway. |
| 2.6.k | the relationship between the number of bits per character in a character set and the number of characters which can be represented (for example ASCII, ~~extended ASCII~~ and Unicode). | 1.2.4h | the relationship between the number of bits per character in a character set, and the number of characters which can be represented, e.g.:<br>o ASCII<br>o Unicode | How the number of characters stored is limited by the bits available<br>The differences between and impact of each character set<br>Binary representation of ASCII in the exam will use 8 bits<br>*No: Memorisation of character set codes* | Medium<br>Clarified | Ensure you cover binary representation of ASCII as well as decimal representation.<br>OCR won't use the term "extended ASCII" in an exam but students can make reference to it in an answer. | This has added a slightly extra layer of complexity but helps in the understanding that the codes are actually binary and so decimal is just an intermediary step for the ASCII code. |
| 2.6.l | how an image is represented as a series of pixels represented in binary | 1.2.4i | how an image is represented as a series of pixels, represented in binary | Each pixel has a specific colour, represented by a specific code | None | | |
| 2.6.m | metadata ~~included in the file~~ | 1.2.4j | metadata | Metadata stores additional image information (e.g. height, width, etc.) | Low<br>Clarified | No action required | |
| 2.6.n | the effect of colour depth and resolution on the size of an image file. | 1.2.4k | the effect of colour depth and resolution on:<br>o the quality of the image<br>o the size of an image file | The effect on image size and quality when changing colour depth and resolution | Medium<br>Added | Ensure you cover the effect on quality (e.g. pixilation, loss of colours) as well as size. | This would probably have been taught anyway. Do be careful though that increasing colour depth doesn't increase quality unless additional colours are also added. Similarly, increasing resolution doesn't increase quality unless smoothing takes place. |
| 2.6.o | how sound can be sampled and stored in digital form | 1.2.4l | how sound can be sampled and stored in digital form | Analogue sounds must be stored in binary | None | | |

| J276 | | J277 | | | What difference does it make? | | |
|---|---|---|---|---|---|---|---|
| Spec | Learning Objective | Spec | Learning Objective | Guidance | Impact Summary | What you need to do | Qualified opinion |
| 2.6.p | how ~~sampling intervals and other factors~~ affect the size of a sound file and the quality of its playback:<br>~~o sample size~~<br>~~o bit rate~~<br>~~o sampling frequency~~. | 1.2.4m | The effect of sample rate, duration and bit depth on:<br>o The playback quality<br>o The size of a sound file | Sample rate – measured in Hertz (Hz)<br>Duration – how many seconds of audio the sound file contains<br>Bit depth – number of bits available to store each sample (e.g. 16-bit) | Medium Terms | Use the term "sample rate" instead of "sampling frequency" and "bit depth" instead of "sample size"<br><br>No need to cover "bit rate" | OCR are now using the correct industry standard terminology. Removing "bit rate" avoids confusion with "bit depth". |
| 2.6.q | need for compression | 2.2.4n | need for compression | Common scenarios where compression may be needed | None | | |
| 2.6.r | types of compression:<br>◯ lossy<br>◯ lossless. | 1.2.4o | types of compression:<br>◯ lossy<br>◯ lossless. | Advantages and disadvantages of each type of compression<br>Effects on the file for each type of compression<br><br>*No: Ability to carry out specific compression algorithms* | Medium Clarified | Ensure you cover advantages and disadvantages of each as well as the effect on the file (e.g. lossy loses data).<br><br>No need to teach RLE. | These things were covered in most textbooks anyway, but it's now clear exactly what OCR could ask questions about.<br><br>Although students don't need to carry out an algorithm like RLE, using RLE as an example helps to explain a lossless technique and is also a useful stretch and challenge activity. |

Topics not on J276

| J276 | | J277 | | | What difference does it make? | | |
|---|---|---|---|---|---|---|---|
| Spec | Learning Objective | Spec | Learning Objective | Guidance | Impact Summary | What you need to do | Qualified opinion |
| Topics not on J276 | | 1.2.1a | The need for primary storage | "Why computers have primary storage | Low Added | Ensure you cover the reasons primary storage are needed | It's likely the need was taught anyway under 1.2.a |
| | | 1.3.2d | Network standards | The principle of a standard to provide rules for areas of computing<br>Standards allows hardware/software to interact across different manufacturers/producers<br>No: Knowledge of individual standards | High Added | This is a new topic that needs covering. | |
| | | 2.1.2a | Identify the inputs, processes, and outputs for a problem | | Medium Added | Give students algorithms or problems and ask them to identify the inputs, processes and outputs separately. | This is covered in AQA and is quite straightforward. |
| | | 2.1.2b | Structure diagrams | "Produce simple diagrams to show: | Medium Added | Give students algorithms or problems and ask them to identify the inputs, processes and outputs separately. | This is covered in AQA and is quite straightforward. |
| | | 2.2.3g | Random number generation | | High Added | Include the use of random numbers in programming tasks so that students get an experience of using them. | Would be nice to have some additional information. |
| | | 2.3.2e | Refining algorithms | | Medium Added | Through the programming task and other tasks, give students experience of refining algorithms so they can refine an unfamiliar algorithm in an exam. | |