



Computer Science Mapping to partner resources

Computer Science: Mapping to partner resources

Working with industry partners

We're working with leading industry and pedagogical organisations in Computer Science that have been specifically chosen to support our qualification's practical approach. Together, we'll save you time and effort in identifying the best resources, where best to use them in the classroom, and how to get the most impact from them.

Specification map

We have mapped the specification to specific content from our industry partners. You can find the mapping on the following pages.

- Sometimes there are specific links, sometimes page references or slide references.
- We have indicated if the material is free, or paid for.
- You can use these resources alongside the Edexcel SoW or your own SoW. We have provided the mapping linked to the specification, rather than the SoW, to give you complete flexibility.

Links to industry partners

[Craig 'n' Dave >](#)

[PG Online >](#)

[Eedi >](#)

[Paul Long >](#)

[John Philip Jones >](#)

[Microsoft MakeCode >](#)

[NCCE >](#)

| Specification reference | Subject content | Where is this covered in Craig'n'Dave resources | Link to resources | What does it include? | Free or paid for |
|--|--|--|---|--|------------------|
| Topic 1: Computational thinking | 1.1 - 1.3 | All of Topic 1 is fully covered as part of a learning journey across the follow CND Structured Learning Records: SLR7: Basic programming concepts SLR9: Robust and secure programming SLR10: Algorithms & computational logic | https://craigndave.org/pearson-edexcel-gcse-1cp2/ | Our paid for resources include: <ul style="list-style-type: none"> • Starter activities for every lesson • Student workbooks with model answers which act as knowledge organisers • End of topic tests with answers • Terminology guide, short term and long term Schemes of learning & delivery calendars • Lesson PowerPoints for every lesson plus 83 theory PowerPoint presentations • Full set of programming resources, guides, challenges and paper 2 revision material all with solutions • "My little extras pack" contain dozens of extra resources to support teachers | Paid |
| Topic 1: Computational thinking | 1.1 - 1.3 | Smart Revise online revision tool covers all of topic 1 in detail speci point for spec point. | https://smartrevise.craigndave.org/ | <ul style="list-style-type: none"> • 100s of Multiple Choice Quiz questions covering every spec point • Interactive terminology flash cards covering all subject specific terms • Full set of teacher and student analytic reports • Customised personal student revision reports • Topic filtering, revision video playlists and student awards and league tables | Paid |
| Specification reference | Subject content | Topic | Link to Craig 'n' Dave video | Description | Free or paid for |
| Topic 1: Computational thinking | 1.1 Decomposition and abstraction | Topic 1A Abstraction | https://student.craigndave.org/videos/gcse-edexcel-topic-1a-abstraction | Abstraction is the representation of essential features without including unnecessary details. It is used to reduce the complexity of systems for users, hiding how things actually work, applying algorithms to different contexts and producing suitable user interfaces. This video explains the concept at a level suitable for GCSE using the example of a sat-nav system. | Free |
| | | Topic 1A Decomposition | https://student.craigndave.org/videos/gcse-edexcel-topic-1a-decomposition | Problem decomposition, or breaking a problem down is an essential skill to becoming an experienced programmer. This video explains the concept using an everyday example of crossing a road and also how it applies to simple games design. | Free |
| | | Topic 1A Introduction to subprograms | https://student.craigndave.org/videos/gcse-edexcel-topic-1a-introduction-to-subprograms | This video provides you with a brief introduction to subroutines. We go into subroutines (procedures and functions) in more detail in a later video. | Free |
| | 1.2 Algorithms | Topic 1A How to produce algorithms | https://student.craigndave.org/videos/gcse-edexcel-topic-1a-how-to-produce-algorithms | This video explains two common methods of describing algorithms for problem decomposition: pseudocode and flow diagrams. The key flowchart symbols students can expect to see in exams are introduced. | Free |
| | | Topic 1A Introduction to programming concepts | https://student.craigndave.org/videos/gcse-edexcel-topic-1a-introduction-to-programming-concepts | This video covers the range of Python programming concepts you will need to be aware of for the paper 2 exam. Each area will have its own video under YouTube playlist 6B which covers specific topics 6.1, 6.2 and 6.3. | Free |
| | | Topic 1A Data structures and arrays | https://student.craigndave.org/videos/gcse-edexcel-topic-1a-data-structures-and-arrays | This video starts by outlining the concept of a data structure before delving deeper into arrays, both one and two-dimensional, and how they are implemented with lists in Python. It is assumed that students already have programming experience with variables. | Free |
| | | Topic 1A The common arithmetic and comparison operators | https://student.craigndave.org/videos/gcse-edexcel-topic-1a-the-common-arithmetic-and-comparison-operators | In this video we overview the main arithmetic operators you need to be familiar with, these include: Addition, Subtraction, Multiplication, Real division and Integer division. | Free |



| | | | | | |
|--|--|---|---|--|------|
| | | Topic 1A The common Boolean operators | https://student.craigndave.org/videos/gcse-edexcel-topic-1a-the-common-boolean-operators | This video explains the Boolean operators: not, and, or, together with their associated symbols. It also covers the various comparison operators such as equals to, greater than, less than etc. | Free |
| | | Topic 1A Using trace tables to determine the purpose of algorithms | https://student.craigndave.org/videos/gcse-edexcel-topic-1a-using-trace-tables-to-determine-the-purpose-of-algorithms | A vital skill for any programmer is to work out what algorithms are doing without actually having to run them! In this video we look at how to use trace tables to help you do just this. | Free |
| | | Topic 1A Identifying errors and suggesting fixes | https://student.craigndave.org/videos/gcse-edexcel-topic-1a-identifying-errors-and-suggesting-fixes | This video explores the importance of not only being able to identify errors within an algorithm, but also how to suggest possible fixes. | Free |
| | | Topic 1A Bubble sort | https://student.craigndave.org/videos/gcse-edexcel-topic-1a-bubble-sort | The bubble sort is one of three sorting algorithms students need to know for GCSE exams. This video explains the bubble sort technique. | Free |
| | | Topic 1A Merge sort | https://student.craigndave.org/videos/gcse-edexcel-topic-1a-merge-sort | The merge sort is one of three sorting algorithms students need to know for GCSE exams. This video explains the merge sort technique. | Free |
| | | Topic 1A Linear search | https://student.craigndave.org/videos/gcse-edexcel-topic-1a-linear-search | An alternative to the binary search, the linear search methodically checks each item in a list until the one you are looking for is found. This video introduces the algorithm. It is usually less efficient than a binary search, unless the item you are looking for is towards the start of the list. | Free |
| | | Topic 1A Binary search | https://student.craigndave.org/videos/gcse-edexcel-topic-1a-binary-search | An alternative to a linear search, the binary search starts in the middle of a list and removes half of the items from the list until the item you are looking for is found. It is usually more efficient than a linear search. This video introduces the algorithm. | Free |
| | | Topic 1A Algorithm efficiency | https://student.craigndave.org/videos/gcse-edexcel-topic-1a-algorithm-efficiency | More than one algorithm can often be used to solve the same problem, so which one should you choose? In this video we take a look how to compare algorithms for solving the same problem based on their time efficiency. | Free |
| | | Topic 1B Simple logic diagrams | https://student.craigndave.org/videos/gcse-edexcel-topic-1b-simple-logic-diagrams | This video introduces you to the simple logic gates and how to combine them. Although this is not in the EDEXCEL specification it provides an excellent foundation for the later videos in this topic. | Free |
| | | Topic 1B Truth tables | https://student.craigndave.org/videos/gcse-edexcel-topic-1b-truth-tables | This video illustrates how the different permutations of zeros and ones applied to logic gates and their output can be recorded in what is known as a "truth table". | Free |
| | | Topic 1B Applying logical operators in truth tables to solve problems | https://student.craigndave.org/videos/gcse-edexcel-topic-1b-applying-logical-operators-in-truth-tables-to-solve-problems | In this video we take a look at how to apply logical operators in truth tables in order to solve problems | Free |

| Specification reference | Subject content | Where is this covered in Craig'n'Dave resources | Link to resources | What does it include? | Free or paid for |
|--|-------------------|---|---|--|------------------|
| Topic 2: Computational thinking | 2.1 - 2.3 | All of Topic 2 is fully covered as part of a learning journey across the follow CND Structured Learning Record: SLR3: Data representation | https://craigndave.org/pearson-edexcel-gcse-1cp2/ | Our paid for resources include: <ul style="list-style-type: none"> • Starter activities for every lesson • Student workbooks with model answers which act as knowledge organisers • End of topic tests with answers • Terminology guide, short term and long term Schemes of learning & delivery calendars • Lesson PowerPoints for every lesson plus 83 theory PowerPoint presentations • Full set of programming resources, guides, challenges and paper 2 revision material all with solutions • "My little extras pack" contain dozens of extra resources to support teachers | Paid |
| Topic 2: Computational thinking | 2.1 - 2.3 | Smart Revise online revision tool covers all of topic 2 in detail speci point for spec point. | https://smartrevise.craigndave.org/ | <ul style="list-style-type: none"> • 100s of Multiple Choice Quiz questions covering every spec point • Interactive terminology flash cards covering all subject specific terms • Full set of teacher and student analytic reports • Customised personal student revision reports • Topic filtering, revision video playlists and student awards and league tables | Paid |
| Specification reference | Subject content | Topic | Link to Craig 'n' Dave video | Description | Free or paid for |
| Topic 2: Data | 2.1 Binary | Topic 2A Bit patterns | https://student.craigndave.org/videos/gcse-edexcel-topic-2a-bit-patterns | In this video we take a look at the humble bit pattern and understand that any given bit pattern can actually represent many different forms of data. | Free |
| | | Topic 2A Number bases | https://student.craigndave.org/videos/gcse-edexcel-topic-2a-number-bases | In this video we provide an introduction to the three different base number systems you need to know about, these include: Decimal (base-10), Binary (base-2) and Hexadecimal (base-16). | Free |
| | | Topic 2A Signed integers using two's complement | https://student.craigndave.org/videos/gcse-edexcel-topic-2a-signed-integers-using-twos-complement | In this video we take a look at how to represent signed binary integers using the two's complement method. | Free |
| | | Topic 2A Converting between denary and 8 bit binary | https://student.craigndave.org/videos/gcse-edexcel-topic-2a-converting-between-denary-and-8-bit-binary | In this video we explain how to convert between the base ten decimal number system that we are familiar with as humans, and the base two binary system that computers use. This explains how computers store unsigned integers. | Free |
| | | Topic 2A Adding two 8 bit binary integers | https://student.craigndave.org/videos/gcse-edexcel-topic-2a-adding-two-8-bit-binary-integers | This video explains how an arithmetic logic unit can add up to two 8 bit binary numbers together. The electronics needed to create these circuits is explored further at A'level. | Free |
| | | Topic 2A Binary shifts | https://student.craigndave.org/videos/gcse-edexcel-topic-2a-binary-shifts | An arithmetic logic unit is also capable of "binary shifting". That is moving the binary digits in the accumulator to the left or right a given number of spaces. In effect this either multiplies or divides the number by a factor of two, but in reality it is used to access and change individual bits in a series. This is useful when the binary stored is not really a number, but represents something else. This often finds uses in low level programming because micro-controllers and microprocessors rely on the values of individual bits for particular operations. This video provides a brief introduction. | Free |

| | | | | | |
|--|---|---|---|--|-------------------------|
| | | Topic 2A Converting between denary and 2 digit hexadecimal | https://student.craigndave.org/videos/gcse-edexcel-topic-2a-converting-between-denary-and-2-digit-hexadecimal | Large binary number sequences become difficult to manage, so we often use hexadecimal, the base sixteen number system to represent nibbles of data in one symbol. This video illustrates how to convert between decimal and two-digit hexadecimal. | Free |
| | | Topic 2A Why hexadecimal is used in computer science | https://student.craigndave.org/videos/gcse-edexcel-topic-2a-why-hexadecimal-is-used-in-computer-science | This video takes a look at why hexadecimal is often used in computer science | Free |
| | 2.2 Data representation | Topic 2B Representing characters and character sets | https://student.craigndave.org/videos/gcse-edexcel-topic-2b-representing-characters-and-character-sets | A defined list of characters that are recognised and can be stored in a computer system is known as a character set. This video explores the ASCII, extended ASCII and Unicode character sets, explaining their relationship to binary and the number of characters in each set. The need for different character sets and how these have evolved over time is also explained. | Free |
| | | Topic 2B Representing images | https://student.craigndave.org/videos/gcse-edexcel-topic-2b-representing-images | This video explains one technique for how binary can be used to store images. This technique is known as bit-mapping. The smallest part of an image, known as a pixel has a binary code for its colour. This video explains the relationship between the number of pixels, the number of colours and file size of the image. | Free |
| | | Topic 2B Representing sound | https://student.craigndave.org/videos/gcse-edexcel-topic-2b-representing-sound | This video explains how sound can be represented and stored in binary. A number of key terms are defined including sample size, bit depth and sample frequency together with the impact they have on both the quality and the file size of a sound. | Free |
| | | Topic 2B Limitations of binary representation | https://student.craigndave.org/videos/gcse-edexcel-topic-2b-limitations-of-binary-representation | This video explores the limitations of the binary representation of data when constrained by the number of bits available. | Free |
| | 2.3 Data storage and compression | Topic 2B The units of data storage | https://student.craigndave.org/videos/gcse-edexcel-topic-2b-the-units-of-data-storage | In this video we explain why computers use binary to store data. The difference between a bit, nibble, byte, kibibyte, mebibyte, gibibyte and tebibyte is explained. These are known as measurements of capacity, or how much data can be stored either in memory or on secondary storage devices. | Free |
| | | Topic 2B Compression | https://student.craigndave.org/videos/gcse-edexcel-topic-2b-compression | This video explains the need for, and types of compression. Lossy and lossless compression are explored with their advantages and uses. | Free |
| Specification reference | Subject content | Where is this covered in Craig'n'Dave resources | Link to resources | What does it include? | Free or paid for |
| Topic 3: Computational thinking | 3.1 - 3.3 | All of Topic 3 is fully covered as part of a learning journey across the follow CND Structured Learning Records: SLR1: Systems architecture & secondary storage SLR2: Systems software SLR9: Robust and secure programming SLR11: Classification of programming languages | https://craigndave.org/pearson-edexcel-gcse-1cp2/ | Our paid for resources include: <ul style="list-style-type: none"> • Starter activities for every lesson • Student workbooks with model answers which act as knowledge organisers • End of topic tests with answers • Terminology guide, short term and long term Schemes of learning & delivery calendars • Lesson PowerPoints for every lesson plus 83 theory PowerPoint presentations • Full set of programming resources, guides, challenges and paper 2 revision material all with solutions • "My little extras pack" contain dozens of extra resources to support teachers | Paid |

| Specification reference | Subject content | Topic | Link to Craig 'n' Dave video | Description | Free or paid for |
|--|---------------------|---|---|---|------------------|
| Topic 3: Computational thinking | 3.1 - 3.3 | Smart Revise online revision tool covers all of topic 3 in detail speci point for spec point. | https://smartrevise.craigndave.org/ | <ul style="list-style-type: none"> • 100s of Multiple Choice Quiz questions covering every spec point • Interactive terminology flash cards covering all subject specific terms • Full set of teacher and student analytic reports • Customised personal student revision reports • Topic filtering, revision video playlists and student awards and league tables | Paid |
| Topic 3: Computers | 3.1 Hardware | Topic 3A Common CPU components and their function | https://student.craigndave.org/videos/gcse-edexcel-topic-3a-common-cpu-components-and-their-function | This video takes the lid off a central processing unit and explain the key components including the registers, the control unit and the cache. | Free |
| | | Topic 3A Von Neumann architecture | https://student.craigndave.org/videos/gcse-edexcel-topic-3a-von-neumann-architecture | This video explores the first computers that had stored programs in the same memory space as the data required for them. This is known as the von Neumann architecture and is still the foundation principle on which most computers are still built today. We take the lid off a central processing unit and explain the key components including the alu, clock, busses and the control unit. | Free |
| | | Topic 3A The fetch-execute cycle | https://student.craigndave.org/videos/gcse-edexcel-topic-3a-the-fetch-execute-cycle | This video introduces the core purpose of the Central Processing Unit (CPU) and its role in fetching and executing instructions. | Free |
| | | Topic 3A The need for secondary storage | https://student.craigndave.org/videos/gcse-edexcel-topic-3a-the-need-for-secondary-storage | This video explains why a computer needs secondary storage, and explains what is meant by primary and tertiary storage. | Free |
| | | Topic 3A Common types of storage | https://student.craigndave.org/videos/gcse-edexcel-topic-3a-common-types-of-storage | In this video various types of secondary storage devices are introduced including optical, magnetic and solid state. Typical uses of these devices are discussed, explaining the reasons why they are a good choice for different situations. This provides students with a broad knowledge of this topic that can be applied to new scenarios in exams. | Free |
| | | Topic 3A Suitable storage devices & storage media | https://student.craigndave.org/videos/gcse-edexcel-topic-3a-suitable-storage-devices-storage-media | In this video we take a look at the most appropriate storage devices for given applications. We discuss different scenarios you are likely to come across in the exam and look at the advantages and disadvantages of choosing certain media. | Free |
| | | Topic 3A Embedded systems | https://student.craigndave.org/videos/gcse-edexcel-topic-3a-embedded-systems | An embedded system is a computer system with a dedicated function within a larger mechanical or electrical system. This video explains the concept at a depth required for GCSE with some examples of embedded systems that students can use in their own work. | Free |
| | 3.2 Software | Topic 3B The purpose and functionality of operating systems | https://student.craigndave.org/videos/gcse-edexcel-topic-3b-the-purpose-and-functionality-of-operating-systems | This video takes a deeper look at vital piece of system software, the operating system. We take a look at why it is needed and its common functions. | Free |
| | | Topic 3B Operating systems part 1 | https://student.craigndave.org/videos/gcse-edexcel-topic-3b-operating-systems-part-1 | This video explains multi-tasking, memory management and device drivers. | Free |
| | | Topic 3B Operating systems part 2 | https://student.craigndave.org/videos/gcse-edexcel-topic-3b-operating-systems-part-2 | This video explains user and file management. The concept of handling multiple users on one computer with personal settings, and providing an abstraction of the physical file system into logical folders for the user. | Free |
| | | Topic 3B Utility software | https://student.craigndave.org/videos/gcse-edexcel-topic-3b-utility-software | This video explores a small range of typical utility software that could be examined at GCSE. This includes encryption, defragmentation, data compression and backup: both full and incremental. | Free |

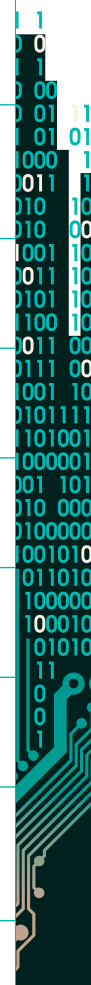
| | | Topic 3B Importance of robust software | https://student.craigndave.org/videos/gcse-edexcel-topic-3b-importance-of-robust-software | This video explores the importance of developing robust software and methods we can use to identify vulnerabilities such as audit trails and code reviews. | Free |
|--|----------------------------------|---|---|--|------------------|
| | 3.3 Programming languages | Topic 3B Characteristics and purpose of different levels of programming language | https://student.craigndave.org/videos/gcse-edexcel-topic-3b-characteristics-and-purpose-of-different-levels-of-programming-language | In this video we take a brief look at the history of programming. We examine the differences between low level languages such as Assembly, and high level languages such as Python. We consider the uses of each of these different types of languages. | Free |
| | | Topic 3B Characteristics of compilers and interpreters | https://student.craigndave.org/videos/gcse-edexcel-topic-3b-characteristics-of-compilers-and-interpreters | This video explains the differences between compilers and interpreters when it comes to translating high level code. They both convert programmers code into binary machine code, but there are advantages and disadvantages to each approach for the user and the execution of the program. | Free |
| Specification reference | Subject content | Where is this covered in Craig'n'Dave resources | Link to resources | What does it include? | Free or paid for |
| Topic 4: Computational thinking | 4.1 - 4.2 | All of Topic 4 is fully covered as part of a learning journey across the follow CND Structured Learning Records: SLR4: Computer networks, protocols and layers SLR5: Network and cyber security | https://craigndave.org/pearson-edexcel-gcse-1cp2/ | Our paid for resources include: <ul style="list-style-type: none"> • Starter activities for every lesson • Student workbooks with model answers which act as knowledge organisers • End of topic tests with answers • Terminology guide, short term and long term Schemes of learning & delivery calendars • Lesson PowerPoints for every lesson plus 83 theory PowerPoint presentations • Full set of programming resources, guides, challenges and paper 2 revision material all with solutions • "My little extras pack" contain dozens of extra resources to support teachers | Paid |
| Topic 4: Computational thinking | 4.1 - 4.2 | Smart Revise online revision tool covers all of topic 4 in detail speci point for spec point. | https://smartrevise.craigndave.org/ | <ul style="list-style-type: none"> • 100s of Multiple Choice Quiz questions covering every spec point • Interactive terminology flash cards covering all subject specific terms • Full set of teacher and student analytic reports • Customised personal student revision reports • Topic filtering, revision video playlists and student awards and league tables | Paid |
| Specification reference | Subject content | Topic | Link to Craig 'n' Dave video | Description | Free or paid for |
| Topic 4: Networks | 4.1 Networks | Topic 4 Types of networks | https://student.craigndave.org/videos/gcse-edexcel-topic-4-types-of-networks | This video explores the two main types of networks: local area networks (LAN) and wide area networks (WAN). The reasons for networking stand-alone computers are outlined, and the potential disadvantages too. The connection options for devices on a network are briefly mentioned and covered in more detail in other videos in this topic. | Free |
| | | Topic 4 How the Internet is structures | https://student.craigndave.org/videos/gcse-edexcel-topic-4-how-the-internet-is-structures | This video explores how the Internet is structured, it covers several concepts including IP addressing and routers. | Free |
| | | Topic 4 Wired and wireless networks and performance | https://student.craigndave.org/videos/gcse-edexcel-topic-4-wired-and-wireless-networks-and-performance | This video explores how the characteristics of wired and wireless connectivity impact on performance, with a focus on speed, range, latency and bandwidth. | Free |
| | | Topic 4 Network speeds | https://student.craigndave.org/videos/gcse-edexcel-topic-4-network-speeds | This with explains how network speeds are measured in bits per second and looks at how to construct expressions involving file size, transmission rate and time. | Free |



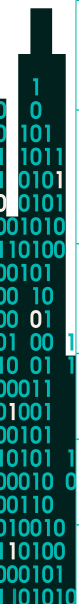
| | | Topic 4 Common protocols | https://student.craigndave.org/videos/gcse-edexcel-topic-4-common-protocols | This video starts by explaining what protocols are as a general concept before moving on to look at some of the common network protocols you will need to know for the exam, these include: Ethernet, Wi-Fi, TCP/IP, HTTP, HTTPS, FTP, POP3, SMTP and IMAP | Free |
|--|-----------------------------|---|---|--|------------------|
| | | Topic 4 TCP-IP protocol model | https://student.craigndave.org/videos/gcse-edexcel-topic-4-tcp-ip-protocol-model | This video takes a deeper look at two of the protocols introduced in the last video, TCP and IP. Together they are known as the TCP-IP 4 layer model. | Free |
| | | Topic 4 Network topologies | https://student.craigndave.org/videos/gcse-edexcel-topic-4-network-topologies | This video examines the topology of networks – how the computers are arranged and connected together. The history of network topologies is outlined briefly, before considering bus, star and mesh networks. | Free |
| | 4.2 Network security | Topic 4 Importance of network security | https://student.craigndave.org/videos/gcse-edexcel-topic-4-importance-of-network-security | This video explores the importance of network security. | Free |
| | | Topic 4 Ways of identifying network vulnerabilities | https://student.craigndave.org/videos/gcse-edexcel-topic-4-ways-of-identifying-network-vulnerabilities | This video examines methods which can be used to help identify network vulnerabilities such as penetration testing and ethical hacking. | Free |
| | | Topic 4 Method of protecting networks | https://student.craigndave.org/videos/gcse-edexcel-topic-4-method-of-protecting-networks | This video examines methods which can be used to help protect networks such as access control, physical security and firewalls. | Free |
| Specification reference | Subject content | Where is this covered in Craig'n'Dave resources | Link to resources | What does it include? | Free or paid for |
| Topic 5: Computational thinking | 5.1 - 5.3 | All of Topic 5 is fully covered as part of a learning journey across the follow CND Structured Learning Records: SLR5: Network and cyber security SLR6: Ethical, legal & environmental issues | https://craigndave.org/pearson-edexcel-gcse-1cp2/ | Our paid for resources include: <ul style="list-style-type: none"> • Starter activities for every lesson • Student workbooks with model answers which act as knowledge organisers • End of topic tests with answers • Terminology guide, short term and long term Schemes of learning & delivery calendars • Lesson PowerPoints for every lesson plus 83 theory PowerPoint presentations • Full set of programming resources, guides, challenges and paper 2 revision material all with solutions • "My little extras pack" contain dozens of extra resources to support teachers | Paid |
| Topic 5: Computational thinking | 5.1 - 5.3 | Smart Revise online revision tool covers all of topic 5 in detail speci point for spec point. | https://smartrevise.craigndave.org/ | <ul style="list-style-type: none"> • 100s of Multiple Choice Quiz questions covering every spec point • Interactive terminology flash cards covering all subject specific terms • Full set of teacher and student analytic reports • Customised personal student revision reports • Topic filtering, revision video playlists and student awards and league tables | Paid |
| Specification reference | Subject content | Topic | Link to Craig 'n' Dave video | Description | Free or paid for |
| Topic 5: Issues and impact | 5.1 Environmental | Topic 5 Environmental impact of computer science | https://student.craigndave.org/videos/gcse-edexcel-topic-5-environmental-impact-of-computer-science | This video explores the environmental impact of computing technologies. This includes the manufacturing of devices, the impact of using limited natural resources, energy consumption, and the disposal of hazardous materials. | Free |

| | | | | | |
|--|------------------------------|---|---|--|-------------------------|
| | 5.2 Ethical and legal | Topic 5 Ethical & legal issues with collection of personal data | https://student.craigndave.org/videos/gcse-edexcel-topic-5-ethical-legal-issues-with-collection-of-personal-data | This video explores the ethical and legal issues associated with the collection and use of personal data, in particular we look at privacy, ownership, consent, misuse and data protection. | Free |
| | | Topic 5 Other ethical & legal issues | https://student.craigndave.org/videos/gcse-edexcel-topic-5-other-ethical-legal-issues | In this video we explore the legal issues associated with the use of artificial intelligence, machine learning and robotics, we pay particular attention to the issues of accountability, safety, algorithmic bias and legal liability. | Free |
| | | Topic 5 Intellectual property protection | https://student.craigndave.org/videos/gcse-edexcel-topic-5-intellectual-property-protection | In this video we explore the methods of intellectual property protection for computer systems and software, paying attention to copyright, patents, trademarks and licencing. | Free |
| | 5.3 Cyber security | Topic 5 Malware | https://student.craigndave.org/videos/gcse-edexcel-topic-5-malware | Malware is a generic catch-all term which covers a large range of malicious software, in this video we take a look at computer viruses, trojans, adware and spyware. | Free |
| | | Topic 5 How hackers exploit technical vulnerabilities | https://student.craigndave.org/videos/gcse-edexcel-topic-5-how-hackers-exploit-technical-vulnerabilities | This video explores how hackers can exploit technical vulnerabilities such as unpatched software and out of date anti-malware. | Free |
| | | Topic 5 Social engineering | https://student.craigndave.org/videos/gcse-edexcel-topic-5-social-engineering | Social engineering is one of the most common ways of exploiting people to give up sensitive information, in this video we take a look at how it can be used to carry out cyberattacks. | Free |
| | | Topic 5 Protecting digital systems and data | https://student.craigndave.org/videos/gcse-edexcel-topic-5-protecting-digital-systems-and-data | This video explores methods which can be used to protect digital systems and data such as anti-malware, encryption, acceptable use policies, backup and recovery procedures. | Free |
| Specification reference | Subject content | Where is this covered in Craig'n'Dave resources | Link to resources | What does it include? | Free or paid for |
| Topic 6: Computational thinking | 6.1 - 6.6 | All of Topic 6 is fully covered as part of a learning journey across the follow CND Structured Learning Records: SLR7: Basic programming concepts SLR8: Advanced programming concepts SLR9: Robust and secure programming SLR10: Algorithms and computational logic | https://craigndave.org/pearson-edexcel-gcse-1cp2/ | Our paid for resources include: <ul style="list-style-type: none"> • Starter activities for every lesson • Student workbooks with model answers which act as knowledge organisers • End of topic tests with answers • Terminology guide, short term and long term Schemes of learning & delivery calendars • Lesson PowerPoints for every lesson plus 83 theory PowerPoint presentations • Full set of programming resources, guides, challenges and paper 2 revision material all with solutions • "My little extras pack" contain dozens of extra resources to support teachers | Paid |
| Topic 6: Computational thinking | 6.1 - 6.6 | Smart Revise online revision tool covers all of topic 6 in detail speci point for spec point. | https://smartrevise.craigndave.org/ | <ul style="list-style-type: none"> • 100s of Multiple Choice Quiz questions covering every spec point • Interactive terminology flash cards covering all subject specific terms • Full set of teacher and student analytic reports • Customised personal student revision reports • Topic filtering, revision video playlists and student awards and league tables | Paid |

| Specification reference | Subject content | Topic | Link to Craig 'n' Dave video | Description | Free or paid for | |
|--|--------------------------------------|---|---|---|--|------|
| Topic 6: Problem solving with programming | 6.1 Develop code | Topic 6A Using abstraction & decomposition to solve problems | https://student.craigndave.org/videos/gcse-edexcel-topic-6a-using-abstraction-decomposition-to-solve-problems | In this video we take another look at how decomposition and abstraction can be used to analyse, understand and solve problems in Python. | Free | |
| | | Topic 6A Algorithmic thinking | https://student.craigndave.org/videos/gcse-edexcel-topic-6a-algorithmic-thinking | This video explores how to take a systematic approach to problem solving, often referred to as "Algorithmic thinking" : getting to a solution by identifying the individual steps needed. An algorithmic way of approaching a word-search is used as an example. The code used is not important in this video, merely the approach of breaking a problem down to achieve a working solution. Advantages algorithmic thinking are discussed in another video, "Decomposition". | Free | |
| | | Topic 6A How to product algorithms using pseudocode and flow diagrams | https://student.craigndave.org/videos/gcse-edexcel-topic-6a-how-to-product-algorithms-using-pseudocode-and-flow-diagrams | This video explores how to convert algorithms written in the form of flowcharts and pseudocode into Python programs. | Free | |
| | | | Topic 6A Program maintenance techniques | https://student.craigndave.org/videos/gcse-edexcel-topic-6a-program-maintenance-techniques | This video looks at various techniques which can be used in a Python program in order to make it easier to read, understand and maintain, such as, layout, indentation, comments, meaningful identifiers and use of white space. | Free |
| | | | Topic 6A How to identify syntax and logic errors | https://student.craigndave.org/videos/gcse-edexcel-topic-6a-how-to-identify-syntax-and-logic-errors | This video takes another look at how to identify logic and syntax errors in Python programs | Free |
| | | | Topic 6A Evaluating a program's fitness for purpose and efficiency | https://student.craigndave.org/videos/gcse-edexcel-topic-6a-evaluating-a-programs-fitness-for-purpose-and-efficiency | This video explores how to use logical reasoning and test data to evaluate a programs fitness for purpose and efficiency by looking at the number of compares, number of passes through a loop and use of memory. | Free |
| | | 6.2 Constructs | Topic 6B The use of data types and casting | https://student.craigndave.org/videos/gcse-edexcel-topic-6b-the-use-of-data-types-and-casting | This video explores the specific data types in Python you need to know for your exam and how to convert between using casting. | Free |
| | | | Topic 6B The use of the three basic programming constructs | https://student.craigndave.org/videos/gcse-edexcel-topic-6b-the-use-of-the-three-basic-programming-constructs | This video explores how to program the three basic programming concepts of sequence, selection and iteration in Python. | Free |
| | | | Topic 6B Using one and two-dimensional structured data types | https://student.craigndave.org/videos/gcse-edexcel-topic-6b-using-one-and-two-dimensional-structured-data-types | This video explores a number of one-dimensional and two-dimensional data structures in Python. | Free |
| | 6.3 Data types and structures | Topic 6B The use of variables, constants and assignments | https://student.craigndave.org/videos/gcse-edexcel-topic-6b-the-use-of-variables-constants-and-assignments | This video explores how to use variables, constants and assignments in Python. | Free | |
| | | Topic 6B The use of basic string manipulation | https://student.craigndave.org/videos/gcse-edexcel-topic-6b-the-use-of-basic-string-manipulation | This video explores the various ways you need to be able to manipulate strings in Python for your exam. | Free | |



| | | | | | |
|--|-------------------------|---|---|---|------|
| | 6.4 Input/output | Topic 6B User input and display output | https://student.craigndave.org/videos/gcse-edexcel-topic-6b-user-input-and-display-output | This video explores how to input data from the keyboard and output it to the screen in Python. | Free |
| | | Topic 6B The user of basic file handling operations | https://student.craigndave.org/videos/gcse-edexcel-topic-6b-the-user-of-basic-file-handling-operations | This video explores how to write programs in Python which read from and write to comma separated value text files. | Free |
| | | Topic 6B Data validation | https://student.craigndave.org/videos/gcse-edexcel-topic-6b-data-validation | This video looks at how you can write programs in Python which implement validation, including length checks, presence checks, range checks and pattern checks. | Free |
| | | Topic 6B Simple authentication routines | https://student.craigndave.org/videos/gcse-edexcel-topic-6b-simple-authentication-routines | This video looks at how to write a simple authentication program in Python. | Free |
| | 6.5 Operators | Topic 6B Using arithmetic, comparison and logic operators in programs | https://student.craigndave.org/videos/gcse-edexcel-topic-6b-using-arithmetic-comparison-and-logic-operators-in-programs | The video looks at the various operators (arithmetic, comparison and logical) which are available in Python. | Free |
| | 6.6 Sub-programs | Topic 6B How to use pre-existing and user-devised subprograms | https://student.craigndave.org/videos/gcse-edexcel-topic-6b-how-to-use-pre-existing-and-user-devised-subprograms | This video looks at how to write programs in Python which use pre-existing built in libraries as well as user-devised subprograms (procedures and functions). | Free |
| | | Topic 6B The Math and Time module | https://student.craigndave.org/videos/gcse-edexcel-topic-6b-the-math-and-time-module | This video explores the functions you are required to know about for the onscreen paper 2 exam from Python's Math and Time modules. | Free |
| | | Topic 6B The Turtle module | https://student.craigndave.org/videos/gcse-edexcel-topic-6b-the-turtle-module | This video explores the functions you are required to know about for the onscreen paper 2 exam from Python's Turtle module. | Free |
| | | Topic 6B Local and global variables | https://student.craigndave.org/videos/gcse-edexcel-topic-6b-local-and-global-variables | This video looks at the difference between using local and global variables, and how this might look in a Python program. | Free |



| Specification reference | Subject content | Unit | Free or paid for |
|--|--|--------------------------------|------------------|
| Topic 1: Computational thinking | 1.1 Decomposition and abstraction | Unit 1: Computational thinking | Paid for |
| | 1.2 Algorithms | Unit 1: Computational thinking | Paid for |
| | 1.3 Truth tables | Unit 1: Computational thinking | Paid for |
| Topic 2: Data | 2.1 Binary | Unit 2: Data | Paid for |
| | 2.2 Data representation | Unit 2: Data | Paid for |
| | 2.3 Data storage and compression | Unit 2: Data | Paid for |
| Topic 3: Computers | 3.1 Hardware | Unit 3: Computers | Paid for |
| | 3.2 Software | Unit 3: Computers | Paid for |
| | 3.3 Programming languages | Unit 3: Computers | Paid for |
| Topic 4: Networks | 4.1 Networks | Unit 4: Networks | Paid for |
| | 4.2 Network security | Unit 4: Networks | Paid for |
| Topic 5: Issues and impact | 5.1 Environmental | Unit 5: Issues and impact | Free |
| | 5.2 Ethical and legal | Unit 5: Issues and impact | Free |
| | 5.3 Cyber security | Unit 5: Issues and impact | Free |
| Topic 6: Problem solving with programming | 6.1 Develop code | Unit 6: Programming | Paid for |
| | 6.2 Constructs | Unit 6: Programming | Paid for |
| | 6.3 Data types and structures | Unit 6: Programming | Paid for |
| | 6.4 Input/output | Unit 6: Programming | Paid for |
| | 6.5 Operators | Unit 6: Programming | Paid for |
| | 6.6 Sub-programs | Unit 6: Programming | Paid for |

| Specification reference | Subject content | Link | Free or paid for |
|--|--|---------------------------------|------------------|
| Topic 1: Computational thinking | 1.1 Decomposition and abstraction | Decomposition - Quiz A | Free |
| | | Decomposition - Quiz B | Free |
| | 1.2 Algorithms | Algorithms - Quiz A | Free |
| | | Algorithms - Quiz B | Free |
| | 1.3 Truth tables | Logic Circuits - Quiz A | Free |
| | | Logic Circuits - Quiz B | Free |
| Topic 2: Data | 2.1 Binary | Binary - Quiz A | Free |
| | | Binary - Quiz B | Free |
| | 2.2 Data representation | Data Representation - Quiz A | Free |
| | | Data Representation - Quiz B | Free |
| | 2.3 Data storage and compression | Storage Q1 | Free |
| | | Storage Q2 | Free |
| Topic 3: Computers | 3.1 Hardware | Hardware Components - Quiz A | Free |
| | | Hardware Components - Quiz B | Free |
| | 3.2 Software | Software Components - Quiz A | Free |
| | | Software Components - Quiz B | Free |
| | 3.3 Programming languages | Programming Constructs - Quiz A | Free |
| | | Programming Constructs - Quiz B | Free |
| Topic 4: Networks | 4.1 Networks | Network Hardware Q1 | Free |
| | | Network Hardware Q2 | Free |
| | 4.2 Network security | Security - Quiz A | Free |
| | | Security - Quiz B | Free |
| Topic 5: Issues and impact | 5.1 Environmental | Environmental Q1 | Free |
| | | Environmental Q2 | Free |
| | 5.2 Ethical and legal | Environmental Q1 | Free |
| | | Environmental Q2 | Free |
| | | Computer Misuse Act Q1 | Free |
| | | Computer Misuse Act Q2 | Free |
| | Digital Divide Q1 | Free | |

| | | | |
|--|--------------------------------------|---|------|
| | | Digital Divide Q2 | Free |
| | | Health and Safety in IT Workplace Q1 | Free |
| | | Health and Safety in IT Workplace Q2 | Free |
| | | Privacy - Quiz A | Free |
| | | Privacy - Quiz B | Free |
| | | System Security Q1 | Free |
| | | System Security Q2 | Free |
| | 5.3 Cyber security | System Security Q1 | Free |
| | | System Security Q2 | Free |
| Topic 6: Problem solving with programming | 6.1 Develop code | How Does Software Get Written - Quiz A | Free |
| | | How Does Software Get Written - Quiz B | Free |
| | 6.2 Constructs | Programming Constructs - Quiz A | Free |
| | | Programming Constructs - Quiz B | Free |
| | 6.3 Data types and structures | Data Types - Quiz A | Free |
| | | Data Types - Quiz B | Free |
| | 6.4 Input/output | Input Devices Q1 | Free |
| | | Input Devices Q2 | Free |
| | 6.5 Operators | Relational and Boolean Operators - Quiz A | Free |
| | | Relational and Boolean Operators - Quiz B | Free |
| | 6.6 Sub-programs | Python Modular Set 1 Q1 | Free |
| | | Python Modular Set 1 Q2 | Free |

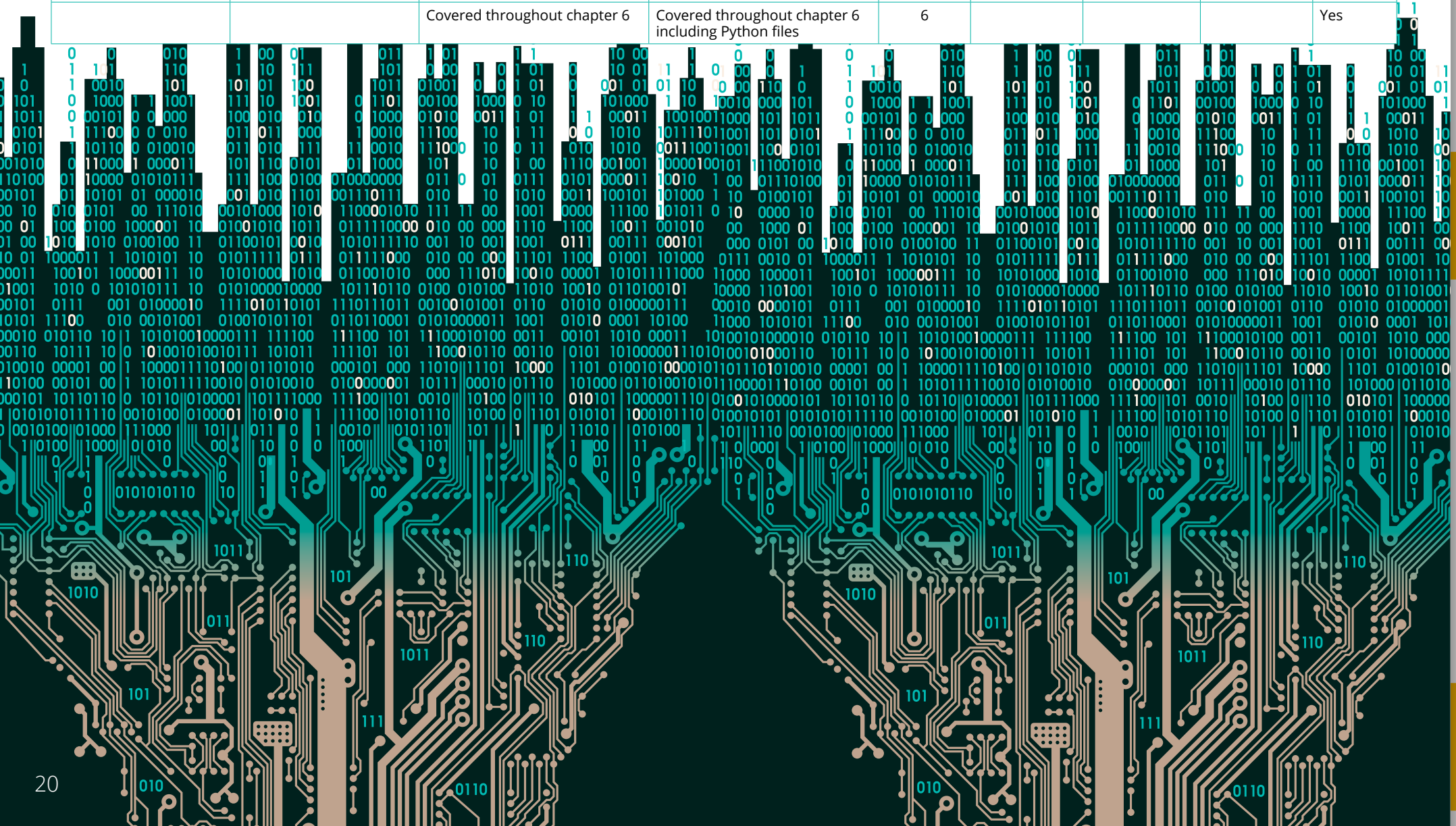
| Specification reference | Subject content | Topics | Sub-topics | Textbook Chapter | Textbook Pages | Presentation (and section) | Slides | Paid |
|--|--|---------------------------------|---|------------------|------------------------------|--|---------------------------------|------|
| Topic 1: Computational thinking | 1.1 Decomposition and abstraction | Representing Algorithms | Algorithms Decomposition Abstraction Pattern Recognition | 1 | 4-6 7-9 10-14 14-17 | 1.1 (a) 1.1 (b) 1.1 (c) 1.1 (d) | 3-18 19-28 29-43 44-55 | Yes |
| | | Subroutines | Decomposition and abstraction in programming | 6 | 113-116 | 6.8 (f) | 46-57 | |
| | 1.2 Algorithms | Subroutines | Benefits of subroutines | 6 | 116 | 6.8 (e) | 39-45 | Yes |
| | 1.3 Truth tables | Representing Algorithms | Problem Solving (Pseudocode) Problem Solving (Flowcharts) | 1 | 18-20 20-25 28-32 | 1.1 (e) 1.1 (f) 1.2 (a) | 56-65 66-83 3-16 | Yes |
| | | Understanding Algorithms | Input, Process and Output | | | | | |
| Topic 2: Data | 2.1 Binary | Covered in 6.3.1 | Variable declaration (initialisation) | 6 | 5 | 6.1 (a) | 3-7 | Yes |
| | | | Constant declaration | | 6 | 6.1 (b) | 8-13 | |
| | | | Assignment | | 7-8 | 6.1 (c) | 14-19 | |
| | | | Arrays | | 75-80 | 6.7 (a) | 3-34 | |
| | | | Lists | | 80-82 | 6.7 (b) | 35-43 | |
| | | | Two-dimensional arrays | | 82-86 | 6.7 (c) | 44-65 | |
| | 2.2 Data representation | Covered in 6.5.1 | Basic arithmetic operators | 6 | 17-18 | 6.3 (a) | 3-8 | Yes |
| Integer division and remainder operators | | | | 18-19 | 6.3 (b) | 9-17 | | |
| Exponent operator | | | | 19-20 | 6.3 (c) | 18-23 | | |
| Mathematical functions | | | | 20-24 | 6.3 (d) | 24-35 | | |
| Random number functions | | | | 25-26 | 6.3 (e) | 36-45 | | |
| Relational operators | | | Covered in 6.5.2 | 70-71 | 6.6 (a) | 3-12 | | |
| Boolean operators | Covered in 6.5.3 | 72-73 | 6.6 (b) | 13-22 | | | | |
| | 2.3 Data storage and compression | Understanding algorithms | Understanding the Purpose of an Algorithm | 1 | 33-36 | 1.2 (b) | 17-23 | Yes |
| | | | Trace Tables | | 37-42 | 1.2 (c) | 24-45 | |
| Topic 3: Computers | 3.1 Hardware | Errors | Introduction | 6 | 147 | 6.12 | 3-8 | Yes |
| | | | Syntax errors | | 147-148 | 6.12 (a) | 9-12 | |
| | | | Logic errors | | 149-150 | 6.12 (b) | 13-18 | |
| | | | Runtime errors | | 151-152 | 6.12 (c) | 19-31 | |
| | | | Interpreting error messages | | 152-153 | 6.12 (d) | 32-36 | |
| | 3.2 Software | Searching Algorithms | Introduction to searching algorithms | 1 | 50 | 1.3 (a) | 3-7 | Yes |
| Linear Search | | | | 50-56 | 1.3 (b) | 8-34 | | |
| Binary Search | | | | 57-62 | 1.3 (c) | 35-51 | | |
| Sorting Algorithms | | | | 73 | 1.4 (a) | 3-6 | | |
| Bubble Sort | | | | 73-80 | 1.4 (b) | 7-33 | | |
| | | | Merge Sort | | 85-87 | 1.4 (c) | 39-53 | |

| | | | | | | | | |
|--|--------------------------------------|--|--|---|--------------------------------------|---|--|-----|
| | 3.3 Programming languages | Understanding Algorithms Searching Algorithms Sorting Algorithms | Evaluating Algorithms Comparing and contrasting search algorithms Comparing and contrasting sort algorithms | 1 | 43-46 69-71 91-92 | 1.2 (d) 1.3 (d) 1.4 (d) | 46-60 68-75 61-66 | Yes |
| Topic 4: Networks | 4.1 Networks | Understanding Algorithms | Understanding the Purpose of an Algorithm Trace Tables" | 1 | 33-36 37-42 | 1.2 (b) 1.2 (c) | 17-23 24-45 | Yes |
| | 4.2 Network security | TBC | | 2 | | | | Yes |
| Topic 5: Issues and impact | 5.1 Environmental | Use OCR J276 Chapter 11 Temporarily 1CP2 Completion by 31st October 2020 | | 2 | | | | Yes |
| | 5.2 Ethical and legal | | | 2 | | | | Yes |
| | 5.3 Cyber security | TBC | | 3 | | | | Yes |
| Topic 6: Problem solving with programming | 6.1 Develop code | Use OCR J276 Chapters 1, 4, 10 Temporarily 1CP2 Completion by 31st October 2020 | | 3 | | | | Yes |
| | 6.2 Constructs | | | 3 | | | | Yes |
| | 6.3 Data types and structures | Network Structures | Introduction to Network Structures Benefits of Networks | 4 | 1 1 | 4.1 (a) 4.1 (b) | 3-5 6-8 | Yes |
| | 6.4 Input/output | Network Structures | Types of Networks | 4 | 2-5 | 4.1 (c) | 9-21 | Yes |
| | 6.5 Operators | Network Structures | The Internet | 4 | 6-10 | 4.1 (d) | 22 -37 | Yes |
| | 6.6 Sub-programs | Transmission Media | Wired Networks Wireless Networks Combining Wired and Wireless networks Network Performance | 4 | 18 19-25 25-26 27-28 | 4.2 (a) 4.2 (b) 4.2 (c) 4.2 (d) | 3-8 9-37 38-43 44-55 | Yes |
| | | Transmission Media | Network Performance | 4 | 29-31 | 4.2 (d) | 56-66 | Yes |
| | | Network Protocols | Introduction to Network Protocols Protocol Families Data transmission and network protocols Application Protocols | 4 | 33 33-34 34-36 36-39 | 4.3 (a) 4.3 (b) 4.3 (c) 4.3 (d) | 3-6 7-11 12-21 22-36 | Yes |
| | | TCP/IP 4-Layer Model | Introduction to TCP/IP 4-Layer Model Application Layer Transport Layer Internet Layer Link Layer | 4 | 41-42 42 43 43 43-44 | 4.4 (a) 4.4 (b) 4.4 (c) 4.4 (d) 4.4 (e) | 3-11 12-14 15-17 18-20 21-24 | Yes |

| | | | | | | | | |
|--|--|--|---|---|--|--|---|-----|
| | | Network Structures | Network Topologies | 4 | 7-16 | 4.1 (e) | 38-36 | Yes |
| | | Network Security | Importance of network security Network security methods Penetration testing | 4 | 46-48 48-51 51-52 | 4.5 (a) 4.5 (b) 4.5 (c) | 3-27 28-45 46-53 | Yes |
| | | TBC | | 5 | TBC | | | Yes |
| | | Use 1CP1 Chapter 6 Temporarily 1CP2 Completion by 31st October 2020 | | 5 | Use 1CP1 Chapter 6 Temporarily 1CP2 Completion by 31st October 2020 | | | Yes |
| | | Subroutines Covered in 1.1.1 | | 6 | 113-115 | 6.8 (f) | 39-45 | Yes |
| | | | Decomposition and abstraction in programming Algorithms Decomposition Abstraction Pattern Recognition | 1 | 4-6 7-9 10-14 14-17 | 1.1 (a) 1.1 (b) 1.1 (c) 1.1 (d) | 3-18 19-28 29-43 44-55 | |
| | | Covered throughout chapter 6 | Covered throughout chapter 6 | 6 | | | | Yes |
| | | Covered throughout chapter 6 | Covered throughout chapter 6 | 6 | | | | Yes |
| | | Maintainability of code | Meaningful identifier names Comments Use of constants Indented code White space Subroutines | 6 | 124-125 126-127 128 129-130 130 131 | 6.10 (a) 6.10 (b) 6.10 (c) 6.10 (d) 6.10 (e) 6.10 (f) | 3-11 12-18 19-22 23-27 28-31 32-35 | Yes |
| | | Errors | Introduction Syntax errors Logic errors Runtime errors Interpreting error messages | 6 | 147 147-148 149-150 151-152 152-153 | 6.12 6.12 (a) 6.12 (b) 6.12 (c) 6.12 (d) | 3-8 9-12 13-18 19-31 32-36 | Yes |
| | | Covered in 1.2.7 | Evaluating Algorithms Comparing and contrasting search algorithms Comparing and contrasting sort algorithms | 1 | 43-46 69-71 91-92 | 1.2 (d) 1.3 (d) 1.4 (d) | 46-60 68-75 61-66 | Yes |
| | | Covered throughout chapter 6 | Covered throughout chapter 6 | 6 | | | | Yes |
| | | Programming Concepts | Sequence (including Time) Selection Iteration | 6 | 49-50 50-55 56-66 | 6.5 (a) 6.5 (b) 6.5 (c) | 3-6 7-24 25-66 | Yes |

| | | | | | | | | |
|--|--|-------------------------------|--|---|---|---|--|-----|
| | | Data types | Character and string Integer Real Boolean Selecting data types | 6 | 10 11-12 12 13 14 | 6.2 (a) 6.2 (b) 6.2 (c) 6.2 (d) 6.2 (e) | 5-11 12-18 19-22 23-28 29-32 | Yes |
| | | Data structures | Casting Arrays Lists Two-dimensional arrays Records | | 14-16 75-80 80-82 82-86 87-90 | 6.2 (f) 6.7 (a) 6.7 (b) 6.7 (c) 6.7 (d) | 33-43 3-34 35-43 44-65 66-74 | |
| | | Working with data | Variable declaration (initialisation) Constant declaration Assignment | 6 | 5 6 7-8 | 6.1 (a) 6.1 (b) 6.1 (c) | 3-7 8-13 14-19 | Yes |
| | | String manipulation | Introduction (Strings) Length Concatenation Slicing ASCII Formatting strings and numbers Positional arguments Case conversion Search functions | 6 | 28 28 28-29 29-31 32 33-41 41-42 42-43 43-47" | 6.4 6.4 (a) 6.4 (b) 6.4 (c) 6.4 (d) 6.4 (e) 6.4 (f) 6.4 (g) 6.4 (h) | 3-5 6-8 9-13 14-21 22-30 31-49 50-52 53-56 57-71 | Yes |
| | | Working with data | Output data to a computer display Obtain user input from the keyboard | 6 | 8 9 | 6.1 (e) 6.1 (f) | 20-24 25-29 | Yes |
| | | Data structures | Text files (including CSV) | 6 | 90-101 | 6.7 (e) | 75-106 | Yes |
| | | Robust and secure programming | Validation routines | 6 | 133-142 | 6.11 (a) | 3-41 | Yes |
| | | Robust and secure programming | Authentication routines | 6 | 143-144 | 6.11 (b) | 42-49 | Yes |
| | | Working with numbers | Basic arithmetic operators Integer division and remainder operators Exponent operator Mathematical functions Random Number functions | 6 | 17-18 18-19 19-20 20-24 25-26 | 6.3 (a) 6.3 (b) 6.3 (c) 6.3 (d) 6.3 (e) | 3-8 9-17 18-23 24-35 36-45 | Yes |
| | | Operators | Relational operators | 6 | 70-71 | 6.6 (a) | 3-12 | Yes |
| | | Operators | Logical operators | 6 | 72-73 | 6.6 (b) | 13-22 | Yes |

| | | | | | | | |
|--|------------------------------|--|---|-----------------------|-------------------------------|----------------------|-----|
| | Subroutines | Procedures Functions covered in 6.6.2 | 6 | 104-106 | 6.8 (a) | 3-14 | Yes |
| | Subroutines | Parameters Functions and return values | 6 | 107 107-110 | 6.8 (b) 6.8 (c) | 15-19 20-33 | Yes |
| | Subroutines | Scope of variables | 6 | 111-112 | 6.8 (d) | 34-38 | Yes |
| | Turtle | Moving and turning Setting positions Turtle properties | 6 | 119 119-120 121 | 6.9 (a) 6.9 (b) 6.9 (c) | 5-8 9-13 14-17 | Yes |
| | Covered throughout chapter 6 | Covered throughout chapter 6 including Python files | 6 | | | | Yes |



| Specification reference | Subject content | Link | Free or paid for |
|--|--|---|---------------------|
| Topic 1: Computational thinking | 1.1 Decomposition and abstraction | | |
| | 1.2 Algorithms | 1. Python Program Design >> | Free to view online |
| | | 2. Coding a Design in Python >> | Free to view online |
| | 1.3 Truth tables | | |
| Topic 2: Data | 2.1 Binary | 1. Number Systems Introduction >> | Free to view online |
| | | 2. Binary Patterns >> | Free to view online |
| | | 3. Binary Number System >> | Free to view online |
| | | 4. Denary to Binary Base Change >> | Free to view online |
| | | 5. Binary to Denary Conversion >> | Free to view online |
| | | 6. Binary to Hexadecimal Conversion >> | Free to view online |
| | | 7. Hexadecimal to Binary Conversion >> | Free to view online |
| | | 8. Finding the 1's Complement >> | Free to view online |
| | | 9. Finding the 2's Complement >> | Free to view online |
| | | 10. Quick 2's complement >> | Free to view online |
| | | 11. Unsigned Binary Numbers >> | Free to view online |
| | | 12. Signed Binary Numbers >> | Free to view online |
| | 2.2 Data representation | | |
| | 2.3 Data storage and compression | | |
| Topic 3: Computers | 3.1 Hardware | 1. Machine Code Instructions >> | Free to view online |
| | | 2. Fetch-Decode-Execute Cycle >> | Free to view online |
| | 3.2 Software | 1. An Introduction to Operating Systems Theory >> | Free to view online |
| 2. Operating System Managers >> | | Free to view online | |
| | 3.3 Programming languages | | |
| Topic 4: Networks | 4.1 Networks | | |
| | 4.2 Network security | | |
| Topic 5: Issues and impact | 5.1 Environmental | | |
| | 5.2 Ethical and legal | | |
| | 5.3 Cyber security | | |

| | | | |
|--|--------------------------------------|---|---------------------|
| Topic 6: Problem solving with programming | 6.1 Develop code | | |
| | 6.2 Constructs | 1. Python Variables and Data Types >> | Free to view online |
| | | 2. Python Assignment Statement >> | Free to view online |
| | | 3. Python Assignment Operators >> | Free to view online |
| | | 4. Python Augmented Assignment Operators >> | Free to view online |
| | | 5. Naming Python Variables >> | Free to view online |
| | | 6. Python's Case Sensitivity >> | Free to view online |
| | | 7. The Sequence Structure >> | Free to view online |
| | | 8. The if Selection Construct >> | Free to view online |
| | | 9. The if... else Selection Construct >> | Free to view online |
| | | 10. Boundary testing the if ... else Selection Construct >> | Free to view online |
| | | 11. Nested Selection Constructs >> | Free to view online |
| | | 12. The elif Selection Construct >> | Free to view online |
| | | 13. The while iteration (repetition loop) > | Free to view online |
| | | 14. Python's Relational Operators (making decisions) >> | Free to view online |
| | | 15. Multiple Conditional Tests (and paths) in a Python program >> | Free to view online |
| | | 16. Python's range function >> | Free to view online |
| | | 17. The for loop iteration >> | Free to view online |
| | | 18. A Python Variable versus a Python Object* | Free to view online |
| | 6.3 Data types and structures | | |
| | 6.4 Input/output | | |
| | 6.5 Operators | 1. Python's Arithmetic Operators >> | Free to view online |
| | | 2. Python's % Operator >> | Free to view online |
| | | 3. Python's logical operators (and, or & not) >> | Free to view online |
| | | 4. Python's logical and operator >> | Free to view online |
| | | 5. Operator Precedence >> | Free to view online |
| | | 6. Multiple Assignment >> | Free to view online |

* This video is outside the scope of the GCSE course but is worth watching to help understand why it is often said that 'Everything in Python is an Object'.

| | | | |
|--|-------------------------|--|---------------------|
| | | 7. Multiple Assignment and Trace Tables >> | Free to view online |
| | | 8. Simultaneous Assignment >> | Free to view online |
| | 6.6 Sub-programs | 1. Python's ord() function >> | Free to view online |
| | | 2. Python's chr() function >> | Free to view online |
| | | 3. Python's print() function >> | Free to view online |
| | | 4. Python's abs() function >> | Free to view online |
| | | 5. Why we have Python functions >> | Free to view online |
| | | 6. Python User Defined Function >> | Free to view online |
| | | 7. How to design a Python function >> | Free to view online |
| | | 8. Python Function with arguments and no return value >> | Free to view online |
| | | 9. Python Function with no arguments but return value >> | Free to view online |
| | | 10. Python Function with no arguments and no return value >> | Free to view online |
| | | 11. Python Keyword Arguments for functions >> | Free to view online |
| | | 12. Python Default Arguments for functions >> | Free to view online |
| | | 13. Local and Global Variable Scope >> | Free to view online |
| | | 14. Lifetime of Variables >> | Free to view online |
| | | 15. In functions do not directly access global variable >> | Free to view online |
| | | 16. Python Parameter Passing Mechanism >> | Free to view online |
| | | 17. Python Exercise 1 Trace Tables>> | Free to view online |
| | | 18. Solution to Python Exercise 1 Trace Tables>> | Free to view online |
| | | 19. Python Exercise 2 Arithmetic Operators and Types>> | Free to view online |
| | | 20. Solution to Exercise 2 Arithmetic Operators and Types>> | Free to view online |

| Specification reference | Subject content |
|---|-----------------------------------|
| Topic 1: Computational thinking | 1.1 Decomposition and abstraction |
| | 1.2 Algorithms |
| | 1.3 Truth tables |
| Topic 2: Data | 2.1 Binary |
| | 2.2 Data representation |
| | 2.3 Data storage and compression |
| Topic 3: Computers | 3.1 Hardware |
| | 3.2 Software |
| | 3.3 Programming languages |
| Topic 4: Networks | 4.1 Networks |
| | 4.2 Network security |
| Topic 5: Issues and impact | 5.1 Environmental |
| | 5.2 Ethical and legal |
| | 5.3 Cyber security |
| Topic 6: Problem solving with programming | 6.1 Develop code |
| | 6.2 Constructs |
| | 6.3 Data types and structures |
| | 6.4 Input/output |
| | 6.5 Operators |
| | 6.6 Sub-programs |

| Specification reference | Subject content |
|---|-----------------------------------|
| Topic 1: Computational thinking | 1.1 Decomposition and abstraction |
| | 1.2 Algorithms |
| | 1.3 Truth tables |
| Topic 2: Data | 2.1 Binary |
| | 2.2 Data representation |
| | 2.3 Data storage and compression |
| Topic 3: Computers | 3.1 Hardware |
| | 3.2 Software |
| | 3.3 Programming languages |
| Topic 4: Networks | 4.1 Networks |
| | 4.2 Network security |
| Topic 5: Issues and impact | 5.1 Environmental |
| | 5.2 Ethical and legal |
| | 5.3 Cyber security |
| Topic 6: Problem solving with programming | 6.1 Develop code |
| | 6.2 Constructs |
| | 6.3 Data types and structures |
| | 6.4 Input/output |
| | 6.5 Operators |
| | 6.6 Sub-programs |

