2021-22 CURRICULUM MAP FOR Computing YEAR 11

HALF TERM 1: **The importance of algorithms in computing** **2.1.1 Computational thinking** **KQ1 – What are the skills required to effectively write algorithms? (2.1.1)** (a) Abstraction -Simplifies the problem / focuses on the important detail -Removing / hiding / obscuring / ignoring unnecessary detail from a problem -Simplifies the problem - reduces complexity, concentrates on details that are needed to solve the problem. -e.g. unnecessary detail could be removed from a driving simulator -This would make the problem easier to solve and reduce the required processing power (b) Decomposition -Breaking down a complex problem or into smaller parts/problems -So that are more manageable and easier to understand. -Smaller parts/problems can be solved individually and are simpler to work with -If a problem is not decomposed, it is much harder to solve. (c) Algorithmic thinking -A way of getting to a solution to a problem through the clear definition of the steps needed -i.e. solving computer related problems by writing algorithms. **KQ1 – How can problems be broken down to make them easier to solve? (2.1.2)** (a) Inputs Identify the inputs required to solve a given programming/pseudocode scenario -Anything which needs to be supplied to the program to meet its goals. -Often input by the user. (b) Processing Identify the processing required to solve a given programming/pseudocode scenario. -Consider the calculations that need to be performed while the program is running. - Consider whether the data needs to change format or data type. (c) Outputs Identify the outputs required to solve a given programming/pseudocode scenario -Consider what the program needs to output -Consider what form this output needs to take -Consider an appropriate variable name and data type for any output. **KQ2 - How can problems be decomposed using structure diagrams?** (a) Structure Diagrams -Help to decompose problems, taking a structured approach. Uses a process known as step-wise refinement.	📚	Use of websites to support the development of programming skills and knowledge
	✏	Written explanation of computer programs that have been written.
	🗣	Peer explanation of computer programs that have been written.
	🔗	Links to ICT – Use of databases Links to maths – Mathematical operations in computer programming.

-Each subsequent level breaks down a problem into smaller parts. e.g. -Level 0 – the main problem. -Level 1 – the main parts needed to solve the problem. -Level 2 – the tasks needed to solve the main parts of the problem. Etc. -Further levels are required where problems are still complex and require breaking down further. -Each lowest level node should achieve a single task. -They can then be coded as a single module/sub-program **KQ3 - How can algorithms be represented using diagrams?** a) Flow diagram uses – -Represents a sequence of steps in diagrammatic form. -Can follow the different paths through the algorithm caused by decisions. (b) Flow diagram symbols – -Terminator – Used to start and end the diagram. Input/output – Used to show where data will be input/output. -Decision – Used to enable the program to take multiple paths. Allows for a single yes/no or true/false decision. -Process/action – Where an action is performed e.g. a calculation. This action is usually a task performed by the computer. -Sub-program – Can be used to represent a larger process, which should be represented in a separate flow diagram. This symbol 'calls' the sub-program. (c) Drawing flow diagrams – Create, interpret, correct, and complete algorithms for real world situations, using flow diagrams. (d) Identify errors Skill - Identify syntax/logic errors in diagrams and suggest fixes. **KQ4 - How can algorithms be represented using text?** a) Pseudocode -Uses English-like words to represent the an algorithm. -Represents a sequence of steps in text form. -Uses the logic of programming code, without the strict rules/syntax. -More generic code that can be applied to any programming language. -There are no established conventions/rules. So OCR Reference Language will be used. (b) OCR Reference Language – -A standard for pseudocode that will be used in exams. (c) Writing pseudocode/OCR reference language – -Create, interpret, correct, and complete algorithms for real world situations, using OCR reference language. **Techniques** -Programming fundamentals**:** Variables, constants, operators(Comparison, Arithmetic, Logical), input/output, sequence, selection (if statements, switch/case statements), iteration (Count-controlled loops, condition-controlled loops), operators (comparison, arithmetic, Boolean)		

-Data types: Data Types, casting -Additional programming techniques: String manipulation, file handling (d) Identify errors Skill - Identify syntax/logic errors in code and suggest fixes. **KQ5 – How can a programmer examine their programming code and detect errors?** (a) Trace Tables -Allows a programmer to examine how a program works -Allow a programmer to locate errors in their code. Particularly logic errors. -Test the accuracy of algorithms. -Examining code, moving through it line by line. -Every variable has its own column. -Records the current state of each variable and each output, as it happens. (b) Creating trace tables -Identify each variable, create own column. -Update the row for each variable when it changes. The next blank row is used. -Skill – Create and use trace tables to follow an algorithm. **1.5.1 Operating systems** **KQ1 - What are the purpose and functions of the operating system?** (a) User interface Provides interface between user and computer / determines look and feel of the computer and how the user interacts with the computer. Command Line – Users enter instructions/commands using the keyboard -Slower, less intuitive, have to remember the commands Graphical User Interface – Options are represented as 'icons' arranged inside rectangular boxes called windows. -Very easy to use, especially for a beginner, very intuitive, needs lots of space/memory to be able to run. (b) Memory management and multitasking Memory management - Transfers programs/tasks into and out of memory (primary and virtual), when they are needed. -There is not enough memory to store all programs at the same time. -Allocates free memory space between programs and keeps track of memory usage. Multitasking - Allows multiple applications to run at the same time -Several programs are stored in the RAM at the same time, however only one is processed at a time by the CPU. -Manages the CPU and which processes/tasks get access to CPU time i.e. the fetch-decode-execute cycle. -The OS transferring programs between memory (primary and virtual) and the CPU, allows this to happen. (c) Peripheral management and drivers -Driver: A small program manage connections with peripherals, allowing the computer and device to communicate.		

-Handles the translation of requests between a device and the computer -Handles data that is sent and received between the computer and the device, and where it is stored, including buffers. -A buffer: temporary storage in memory from which the printer can draw the data at its own pace. (d) User management -Managing users that can access a computer/device. -Perform such tasks as create/allocate user accounts, setting access rights, providing security features such as passwords etc. (e) File management -Manage files that are being stored on secondary storage. -Perform such tasks as naming files, allocating files to folders, moving, saving etc. **1.5.2 Utility software** **KQ1 – What software is used to maintain a computer system?** (a) Purpose -Small programs which allow the user to maintain or manage the computer. -Perform 'housekeeping' tasks e.g. software updates, anti-virus, formatting etc. (b) Encryption software -To enable users to encrypt/decrypt files before sending them over the internet. -To reduce the risk of data interception. (c) Defragmentation Why files become fragmented - -When a file is saved, it is saved in any available space. -This may be space left by a smaller file which has been deleted, previously. -The new file will be split up between the pieces of free space. Defragmentation – -Parts of files on the hard disc drive are moved so that they are stored together. -Which makes opening files quicker / takes less processing. -Empty spaces collected together. - Which makes saving files quicker / takes less processing. (d) Data Compression -To enable user to compress files before sending them over the internet -To reduce file size before emails are sent over the internet, to reduce bandwidth requirements etc.		
HALF TERM 2: **Advanced programming techniques / Computer software** **2.2.3 Additional programming techniques** **KQ1 – How can string values be changed/manipulated?** (a) String Manipulation Skill - Practical use of string manipulation in a high-level language	📚	-
	✏	Explanation of algorithms that have been written using flow diagrams/pseudocode

-The process of changing the length/properties of a string. -Concatenation – Joining multiple strings together. -String length – Returns the length of a string -Substrings – Returns part of a string, splicing it at the specified points. -Uppercase – Converts a string to uppercase. -Lowercase – Converts a string to lowercase. -ASCII Conversion – Converts character into the ASCII number and vice versa. -ASC(…) returns the ASCII number of a character -CHR(…) returns the character for an ASCII number **KQ2 – How can data be saved to secondary storage using Python?** (a) File Handling Skill - Practical use of file handling, including open, read, write, append and close, in a high-level language -Open – Specifies which file is opened, or creates a file, if a file of the specified name does not exist. -Read – Specifies that data is going to be extracted from the open file and to be used by python. -Write – Add data to an existing file. The file is opened and the contents is replaced -Append – Add data to the end of the data already contained within a file. -Close – Closes a file. A file must be closed before any write changes are committed. **KQ3 - How can related sets of data be stored in programming?** (a) Data Storage -Variables are individual pieces of data. Inefficient to store related data in variables as each piece of data needs its own variable. -Text files are sequential (each line) and not structured. Only efficient with small sets of data. -Large sets of data are stored in databases which offer structure. (b) Records -A collection of related fields -Each field is its own variable -Each field in a record can have a different data type -Used to collect together variables that are related to each other -Stored under a single name/identifier. -You can many records which will follow the same template. item1.name = “”, item2.name = “” Creating a record -Define the structure of the records – which fields. -Declare a variable to use with the record structure. -Assign data to a variable in the record. **KQ4 - How can data be selected from a database using SQL commands** (a) Databases -Used to store large amounts of data -Structure in tables -Columns are called ‘fields’	🗣	Explanation of algorithms that have been written using flow diagrams/pseudocode
	🔗	Links to technology – use of flow diagrams to represent a program

-Rows are called 'records' -Queries are used to extract data from a database, usually answering a particular question. (b) SQL Commands -SELECT – Defines the type of query. Specifies that data is being selected from the database. -FROM – Specifies which table data is going to be selected from. -WHERE – Specifies the criteria for which data will be selected. -AND, OR, NOT – Logical operators can be used between WHERE criteria. -Skill - Practical use of writing SQL commands. **KQ5 - How can multiple values be stored together when programming?** (a) Arrays -Similar to a variable but can store more than one data item. -Allocates memory addresses where this data is stored. Data items are stored contiguously (one after another) -As they are stored contiguously, they are fast to search etc. -Data items are accessed by their index. -Index – the position of the item in the list, starting from 0. -Arrays are a static/fixed length data structures*. You cannot change their length after they have been created. *This is not clear from python as when a new data item is added, the list is moved to a new part of memory. -Lists are different to arrays – not contiguous. But for GCSE can be thought of as the same. (b) One-dimensional -Can be visualised to as a list of values. -e.g. arrayname[0] - Skill - Practical use of one-dimensional arrays in a high-level language. (c) Two-dimensional -Can be visualised as a table of values. -Two indexes are required. An index for the row and an index for the column. Both starting at 0. -e.g.arrayname[0,2] / arrayname[row,column] - Skill - Practical use of two-dimensional arrays in a high-level language. **KQ6 - How do subroutines improve the efficiency of programming code?** Skill - Practical use of functions and procedures in a high-level language. -Used when sections of the program need to be re-used or repeated. -A self-contained set of instructions within a program which has been given a name. -The set of instructions can be called/run at any time from elsewhere within the program. -Parameter – a value that is passed to a subroutine, for it to be used within the subroutine when it is run e.g. procedurename(parameters). -Calling – running a procedure. Advantages -Each subroutine can be tested separately		

-Reuse code in a different programs so it is quicker to develop (new) programs -Avoid repetition of code in the same program -Easier to maintain as code is easier to understand/read/shorter -Code is easier to debug as code is shorter -Work can be split up in a team as they can work on a different subprogram at the same time -Allows for abstraction / removes complexity as subprograms do not need to be understood to be used. (a) Procedure A subroutine that performs a specific task. When complete, the subroutine ends and the main program continues from where it left off. (b) Function -Similar to procedures but returns a value. No difference in python/pseudocode other than returning a value. -Return – sending a value back to be used within the main programming code. -Uses the 'return' command word. **KQ7 - How can random numbers be generated using Python?** Skill - Practical use of generating random numbers in a high-level language **KQ1 – How do computers search data? (2.1.3)** -Search for a value within a set of values, following the same steps each time**.** -Skill - Identify the pseudocode for each searching algorithm. (a) Linear Search -Can search for a value regardless of the order of the data set. -Starting at the beginning of the data set, each item of data is examined until the value is found. -Compare the search value to the first value in the list. -If it matches, the value is found. End the search. -Move to the next value and repeat the steps until found. Skill - How to search for a value in a list of values using a linear search. (b) Binary Search -Can search for a value regardless of the order of the data set. -Finds the middle item in an **ordered** list. -Compares the search value to the middle value -If the values are the same, the value has been found. -If the search value is less than the middle value, the middle value and all values to the right are removed. -If the search value is more than the middle value, the middle value and all values to the left are removed. -These steps are repeated until the search value has been found. Skill - How to search for a value in a list of values using a binary search. **KQ7 – How do computers sort data?** -Sort an unordered list of values into order, following the same steps each time -Skill - Identify the pseudocode for each sorting algorithm.		

(a) Bubble Sort -Look at the first number in the list. -Compare the current number with the next number. -If the next number is smaller than the current number, swap the two numbers. If not, do not swap. -Move to the next number in the list and make this the current number. -Repeat until the last number in the list has been reached. -If any numbers were swapped, repeat again from step 1. -If the end of the list is reached without any swaps, the algorithm will stop. Skill - How to sort a set of data into order using a bubble sort. (b) Merge Sort -More complex sort, but highly efficient. -Uses a technique called divide and conquer. -The list is repeatedly divided into two until all the elements are separated individually, this is done by halving. -e.g. xxxxxxxx, xxxx xxxx, xx xx xx xx, x x x x x x x x -Values are then compared in pairs, placed in order and combined. -This is repeated until the list is back together – in reverse of the separation. Skill - How to sort a set of data into order using a merge sort. (c) Insertion Sort -Less complex and efficient than a merge sort, but more efficient than a bubble sort. -Compares values in turn, starting with the second value in the list -If this value is greater than the value to the left, no changes are made. -If it less than the value to the left, it is repeatedly moved left until a value that is less than it is found. -This is then repeated with the next value until the end of the list is reached. Skill - How to sort a set of data into order using an insertion sort.		
HALF TERM 3: **Networks / Data representation / Compression / Legal and Ethical Impacts** **Networks** **KQ1 - What are the different modes of communication on a network? (1.3.2)** (a) Wired Ethernet - -A type of wireless connection used in networking to connect devices to a LAN/WAN e.g. connect devices to your router, connect devices to switches etc. -Bandwidth differs, dependant on type of cable used.	📚	Reading of articles relating to legal, ethical and environmental issues in computing
	✏	Extended answer questions relating to legal, ethical and environmental issues in computing
	💬	Class debate on ethical issues in computing.

(b) Wireless Wi-Fi - -Longer range up 50m indoor, 100m outdoor, dependant on interference. -Connected devices such as laptops to wireless access points, to connect to a network. -Faster than other technologies such as Bluetooth. Bluetooth - -Short range – 10-100 metres, dependant on class. -Connect devices such as phone, to accessories e.g. speakers, earphones etc., to send audio. -Send data between devices such as phones, over a short distance. -Connect accessories to games consoles e.g. control pads etc. (c) Benefits/Drawbacks Wired -More secure than wireless as devices are directly connected and data is sent through a wire. -More bandwidth, as it is faster to transfer data via cable. -Less chance of interference. -Restricted to one location by cable. -More likely that data is sent successfully. Wireless -More freedom to more around as you are not restricted to one location. -Less secure than wireless as data is broadcast through the air and can be intercepted by other devices. -Less bandwidth than transferring data via cable. -Interference by objects e.g. walls, people. -Interference by other signals. -More chance of losing connection. **KQ2 – How files be protected when they are sent between devices?** (a) Encryption -The process of disguising a message so that it cannot be understood by anyone but its intended recipient. -Encryption requires the use of a key. The key is secret as to how the message has been disguised. -A key is needed to decrypt a message and get it back to plain text. -A Caesar cipher is a simple method of encryption. -The cipher works by moving each letter in the alphabet along by a certain number of places. -The key would be the number that the letters have been moved along by. -Most communications sent via the internet are encrypted: purchases made online are encrypted to try to prevent theft of credit card details, documents, such as a spreadsheet emailed to colleagues, satellite TV transmissions are encrypted to prevent users who are not subscribed from watching TV shows **KQ3 – How are computers on a network addressed?** (a) IP addressing -When connected to a network each device is given a unique IP address.	🔗	Links to citizenship/RS - moral/ethical issues Links to mathematics – mathematical calculations to determine the size of a file

-IPv4 - This address consists of four sets of up to three digits, each with a maximum value of 255, which are separated by dots. Takes 32 bits to store. This gives just over 4 billion unique values.

-With the number of Ipv6 addresses running out, Ipv6 was created. This address consists of sixteen sets of up to three digits. Takes 128 bits to store. This gives just over 340 billion billion billion billion unique values.

-When a device wants to send data to another node, it uses the recipient device's IP address.

-A switch on the network knows where the device with this address is and routes the message to it.

(b) MAC addressing

-Media Access Control

-A unique serial number assigned to each network interface controller (NIC).

-Assigned by the NIC's manufacturer and cannot be changed.

-Allows a network to uniquely identify any device, even when the IP of the device changes.

-Wireless access points/routers can grant/deny access to devices using the device's MAC address.

-Consists of a string of hexadecimal numbers e.g. 1A:5B:6H:98:78:35

-If a device has more than one NIC, for e.g. a wired NIC and a wireless NIC, each NIC will have its own MAC address.

KQ4 - How is the communication between devices managed?

(a) Standards

-Manufacturers/organisations agree to a standard to provide rules for areas of computing, including standards of communication (protocols), use of standard connectors e.g. USB, HDMI etc.

-Allows hardware/software to interact across different manufacturers/producers

(b) Protocols

- A (communication) protocol as a set of rules for transferring data.

-Agreed protocols need to be used so that the devices are communicating in the same way.

-TCP/IP (Transmission Control Protocol/Internet Protocol) – Organises how data packets divided/sent, including the information that is added to packets e.g. sender IP address, recipients IP address, sequence number etc.

-HTTP (Hyper Text Transfer Protocol) - Transfers web pages from the web server to the client's computer, to be viewed in the web browser.

-HTTPS (Hyper Text Transfer Protocol Secure) - Transfers web pages from the web server to the client's computer, to be viewed in the web browser, using a secure/encrypted connection.

-FTP (File Transfer Protocol) - Transmitting a file over the internet, from a client to a server or vice versa.

-POP (Post Office Protocol) – Accessing emails. Downloads emails from a provider's mail server to a device, and deletes them from the server.

-SMTP (Simple Mail Transfer Protocol) - Sending an email from one mail server to another. -IMAP (Internet Message Access Protocol) - Accessing emails. Views emails which are on the server. The emails exist on the server until you delete them. (c) The concept of layers -A layer is a group of protocols which have similar functions. Each layer has its own purpose and is self-contained. -Each layer handles a different part of the communication. -Protocols are self-contained. Protocols in each layer can do their job without needing to know what is happening in other layers. -Layers are not physical things inside a computer or network. They're just a way of categorising protocols, of making them easier to think about/work with. Advantages - -Layers can be changed/removed without the other layers being affected. -Having set rules for each layer forces companies to make compatible, universal software, so different brands will work with each other and always working basically the same way. -This can be referred to as the Transmission Control Protocol/Internet Protocol (TCP/IP) model. (d) 4 Layer Model Application layer - encodes/decodes the message in a form that is understood by the sender and the recipient. Transport layer - breaks down the message into packets, giving it a packet number etc. This data is used to reassemble the packets. Network layer - adds the sender's IP address and that of the recipient. The network then knows where to send the message, and where it came from. Data link layer - enables the transfer of packets between nodes on a network, and between one network and another. **1.2.4 Data storage** **KQ1 – How is text represented in a computer system? (1.2.4)** (a) Character representation - -Each character is represented using a unique binary number. -Different character sets use a different number of bits to represent a character. The more bits used, the more unique values available, which means more characters can be represented. -An increased number of bits per character means each character would take more bits to store, making documents bigger. -The codes are logically ordered e.g. A = 01000001, B = 01000010 (b) Character Sets -The characters / symbols a computer can use / understand / display -A computer can represent characters from whichever character set is installed. (b) ASCII –		

-We use the ASCII character set. -Only represents English characters/numbers. -8 bits but only makes used of 7 with a leading 0. -This allows for 127 (128 with 00000000) characters. (c) Extended ASCII -Used to represent languages with more characters e.g. European languages. -8 bits and makes use of all 8. This allows for 255 (256 with 00000000) characters. (d) Unicode -Used to represent all living languages in the same character set. -16 bits are used. -This allows for 65,536 characters **KQ2 – How are images represented in a computer system?** (a) Image Representation - -Images are divided into pixels. -A pixel is set to a colour. -A pixel is only ever a single colour. -Each colour is given a unique binary value. -The binary value for the colour of each pixel needs to be stored. (b) Colour depth: -Colour depth is the number of bits used to store/represent each pixel. -Each colour requires its own unique value. -If more colours are used, move unique values are needed. -1 bit, 2 colours, 1 and 0. 2 bits, 4 colours, 00, 10, 01, 11 etc. -A bigger colour depth increases the size of the file as more data needs to be stored. (c) Resolution: -Resolution is how clearly each pixel can be seen / the number of pixels in an inch of space (Dots Per Inch). -Higher resolution means more pixels need to be stored, which means more binary digits need to be stored. This increases file size. -As pixels are harder to see, the quality of the image improves. (d) Meta Data -Additional that is stored about an image file e.g. dimensions / height / width / no. of bits per pixel / colours used / location / date / file type -To enable the computer to interpret the binary data that is stored for the image. **KQ3 – How is sound represented in a computer system?** (a) Sound representation - -Sound waves must be stored in binary to be represented in a computer. -The process of 'digitising' a sound is called sampling. -The amplitude of sound waves is measured. -Measurements take place thousands of times a second – 44000 for CD quality audio. -Measurements are stored as binary values. (b) Sampling Rate/Sampling Interval -The number of samples per second (in hz) is called the sampling rate.*		

-The duration between samples is the sampling interval *(both are different names for the same concept) -More samples per second means a bigger file size, as there is more binary to store. -A higher sampling rate means a better quality sound, as sounds are measured more accurately. (c) Bit Depth The number of bits available to store each sample that is taken. Measured in bits -A higher bit depth means a bigger file size, as there is more binary for each sample. -A higher bit depth means a better quality sound, as samples are represented more accurately. (d) Duration -The number of seconds in the audio file. -A longer sound would mean more binary has to be stored. -Adjusting the length of a sound will change the size of its file. **1.2.3 Units** **KQ2 – How is the size of files/capacity of devices measured?** (c) Calculation of file size -Calculate the size of a sound file from given properties, using the formula: sound file size = sample rate x duration (s) x bit depth -Calculate the size of an image file from given properties, using the formula: image file size = colour depth x image height (px) x image width (px) -Calculate the size of a text file from given properties, using the formula: bits per character x number of characters **1.2.5 Compression** **KQ1 – How can the size of files be reduced, if required?** (a) The need for compression -To reduce the size of a file. -Transferring files over a network/the internet to send them quicker/use less bandwidth. -Attaching files to an email where there is a file size limit -Streaming video over the internet. (b) Lossy compression -When the file is compressed some detail / data / quality / resolution is permanently removed. -The compressed file is not identical to the original but the difference may not be noticeable to humans. -The higher the compression, the more noticeable the reduction in quality. -The file size will be reduced. -Can achieves higher compression / smaller file size than lossless -Often used for compressing images, video, sound, where a small loss of quality -Cannot be used for the compression of documents were they would be effected by the removal of data e.g. text files. (c) Lossless compression		

-Lossless compression reduces the size of a file without any damage to the file or reduction in quality.
-The file can be decompressed to its original state, with all the data bits reconstructed.
-Cannot achieve the same levels of compression as lossy compression.
-Ideal for compressing text or numeric files where a loss of data is unacceptable.

1.6.1 Ethical, legal, cultural and environmental impact

KQ1 - How does digital technology impact on the wider society?
(a) Morals –
-An individual's standards of behaviour/principles of what is right and what is wrong.
-Unauthorised access – should not access files without permission/use them to cause harm.
-Unauthorised use of software - should not use software that you have not purchased etc.
-Inappropriate behaviour - should behave 'correctly' when using the internet, social media etc.
-Inappropriate/offensive content - should not create/distribute inappropriate content/images/videos.
-Freedom of speech – The internet gives individuals a way of being heard by millions of people. They should act responsibly.
(b) Ethics –
-Principles that govern a person's behaviour – given by an organisation e.g. ethical code of practice.
-Ensuring public safety - ensuring software that is 'safety critical' is robust and free from errors.
-Data security - companies should ensure that individual's data is stored securely and only used for the correct purposes.
-Environmental Issues - using of goods that are harming the planet and companies that encourage this.
-Ethical sourcing of goods - using of goods/materials that are sourced from child/forced labour.
-Artificial Intelligence - computers can make 'decisions' for themselves, is this right?
-Privacy - administrators/organisations have access to individual's data, how should they act?
-Fake news – do organisations have the responsibility to control the spread?
-Safety issues - Ensuring public safety is paramount. As new technologies are introduced, they bring safety concerns e.g. driverless cars and automated decision making etc.
(c) Cultural issues
The digital divide -
-The digital divide is the division that between those who have/can use technology and those who cannot.
-Age - Younger people embrace and can use technology confidently / older generations avoid/fear it and feel excluded.
-Prosperity level - those who can afford technology and those that cannot. Not everyone can afford the latest smartphone/console.

-This is a national and international issue. The UK is prosperous, but in many countries access to computers is expensive/unaffordable. -Location - Cities tend to have access to high-speed broadband, rural or remote areas often do not -Ability – those who have good computer skills and those who do not. Many employers require good IT skills, and jobseekers who lack those skills may find it difficult to gain employment. (d) Environmental issues Natural Resources - -Devices are made of lots of raw materials, materials are running out -Lots of precious metals are being used in devices: gold, silver, copper, mercury, etc. -Mining resources uses lots of energy and causes pollution E-Waste - -Electronic waste creates 20-50 million tonnes a year. -Devices are designed to be short lasting, to encourage people to upgrade early/quickly. -Devices are made cheaper to replace rather than repair. -Lots of waste is often sent to developing countries and not recycled or just thrown away. -This causes further pollution and some of the metals can be toxic. Energy - -The manufacture of devices takes up a great amount of energy - coal, oil and gas, which causes pollution. -Charging devices also consumes power. -Computers and servers generate heat, therefore then need to be cooled, which consumes power. -Leaving devices idle also uses power, when we aren't using them! Positive Effects - -Development in green technologies. -More communication being done over the internet, rather than physically sending or travelling. -Less physical products which items that are digital e.g. games, films etc. Which means less waste and energy to produce. (e) Privacy issues Privacy – -Users enter data into websites, stored data digitally, they have a right privacy. -Search engines collect a lot of information about its users and their internet usage. -Identity theft - millions of people are victims of identity theft which leads to financial loss or even legal problems. -Tracking - every website you visit uses cookies, these can be used to track internet usage. -All data flows through your ISP. They have access to everything that you visit. -Should the authorities be able to actively monitor internet usage? -Extremism, exploitation etc. -Should the internet be regulated? Yes- -No different to any other type of media,		

-Harmful, offensive and illegal content is created -Children are not well protected -Should organisations making profit from the internet should help police what they have created. No- -Freedom of expression is a right. -The internet is different to other media outlets as anyone can broadcast -The internet is growing too quickly to be effectively filtered -Censorship is bad for democracy **KQ2 – Which pieces of legislation affect how we use computer systems?** (a) The Data Protection Act 2018 -The purpose of the act is to ensure that data held about an individual is used in the correct way and that they have rights. -The 8 data protection principles - Fair and lawful, specific for its purpose, adequate and only for what is needed, accurate and up to date, not kept longer than needed, take into account people's rights, kept safe and secure, not be transferred outside the EEA. -Before companies collect data they must register with the Information Commissioners Office -Individuals may be entitled to compensation if companies use your their incorrectly. (b) Computer Misuse Act 1990 -The purpose of the act is to ensure that it is illegal to use a computer to cause harm, to reduce cybercrime. -What contravenes the act – accessing data without permission (hacking), editing data without permission, accessing data with the intent of committing further crime, making, supplying or obtaining anything which can be used in computer misuse offences (viruses, spyware etc.) (c) Copyright Design and Patients Act 1998 – -The purpose of the act is to ensure that an individual has the right to control the ways in which their material can be used. -Protect intellectual property – so anything that someone has created. -It's illegal to share/use without owner's permission. -Types of work covered - literary, dramatic, musical, artistic, typographical, sound recordings, films -Each type of work is protected for a different duration - literary, dramatic, musical, sound recording etc. -Patents protect the use of new inventions – ideas and concepts – not actual content (d) Impacts of digital technology on society -Stakeholder – -Key stakeholders – Customers, staff, the company/shareholders etc., the community -How individuals may be affected by using technology/key impacts - Increase profits, increase productivity, loss of jobs, less 'personal' service, 27/7 access, less outgoings buildings, staff, bills. **KQ3 – What governs the way in which software can be used?**		

Software license - A document that provides legally binding guidelines for the use and distribution of a piece of software e.g. the terms for use, purpose, ability to make copies etc. (a) Open Source Open source software is software that is free of copyright. -Examples: Linux OS, Firefox browser, Python programming language, Open Office Advantages: -Usually available for free. -Provided with the source code so that anyone can modify the software for their own purposes. -Any modified version, must also be made freely available for anyone else to use or adapt. -It can have many authors. Enabling programmers to contribute to the development of a program over time, refining, improving it and adding extra features. -Problems can be solved by the 'open source' community. Disadvantages: -No guarantee that it works properly as no requirement for anyone to ensure it is bug free. -Support might not be readily available, especially if the program is not in widespread use. -Not provided with warranty etc. (b) Proprietary -Software that is copyrighted, which means it can only be obtained by paying for a licence. The license restricts the copying / modifying / distribution of the software. Advantages: -The product should be free of bugs and has been well tested. -If bugs are found, updates known as patches are often provided free of charge. -Support can be sought from the organisation who supplied the software if problems occur. -Feature updates which extend the software's facilities are often available, although usually at a cost. -Provided with a warranty/often has support available from many sources. -Cheaper for companies, rather than developing their own. Disadvantages: -Can be expensive to buy initially or an ongoing (subscription) cost. -Proprietary software is distributed only as a compiled program / source code not available -Software may not meet the needs of the user/cannot be adapted to meet the needs of the user. -Companies might not maintain older software, so people will buy the latest versions.		
HALF TERM 4: Programming Language / Building robust programs **2.5.1 Languages** **KQ1 What are the different programming languages available to programmers?** (a) High-level languages -Human oriented languages written by programmers	📚	Of the history/development of programming languages.
	✏	Explanation/justification of use of the different techniques to make a computer program robust.

-Contains words for commands which are close to the English language -Machine independent / portable to different systems -Needs to be translated before it can be executed. -Allow the programmer to focus on what needs to be done, rather than how the computer actually works. -One (high level) command equates too many machine code instructions. -Easier to understand and less complex than machine code. -Easier to debug/sport errors as written using English type words. -Faster to code as the programmer can write in natural language (b) Low-level languages -Written as binary instructions -Code that is directly executed by the CPU -Instruction set - the set of instructions that the processor understands -Each CPU has its own instruction set, so machine code is not portable to a different CPU -Not readily understandable by humans so it takes longer to program and spot errors. -Does not need to be translated for the CPU to run it. **KQ2 - How can a programmer convert between different programming languages?** (a) Translators -Software that convert high level language to binary / machine code -The processor can only understand machine code (b) Interpreters -Translates and executes source code one line at a time -Source code – the code that the programmer writes in high level language. -Stops translating when it finds a syntax error and shows the error location -Quicker to re-interpret than recompile the code. -Used to in development to the program / to find errors. -Code most be interpreted each time it is run as it is run immediately and not stored. Do not optimise code - the translated code is executed as it is. (c) Compilers -Translates all of the code in one go -Creates an executable file/object code which can be run unassisted at any time. -Executable file - a file that is ready to run by the CPU. -Compiler report all errors at the end of compilation as an error report. -Compilers optimise code. Optimised code can run quicker and take up less memory space. -Used at the end of development when programmers want to distribute the software, as cannot be easily modified -The source code must be re-compiled every time the programmer changes the program. **2.5.2 The Integrated Development Environment (IDE)**	💬	Peer discussion of where students have used robust programming techniques within their programming tasks.
	🔗	-

KQ1 - Which tools and facilities do programmers use to write computer programs? Skill - Practical experience of using a range of these tools within at least one IDE (a) Integrated Development Environment -Software that a programmer uses to develop programming code -Contains tools that help a programmer develop code. (b) IDE Tools: Editors -Allows the programmer to enter and save programming code. -Colour codes keywords, making the code easier to read. -Auto-completes code as the programmer types, saving time. -Checks the syntax of code and autocorrects/highlights errors. -Bracket matching highlights when a bracket is missing, saving time looking for missing brackets. (c) IDE Tools: Error diagnostics -Highlights errors/displays information about syntax errors, including the error location -Suggest solutions of how the programmer can correct the error. (d) IDE Tools: Run-time environment -Special software that allows a program to run on a computer, regardless of the hardware it is running on. -Allows the programmer to run/test programming code. (e) IDE Tools: Translators -Contains an interpreter to run/convert the code to machine code, line by line. -Stops running code when it encounters an error, which allows the programmer to find errors easily. **2.3.1 Defensive design** **KQ1 How can the design of a program ensure that it always runs as expected?** -Defensive design - Ensure that a program runs correctly no matter what actions a user takes. -Planning for all possibilities, thinking about what a user may do that the program does not expect. (a) Anticipating misuse -Anticipating how a user may misuse a program, accidently or maliciously, and preventing the action from having an effect. -Programmers need to be able to protect their programs from misuse, so should design them accordingly. -Protection against unexpected user inputs or actions. -Input sanitisation - Checks data that is entered and removes anything that might be potentially dangerous e.g. an SQL injection attack. (b) Planning for contingencies -Ensuring that there is a 'back up plan' in case software does not function as expected. -E.g. communication error over the internet – the transfer of data should resume or notify of connection loss.		

E.g. interacting with peripherals – it should not just be assumed that the document has printed etc. -E.g. read/write error when interacting with secondary storage – retry or notify the user of disk error. In all cases, should not just be 'lost'. **KQ2 - How can programmers ensure a system is accessed by the correct users?** (a) Authentication -The process of confirming the identity of a user. -Improves the security of a system and prevents unauthorised access -Username/password – Can be guessed/forgotten. -Biometrics – Cannot be guessed/needs to be present. -reCAPTCHA – Verifies a human is present but not who you are. -Skill - Practical experience of designing simple authentication using programming code/pseudocode. **KQ3 - How can we ensure that the correct data is entered into a program?** (a) Validation -Checks if data meets certain criteria/rules, when it is entered into a program -Takes place before the data is processed by the program. -Data is rejected if it doesn't meet the rules, as incorrectly input data may crash a program. Skill - Practical experience of designing simple input validation using programming code/pseudocode. (b) Validation Checks Range check - The input must fall within a specified range. Usually applied to numbers and dates. Length check - The length of the input must be over/under a certain number. Presence check - A value must be entered. Format check - The data must be in the correct format, e.g. a date DD/MM/YYYY. Type check - The data must be of a specified data type, such as an integer. **KQ4 - How can coding be produced to aid maintainability?** (a) Use of sub programs -Sub programs are created to perform specific tasks/section code, making it easier to follow what the code is doing. -Can reduce the number of lines in a program. (b) Naming conventions -A set of rules for choosing the naming of identifiers e.g. variable names etc. -Well-chosen identifiers make it easier for developers to understand what a is doing -e.g. a = b * c vs weekly_pay = hours_worked * hourly_pay_rate; (c) Indentation -To show where constructs / sections start and finish e.g. Code within selections or iterations should be indented. -Helps other programmers to see the flow of your program clearly. (d) Commenting		

-Lines in programs that explain the key functions/sections / what the different parts of the program do. -Not executed when the program is run – they are ignored. -Preceded by a symbol to indicate where a comment starts. Skill – Be able to apply commenting appropriately within programming/pseudocode. **2.3.2 Testing** **KQ5 - Why is it important that programs are thoroughly tested?** (a) Purpose -To correct any errors made during code development – find 'bugs'. -Some programs are 'critical systems'. They can be life or death. -To ensure that the code functions correctly/as intended. (b) Iterative Testing -Carried out while a program is being developed. -The programmer writes a section of code (module) then tests it. -The code may work fine, if not the programmer will amend or fix the code, and test it again. -The process repeats until the module works as intended (c) Final/terminal -Carried out when all modules are complete/production of the program is complete. -Undertaken before the software is distributed. -The program is tested as a whole to ensure that it functions as it should / meets the initial requirements. **KQ6 - What are the different types of errors that programmers must identify/overcome?** (a) Syntax errors -The interpreter does not understand something has been entered. -Because it does not follow the grammar/rules of the programming language. -The interpreter stops the code from being run/translated. (b) Logic errors -Errors which produce unexpected output. -The error does not prevent program from running. -It does not do what the programmer intended. **KQ7 - Which is it important to Select and use suitable test data?** (a) Test Plans -Test data/plan – The data/tests which are going to be used to test a program. -Should be defined before the program is created. E.g. description of the test, the test data being used, the type of test, expected outcome, actual outcome. Skill - Identify suitable test data for a given scenario Skill - Ability to create/complete a test plan (b) Types of test data -Normal - Data which should be accepted by a program without causing errors		

-Boundary - Data of the correct type which is on the very edge of being valid -Invalid - Data of the correct type but outside accepted validation limit -Erroneous - Data of the incorrect type which should be rejected by a computer system		
HALF TERM 5: Re-teaching of key topics, as identified from October and February mock examinations.	📚	N/a
	✏	N/a
	🗣	N/a
	🔗	N/a