

Scheme of Work 2020 - 2021
Subject: Computer Science

Year Group: 11

Specification: AQA GCSE Computer Science

Lesson No	Topic & Objectives	Big Question – What will students learn?	Key Activities & Specialist Terminology (Do Now Task / Starter/Tasks/Plenary)	Planned Assessment	Homework or flipped learning resources DODDLE resources	Lit Num SMSC Codes
	Project released 1st September 2020.	Unsure until September	NEA Project – Students are required to complete a in class assessment – this document is completed in controlled conditions and must not leave or be discussed outside of the classroom	9th March to 20th March 2021	Units 2A and 2B to support NEA Project.	C1 C2 C3 Sp3 Sp4 Sp7
W.B. 3 rd Sept	Hardware and software	<p>Define the terms hardware and software and understand the relationship between them.</p> <p>Explain what is meant by systems software and application software and be able to give examples of them.</p> <p>Understand the need for and functions of the OS and utility programs.</p>	<p>This is very much a theory topic so is probably best delivered by the teacher talking and discussing with the class.</p> <p>For the first point, students simply need to know that hardware is that the electronic or electro-mechanical components of the computer and that software are the programs that run on the hardware and tell it what to do to perform a task.</p> <p>Students need to know that application software completes user-oriented tasks that the user would need to do with or without a computer whereas system software performs tasks related to the management of the computer system.</p>	<p>Purple book –End of Topic Assessment</p> <p>Asking appropriate questions in CATs</p> <p>Minute Paper</p>	<p>Iteration Homework</p> <p>Research on keywords and definitions</p> <p>Documented problem solutions</p> <p>senecalearning.com</p> <p>doddlelearn.co.uk</p>	C1 C2 C3 Sp3

			<p>Students need to know that the OS manages processor(s), memory, I/O devices, applications and security but do not need to know how.</p> <p>A utility is a program that helps manage a computer but is not core to its operation eg a compression program, a virus-checker. It might be useful to make students aware that utilities are increasingly being bundled with the OS.</p>			
W.B. 7 th Sept	Networks	<p>Students should be able to explain what a computer network is, discuss risks and benefits of networks and the relative merits of wired and wireless networking</p> <p>Students can describe LAN, WAN and PAN and understand star and bus topologies, including their relative merits</p>	<p>Students will have direct experience of using networks, both wired and wireless, so this makes a good discussion topic – pros and cons of having a network and also of wired vs wireless networks.</p> <p>Differences between LAN and WAN should be considered in terms of size, ownership and the hardware used.</p> <p>Topologies are best visualised; it is worth noting that physical bus networks have limited applications nowadays.</p> <p>This topic can be taught as a discussion or there are many online videos and resources.</p>	<p>Purple book –End of Topic Assessment</p> <p>Asking appropriate questions in CATs</p> <p>Minute Paper</p>	<p>senecalearning.com</p> <p>Research on: -Video showing use of MAC address whitelist -Very short video on firewalls</p> <p>Iteration Homework</p> <p>Documented problem solutions</p> <p>doddlelearn.co.uk</p>	<p>C1 C2 C3 Sp3 Sp4 Sp7</p>
W.B. 14 th Sept	Protocols	<p>Define the term ‘network protocol’.</p> <p>Explain the purpose and use of common network protocols including: Ethernet, Wi-Fi, TCP, UDP, IP, HTTP, HTTPS, FTP, SMTP, IMAP.</p>	<p>This topic is a fairly theoretical one. Students could use textbooks, online notes or videos to learn from.</p> <p>They need to understand why a stack is used (abstraction), what the four layers are and some functions of each layer of the stack and at which layers the listed protocols work.</p>	<p>Purple book –End of Topic Assessment</p> <p>Asking appropriate questions in CATs</p> <p>Minute Paper</p>	<p>senecalearning.com</p> <p>doddlelearn.co.uk</p>	<p>C1 Sp2 Sp3</p>

		<p>Understand the need for, and importance of, network security.</p> <p>Explain the following methods of network security: authentication, encryption, firewall, MAC address filtering.</p>				
W.B. 21 th Sept	TCP Layers	Students should know what the four layers are and some functions of each layer, together with which of the protocols listed operate at which layer.	<p>This topic is a fairly theoretical one. Students could use textbooks, online notes or videos to learn from.</p> <p>They need to understand why a stack is used (abstraction), what the four layers are and some functions of each layer of the stack and at which layers the listed protocols work.</p>	<p>Purple book –End of Topic Assessment</p> <p>Asking appropriate questions in CATs</p> <p>Minute Paper</p>	<p>Video tutorial - Youtube senecalearning.com</p> <p>doddlelearn.co.uk</p>	<p>C1 C2 C3 Sp3</p>
W.B. 28 th Sept	Cyber Security	Be able to explain cyber security and the cyber security threats covered by the specification.	<p>This topic works well as a class discussion as most students will be familiar with some of these topics from their own personal experiences.</p> <p>Students could make a presentation, each focusing on one or more topics.</p>	<p>Purple book –End of Topic Assessment</p> <p>Asking appropriate questions in CATs</p> <p>Minute Paper</p>	<p>Documentary on cybercrime in the UK</p> <p>Five of the worst computer viruses</p> <p>Notes on some topics of computer security senecalearning.com</p> <p>doddlelearn.co.uk</p>	<p>M8 C1 C2 C3 Sp3</p>

W.B. 5 th Oct	Protection	Be able to describe methods that are suitable for protecting from cyber security threats	This topic works well as a discussion, as students will be aware of some of these topics from their own experiences. They may need to be focused somewhat to ensure that they cover all of the topics on the specification.	Purple book –End of Topic Assessment Asking appropriate questions in CATs Minute Paper	Novalabs cyber security protection game senecalearning.com Cyber security threats and solutions doddlelearn.co.uk	M8 C1 C2 C3 Sp3 Sp4 Sp7
W.B. 12 th Oct	Security Measures	<p>Students understand what hacking is, what the dangers of hacking are and how cyber security measures can be used to prevent it.</p> <p>Students understand what cloud storage is and its advantage and disadvantages in comparison to local storage.</p> <p>Students understand that cracking can be used as an alternative name for hacking or to mean breaking copy protection of software.</p>	<p>Students should consider what hacking is and the motivation for it. A look at some simple hacking techniques might be of interest but is not required.</p> <p>This topic should be linked into cyber security and the measures that can be taken to prevent hacking.</p> <p>The issue of hacking by governments and whether this can be justified and under what circumstances could also be looked at.</p> <p>The advantages and disadvantages of cloud storage should be considered. This would probably be best achieved in the context of a real cloud service that students' might have used. Examples of security breaches of such services could be examined.</p> <p>Cyber security methods are considered in greater detail in specification section 3.6.</p> <p>This topic could be tackled by students doing individual research and then having a class discussion.</p>	Purple book –End of Topic Assessment Asking appropriate questions in CATs Minute Paper	<p>Novalabs cyber security protection game</p> <p>Cyber security threats and solutions</p> <ul style="list-style-type: none"> • 5 most dangerous hackers of all time • 10 biggest computer hacks of all time • Hacking a car with an ex-NSA hacker <p>Cloud storage</p> <p>senecalearning.com</p> <p>doddlelearn.co.uk</p>	M8 C1 C2 C3 Sp3

W.B. 19 th Oct	Mobile Technology	Students should understand the risks and benefits of wireless and mobile technologies.	<p>Students should consider the risks and benefits of wireless networking, including ease of access to the Internet and possible security risks.</p> <p>Mobile technologies are facilitating many new applications of computing and also making access to the Internet more widely available.</p> <p>Some of this content may have already been covered as part of specification section 3.5.</p> <p>This topic would be suitable for students to do individual research on and then have a class discussion about.</p>	<p>Purple book –End of Topic Assessment</p> <p>Asking appropriate questions in CATs</p> <p>Minute Paper</p>	<ul style="list-style-type: none"> • 5 most dangerous hackers of all time • 10 biggest computer hacks of all time • Hacking a car with an ex-NSA hacker <p>Cloud storage</p>	<p>M8</p> <p>C1</p> <p>C2</p> <p>C3</p> <p>Sp3</p> <p>Sp9</p>
W.B. 2 th Oct	Data types	<p>Understand the concept of data type</p> <p>Apply the listed programming techniques.</p> <p>Choose appropriate data types.</p> <p>Use meaningful identifier names and know why.</p> <p>Understand what an algorithm is and the</p>	<p>Understand and use the following appropriately:</p> <ul style="list-style-type: none"> • integer • real • Boolean • character • string 	<p>Class activities/exercises – Green book</p> <p>Asking appropriate questions in CATs</p> <p>Asking appropriate questions in CATs</p> <p>Minute Paper</p>	<p>Online activities - Doodle Learning</p> <p>Research on keywords and definitions</p> <p>Sequence and Selection Homework</p> <p>Notes on variables and data types (and some other concepts not required until later)</p> <p>senecalearning.com</p>	<p>Sp2</p> <p>Sp3</p>

		difference between an algorithm and program			doddelearn.co.uk	
W.B. 2 th Nov	Programming basic	<p>Understand and use string, integer and real data types appropriately.</p> <p>Understand how variable declaration and assignment can be used in programs.</p> <p>Be able to use addition, subtraction, multiplication and real division.</p> <p>Be able to perform input and output.</p>	<p>Students should be introduced to basic input and output commands, declaring variables (if required by language), and using arithmetic operations.</p> <p>Students will also need to be taught basic aspects of the IDE for their programming language e.g. how to run a program, how to load/save, how error messages are presented and what they mean.</p> <p>Students should be introduced to the idea of an algorithm and that a program is an implementation of an algorithm.</p>	<p>Purple book –End of Topic Assessment</p> <p>Class activities/exercises – Green book</p> <p>Asking appropriate questions in CATs</p>	<p>2A L1 Programming Data Homework</p> <p>Employment of PRIMM (Predict, Run, Investigate, Modify and Make)</p> <p>Research on keywords and definitions</p> <p>Muddiest point</p> <p>Documented problem solutions</p> <p>Directed paraphrasing</p> <p>Classroom opinion polls</p> <p>senecalearning.com</p>	<p>C1</p> <p>C2</p> <p>C3</p> <p>Sp3</p> <p>Sp5</p>
W.B. 9 th Nov	Using Python	<p>Use meaningful identifier names and know why it is important to use them.</p> <p>Understand and explain the term algorithm.</p>	<p>Getting the computer to display “Hello World” getting the user to type in their name and outputting hello to them (possibly concatenating forename and surname input separately)</p> <p>Doing simple calculations, for example adding three numbers, multiplying two numbers together doing more complex calculations, for</p>	<p>Purple book –End of Topic Assessment</p> <p>Class activities/exercises – Green book</p> <p>Asking appropriate questions in CATs</p>	<p>Programming tasks/ mini assignment</p> <p>Employment of PRIMM (Predict, Run, Investigate, Modify and Make)</p>	<p>C1</p> <p>C2</p> <p>C3</p> <p>C5</p> <p>Sp9</p>

			example area of a rectangle, area of a triangle, area of a circle, area of a trapezium.		<p>Research on keywords and definitions</p> <p>Muddiest point</p> <p>Documented problem solutions</p> <p>Directed paraphrasing</p> <p>Classroom opinion polls</p> <p>doddelearn.co.uk</p>	
W.B. 16 th Nov	Boolean Operators	<p>Be able to use selection (if, else, else if, case/switch if appropriate)</p> <p>Be able to use a range of relational operators.</p> <p>Be familiar with and able to use NOT, AND, OR.</p> <p>Using nested selection structures.</p> <p>Be able to select suitable test data that covers normal (typical), boundary and erroneous data. Be able to justify the choice of test data.</p>	<p>The focus in this section is on the use of selection statements to determine the path of code execution. Exercises should build in difficulty, starting with simple Yes/No answers using just an If statement then building in complexity in terms of the number of possible outcomes and the complexity of the criteria used.</p> <p>Pseudo-code and flowcharts could be used to illustrate some algorithms which students could then write program code for.</p> <p>Whilst completing these exercises, consideration should be given to choosing test data, which is particularly important in boundary situations of which there are many in these exercises.</p>	<p>Purple book –End of Topic Assessment</p> <p>Class activities/exercises – Green book</p> <p>Asking appropriate questions in CATs</p>	<p>2A I2 Sequence and Selection Homework</p> <p>Online activities - Dodde Learning</p> <p>Research on keywords and definitions</p> <p>Muddiest point</p> <p>Documented problem solutions</p> <p>Directed paraphrasing</p> <p>Classroom opinion polls</p> <p>doddelearn.co.uk</p>	<p>C1</p> <p>C2</p> <p>C3</p> <p>Sp3</p> <p>Sp5</p>

		Be able to understand pseudo-code and flowcharts.	<p>Exercises could include:</p> <ul style="list-style-type: none"> • exam mark pass/fail • determining if a person is a child/adult/pensioner based on their age • allocating an exam grade based on mark ranges • identifying the biggest of two or three numbers • Identifying if a triangle is scalene, isosceles or equilateral classifying the temperature based on a range e.g. 0 or below = freezing, above 0 but 10 or below = warm. 			
WB 23 th Nov	Iteration	<p>Be able to use definite iteration.</p> <p>Be able to use nested iteration.</p>	<p>Students should be introduced to the concept of definite iteration and a loop counter. Pseudo-code and flowcharts could be used to illustrate algorithms.</p> <p>Exercises could include:</p> <ul style="list-style-type: none"> • counting from one to ten • displaying a times table, or all times tables • adding up five numbers (average the same numbers and identify the highest and lowest) 	<p>Paper 1 mini assessment</p> <p>Purple book –End of Topic Assessment</p> <p>Class activities/exercises – Green book</p> <p>Asking appropriate questions in CATs</p>	<p>2A L3 Iteration Homework</p> <p>Research on keywords and definitions</p> <p>Muddiest point</p> <p>Documented problem solutions</p> <p>Directed paraphrasing</p> <p>Classroom opinion polls</p>	<p>C1</p> <p>C2</p> <p>C3</p> <p>Sp3</p> <p>Sp5</p>

			<ul style="list-style-type: none"> Working out factors of a number using brute-force approach - identifying prime numbers using brute-force approach. 			
W.B. 30 th Nov	Looping	<p>Be able to use indefinite iteration with conditions at start and end of loop.</p> <p>Be able to use random number generation.</p> <p>Be able to use some string handling techniques.</p> <p>Be able to write simple data validation routines.</p> <p>Be able to write simple authentication routines.</p> <p>Understand and explain the term abstraction.</p>	<p>Students need to be taught about indefinite iteration and how to use this in their programming language. For students using Python which does not have a post-conditioned loop, they should be taught how to implement post-conditioned loops as equivalent pre-conditioned loops.</p> <p>Students also need to know how to express these types of loop as pseudo-code and flowcharts.</p> <p>As students are now starting to tackle more complex problems, the concept of abstraction, i.e. removing unnecessary details from a problem, could be introduced at this point.</p> <p>Exercises could include:</p> <ul style="list-style-type: none"> performing simple validation e.g. that a typed value falls within a range or that an entered value cannot be left blank or is shorter than a minimum length adding up a sequence of numbers of unknown length asking users to enter a password until the correct password is entered, displaying suitable messages 	<p>Purple book –End of Topic Assessment</p> <p>Class activities/exercises – Green book</p> <p>Asking appropriate questions in CATs</p>	<p>Research on keywords and definitions</p> <p>Muddiest point</p> <p>Documented problem solutions</p> <p>Directed paraphrasing</p> <p>Classroom opinion polls</p>	<p>C1</p> <p>C2</p> <p>C3</p> <p>Sp3</p>

			<ul style="list-style-type: none"> guessing randomly chosen number until they guess correctly, with clues given about whether guess is too high/low Rolling two dice until a double six is scored, counting how many goes this takes. <p>Throwing darts and getting a random score on board (game starts at a total and plays with the total decreased by each dart thrown until 0 is achieved).</p>			
WB 7 TH Dec	Arrays	<p>Understand the concept of data structures.</p> <p>Use one-dimensional arrays (or equivalent) in the design of solutions to simple problems.</p> <p>Understand that more than one algorithm can be used to solve the same problem.</p> <p>Compare the efficiency of algorithms.</p> <p>Understand and explain how linear and binary search algorithms work and compare them.</p> <p>Understand and explain how bubble and merge sort</p>	<ul style="list-style-type: none"> inputting a list of names (or other data) and redisplaying them inputting a list of parcel weights (total the weights and work out the average, lowest and highest weight) searching a dictionary to check whether a word is in it using the linear search method improving the dictionary program to use the binary search method using the bubble sort algorithm to sort data (e.g. names) in an array looking theoretically at how the merge sort algorithm would perform the same sort (implementing merge sort is beyond GCSE but more able students could attempt this) comparing the efficiency of the search and sort algorithms 	<p>Purple book –End of Topic Assessment</p> <p>Class activities/exercises – Green book</p> <p>Asking appropriate questions in CATs</p>	<p>2A L4 Arrays Homework</p> <p>Research on keywords and definitions</p> <p>Muddiest point</p> <p>Documented problem solutions</p> <p>Directed paraphrasing</p> <p>Classroom opinion polls</p>	<p>C1</p> <p>C2</p> <p>C3</p> <p>Sp3</p>

		<p>algorithms work and compare them.</p> <p>Use trace tables.</p>	<p>Representing a game of snakes and ladders using a one-dimensional array to indicate the positions of snakes and ladders.</p>			
<p>WB 14TH Dec</p>	<p>Think like a computer!</p>	<p>Understand and explain the term decomposition.</p> <p>Describe the structured approach to programming.</p> <p>Explain the advantages of the structured approach.</p> <p>Understand the concept of subroutines and be able to use them in programs, including the use of local variables.</p> <p>Explain the advantages of using subroutines in programs.</p> <p>Integer division, including remainders.</p>	<p>Students should be taught about why, when writing longer programs, it is useful to decompose them, and the facilities in their programming language to do this. They should also cover the difference between local and global variables. At this stage, parameters and return values can be ignored.</p> <p>Exercises could include:</p> <ul style="list-style-type: none"> making a maths toolkit, with a menu that is used to call different subroutines to work out (for example) the area of different shapes making a program that will allow conversion of numbers between different number bases, with different functions being used for different conversions e.g. binary to decimal <p>In all subsequent programs, students should be encouraged to consider how the programs can be decomposed into functions.</p>	<p>Purple book –End of Topic Assessment</p> <p>Class activities/exercises – Green book</p> <p>Asking appropriate questions in CATs</p>	<p>Research on keywords and definitions</p> <p>Muddiest point</p> <p>Documented problem solutions</p> <p>Directed paraphrasing</p> <p>Classroom opinion polls</p> <p>doddlelearn.co.uk</p>	<p>C1 C2 C3 C4 Sp3</p>
<p>WB 5th Jan</p>	<p>Parameters and Functions</p>	<p>Use a structured approach to programming, in particular focussing on the</p>	<p>Emphasis should be on passing input to the functions as parameters and using return to pass values back to the calling program.</p>	<p>Purple book –End of Topic Assessment</p>	<p>Research on keywords and definitions</p> <p>Muddiest point</p>	<p>Sp3 Sp7</p>

		<p>use of parameters and return values.</p> <p>Use a range of string handling operations.</p> <p>Use the char and Boolean data types.</p>	<p>Input/output via the keyboard/screen should not happen within the functions.</p> <p>Students should be taught why this is important, for example in terms of being able to develop and test modules independently and reuse code.</p> <p>Exercises could include:</p> <ul style="list-style-type: none"> • developing a function that returns the highest of two numbers and adapting this to find the highest of three numbers or to perform other mathematical operations • developing a function that indicates whether a number is even or not • Developing a function that works out n factorial (n!) • developing a function that returns a string that has been encrypted using the Caesar Cipher with a key selected by the user and adding a decryption function • developing a function to convert a string into Morse code • developing a function that will return a true/false value, indicating if two words are anagrams of each other • Developing a function that, when sent a number, will return a true/false value indicating whether the number is a perfect number or not and using this in 	<p>Class activities/exercises – Green book</p> <p>Asking appropriate questions in CATs</p>	<p>Documented problem solutions</p> <p>Directed paraphrasing</p> <p>Classroom opinion polls</p>	
--	--	---	---	--	---	--

			<p>a program to search for perfect numbers using brute-force.</p> <p>In all subsequent programs, students should be encouraged to consider how the programs can be decomposed into functions with interfaces that use parameters and return values.</p>			
WB 11 th Jan	Using Files	Be able to read/write from/to a text file.	<p>Students first need to understand what a text file is. They should then modify some of the previous programs that they have written to read input from/ save output to a text file.</p> <p>Some suitable programs to modify would be:</p> <p>Snakes and ladder positions could be stored in a text file (allowing for the possibility of different boards).</p> <p>The dictionary that is searched could be stored in a text file.</p> <p>The sorting program could load the list to sort from a text file and save the sorted list to a different text file.</p> <p>Students should consider how to deal with possible issues such as saving over an existing file or being asked to load a file that does not exist. This would be a point where exception handling could be considered.</p>	<p>Purple book –End of Topic Assessment</p> <p>Class activities/exercises – Green book</p> <p>Asking appropriate questions in CATs</p>	<p>2A L5 Records and Files Homework</p> <p>Python worksheets activities</p> <p>Research on keywords and definitions</p> <p>Muddiest point</p> <p>Documented problem solutions</p> <p>Directed paraphrasing</p> <p>Classroom opinion polls</p> <p>senecalearning.com</p>	

<p>WB 18th Jan</p>	<p>Arrays, variables and constants</p>	<p>Use two-dimensional arrays (or equivalent) in the design of solutions to simple problems.</p> <p>Use nested iteration.</p> <p>Use of constants.</p>	<p>Students should have the opportunity to write programs using two-dimensional arrays. They will need to consider/design how the arrays can be used to represent the problem. Data stored in a two-dimensional array is usually displayed most conveniently using nested loops.</p> <p>A range of games can be readily implemented using two-dimensional arrays.</p> <p>If students have not yet encountered constants, they could be introduced here, for example, to store the size of a game board.</p> <p>Exercises could include:</p> <ul style="list-style-type: none"> • snakes and ladders • noughts and crosses • Battleships. 	<p>Purple book –End of Topic Assessment</p> <p>Class activities/exercises – Green book</p> <p>Asking appropriate questions in CATs</p>	<p>Research on keywords and definitions</p> <p>Muddiest point</p> <p>Documented problem solutions</p> <p>Directed paraphrasing</p> <p>Classroom opinion polls</p>	<p>C1 C2 C3 Sp3</p>
<p>WB 25th Jan</p>	<p>Routines and subroutines</p>	<p>Use records (or equivalent) in the design of solutions to simple problems.</p> <p>Be able to write simple authentication routines.</p>	<p>Students should be introduced to the concept of records and why logically grouping related data together is a sensible approach.</p> <p>Exercises could include:</p> <ul style="list-style-type: none"> • adapt the dictionary program that was written earlier to store equivalent words in two languages in an array of records and perform translation between them 	<p>Purple book –End of Topic Assessment</p> <p>Class activities/exercises – Green book</p> <p>Asking appropriate questions in CATs</p>	<p>Research on keywords and definitions</p> <p>Muddiest point</p> <p>Documented problem solutions</p> <p>Directed paraphrasing</p> <p>Classroom opinion polls</p>	<p>C1 C2 C3 Sp3</p>

			<ul style="list-style-type: none"> write an address book program, or a program to keep track of any other data (this data could be saved/loaded from a text file using CSV format) <p>write a login system with usernames and passwords stored in a file and then loaded into an array of records</p>			
WB 1st Feb	Flowcharts and Pseudo-code	<p>Use a systematic approach to problem solving and algorithm creation representing those algorithms using pseudo-code and flowcharts.</p> <p>Explain simple algorithms in terms of their inputs, processing and outputs.</p> <p>Determine the purpose of simple algorithms.</p>	<p>Throughout learning to program, students should be exposed to how algorithms can be expressed using pseudo-code or flowcharts.</p> <p>Students need to have some practice at being able to understand and write algorithms using these methods.</p> <p>They also need to be able to use trace tables to record the values of variables as an algorithm is stepped through and to be able to identify the purpose of an algorithm by tracing it.</p> <p>These skills will be assessed in the exam. It is useful to teach them in parallel with learning to program (perhaps as homework exercises) but it could also be worth giving students the opportunity to consolidate their ability to apply these skills.</p> <p>Students should complete exercises where they have to read and write pseudo-code and flowcharts, complete trace tables and deduce the purpose of algorithms.</p>	<p>Full Paper 1 assessment</p> <p>Purple book –End of Topic Assessment</p> <p>Class activities/exercises – Green book</p> <p>Asking appropriate questions in CATs</p>	<p>Research on keywords and definitions</p> <p>Muddiest point</p> <p>Documented problem solutions</p> <p>Directed paraphrasing</p> <p>Classroom opinion polls</p> <p>senecalearning.com</p>	<p>C1</p> <p>C2</p> <p>C3</p> <p>Sp3</p>

<p>WB 8th Feb</p>	<p>Programming language</p>	<p>Know that there are different levels of programming language: low-level, high-level and explain the main differences between them.</p> <p>Know that machine code and assembly language are considered to be low-level languages and explain the differences between them.</p> <p>Understand that ultimately all programming code written in high-level or assembly languages must be translated into machine code.</p> <p>Understand that machine code is expressed in binary and is specific to a processor or family of processors.</p> <p>Understand the advantages and disadvantages of low-level language programming compared with high-level language programming.</p>	<p>Students only need a theoretical understanding of this topic, so this topic would be best delivered by the teacher as a presentation or through notes or a video with students given the opportunity to answer questions on it. Including real examples of assembly language and machine code is helpful.</p> <p>Students could be given some opportunity to write very simple programs in assembly language so that they can see how assembly language compares with a high-level language but this is not a requirement.</p> <p>Students using Visual Basic or C# could find the exe file that is output when they compile a program.</p> <p>Students could look at an assembly language instruction set (e.g. ARM) to consider the types of instruction available and their limitations.</p>	<p>Purple book –End of Topic Assessment</p> <p>Class activities/exercises – Green book</p> <p>Asking appropriate questions in CATs</p>	<p>Research on keywords and definitions</p> <p>Muddiest point</p> <p>Documented problem solutions</p> <p>Directed paraphrasing</p> <p>Classroom opinion polls</p> <p>senecalearning.com</p>	<p>C1 C2 C3 Sp3 Sp7 Sp9</p>
----------------------------------	------------------------------------	--	--	--	---	---

		<p>Understand that there are three common types of program translator: interpreter, compiler, assembler.</p> <p>Explain the main differences between these three types of translator and understand when it would be appropriate to use each type of translator.</p>				
WB 22 nd Feb	Images	<p>Understand how images can be represented as bitmaps, including key terms.</p> <p>Be able to calculate file sizes.</p> <p>Be able to convert between binary and image data for black and white images</p>	<p>Look at bitmap images using a graphics package, use zoom to identify pixels and colours (possible link to hex).</p> <p>Introduce colour depth by considering how different patterns of 0s and 1s could be used to represent colours. A colour depth of n bits allows 2^n colours.</p> <p>Perform some exercises where students have to convert small images between images and binary data and vice-versa. Only black and white images need to be considered.</p> <p>Explain how to calculate the size of an image file and then students complete some sample calculations.</p>	<p>Purple book –End of Topic Assessment</p> <p>Class activities/exercises – Green book</p> <p>Asking appropriate questions in CATs</p>	<p>Unit 3 Lesson 4 Images Homework</p> <p>Research on keywords and definitions</p> <p>Muddiest point</p> <p>Documented problem solutions</p> <p>Directed paraphrasing</p> <p>Classroom opinion polls</p> <p>senecalearning.com</p>	Sp9
WB 12 th Apr	Embedded Systems	<p>Understand the term ‘embedded system’ and explain how an embedded system differs from a non-embedded system.</p>	<p>This is a relatively small topic. Students need to understand that many computer systems are embedded in other devices and the constraints</p>	<p>Assessment for CPU – secondary storage and Embedded systems</p>	<p>Research on keywords and definitions</p> <p>Muddiest point</p>	<p>C1 C2 C3 Sp3 Sp9</p>

			<p>and differences that this produces when compared with non-embedded systems.</p> <p>Students could be given some scenarios (e.g. washing machine) and be asked to consider what functionality the system would need and why a non-embedded system would not be suitable.</p> <p>Differences such as processor speed, amount and type of main memory, secondary storage, input and output devices and upgradeability could be considered.</p>	<p>Purple book –End of Topic Assessment</p> <p>Class activities/exercises – Green book</p> <p>Asking appropriate questions in CATs</p>	<p>Documented problem solutions</p> <p>Directed paraphrasing</p> <p>Classroom opinion polls</p>	
WB 19 th Apr	Revision					
WB 26 th Apr	Revision					
WB 4 th May	Revision					
WB 10 th May	Revision					

- Lesson/Week Number (Once we know plans for next academic year)
- Topic and Objectives

- Big Questions
- Key Activities
- Planned Assessments (once we know plans for next academic year)
- Homework/Flipped learning
- **SMSC Codes**

References:

<https://www.cmu.edu/teaching/assessment/assesslearning/CATs.html>

<https://filestore.aqa.org.uk/resources/computing/specifications/AQA-8525-SP-2020.PDF>

<https://app.senecalearning.com/classroom/course/90cb1d71-3c44-4a2d-a07e-d76a5f54cd05/section/8c1b3f7a-9650-4943-8884-8cdc32d1c814/session>

<https://www.doddlelearn.co.uk/app/login?>

<http://www.neuroproductions.be/logic-lab/>

<http://neuroproductions.be/>

<https://www.youtube.com/watch?v=cqvh1Z1Vvck>

https://www.netdevgroup.com/support/documentation/netlabve/netlabve_instructor_guide.pdf