Year 10 Computer Science

2020 – 2021

There are two parts to the Computer Science course, Programming and Computational thinking, and these two parts are taught concurrently to enable students to make progress with the theory whilst practising and developing their programming skills. Students are thus able to build on all aspects of the work introduced in year 9 and make the important links between these two areas.

In half term 1 students will cover Programming 1- 6 and Computational thinking 1 - 6.

In half term 2 students will cover Programming 7 - 12 and Computational thinking 7 – 12

Each half terms work is concluded by some review and assessment

The scheme of work for computer science is designed to ensure any gaps and misconceptions from year 9 are addressed. The curriculum re-visits key concepts regularly, and recaps and builds on the preliminary and introductory work from year 9 in the context of the GCSE specification. It is not necessary therefore to change the planned curriculum for year 10.

Low-stakes quizzing is used via Microsoft forms and BBC bitesize topic test to track students' knowledge and address gaps as they occur. Results are recorded in the Class Notebook, so that gaps in knowledge can be identified and individual support in the form of one to one/small group catch up sessions can be given to students who continue to have misconceptions.

Key Knowledge which we will re-visit during year 10 include

9.6 Computational Thinking, covering decomposition, pattern recognition, abstraction and problem solving.

Computer Science - Year 10 Term 1.1 CT 1 - 6

What are we learning?	What knowledge, understanding and skills will we gain?	What does mastery look like?	How does this build on prior learning?	What additional resources are available?
Binary numbers Unsigned integers Binary arithmetic Two's complement	**Knowledge**: Define what is meant by the term 'digital computer' Give examples of different types of computer Define what is meant by the terms 'binary' and 'bit' Define what is meant by the terms 'nibble' and 'byte' Define what is meant by the term 'overflow error' **Understanding:** Explain why binary is used to represent data and program instructions in a computer Describe the relationship between the number of available bits and the range of unique values that can be represented Describe the effects of an overflow error Differentiate between signed and unsigned integers Describe how positive and negative numbers are represented in two's complement **Skills:** Determine the number of unique values that can be represented by a binary pattern of a given length (2^n) Convert between denary and 8-bit binary numbers Add together two positive 8-bit binary integers Find the two's complement of a positive binary number	Define what is meant by the terms 'bit', 'nibble' and 'byte' List three types of data represented in binary in a computer system Give the 8-bit binary equivalent of an unsigned denary number Give the denary equivalent of an unsigned 8-bit binary number Add together two positive 8-bit binary numbers Explain what is meant by an overflow error	8.2 Binary Bits and Bobs	Seneca BBC Bitesize cs4fn, Queen Mary, University of London (www.cs4fn.org). Computer Science Unplugged (http://csunplugged.org/ *BCS Glossary of Computing and ICT*, 13th edition (ISBN 9781780171500) A range of articles on teaching coding: www.edsurge.com/guide/teaching-kids-to-code **Python resources** Official Python documentation (also available through help in IDLE): www.docs.python.org/3/ Python summer school from Anglia Ruskin University is an excellent resource with videos and programming challenges: http://www.pythonschool.net/ Python code for kids is a clearly written summary of the Python language written in accessible language: www.pythondictionary.code-it.co.uk/ 'Python in 10 minutes' is a quick run through of the basic concepts: www.korokithakis.net/tutorials/python/ 'Quintin Cutts – Too much doing, not enough understanding' is a 20-minute video containing useful ideas and concepts on how to teach programming: www.youtube.com/watch?v=Pim4aYfiZiY

Computer Science - Year 10 Term 1.1 P 1 - 6

What are we learning?	What knowledge, understanding and skills will we gain?	What does mastery look like?	How does this build on prior learning?	What additional resources are available?
Intro to programming Decomposition Algorithms Data types Variables Input and integer functions Debugging tools Flowcharts	Knowledge: Define the term ‘program’ Identify types of programs used every day Identify Python as a programming language Define the term ‘decomposition’ Define the term ‘algorithm’ Define the term ‘sequence’ Define the term ‘variable’ Define the term ‘runtime error’ Understanding: Layout code to be readable and maintainable Decompose a problem Order the pieces of an algorithm (unplugged) Recognise primitive data types (int, real, char, string) Create variables of all types Create meaningful identifier names Translate code into flowchart symbols Represent an algorithm in a flowchart Skills: Access an integrated development environment Load and run a Python program Change a Python program Save a Python program Use arithmetic operators and BIDMAS Correct errors in programs Use variables in algorithms and programs Order the pieces of an algorithm (IDE) Use sequence in algorithms and program code Interpret error messages Correct errors in ordering Assign values to variables, with the correct data types View contents of memory (variable) in IDE	Represent algorithms in flowcharts Create code from algorithms represented in flowcharts	7.1 Introduction to Scratch 7.6 Micro:Bit Madness 8.3 From Scratch to Python 9.1 More Python 9.6 Computational Thinking	Seneca BBC Bitesize cs4fn, Queen Mary, University of London (www.cs4fn.org). Computer Science Unplugged (http://csunplugged.org/ *BCS Glossary of Computing and ICT*, 13th edition (ISBN 9781780171500) A range of articles on teaching coding: www.edsurge.com/guide/teaching-kids-to-code Python resources Official Python documentation (also available through help in IDLE): www.docs.python.org/3/ Python summer school from Anglia Ruskin University is an excellent resource with videos and programming challenges: http://www.pythonschool.net/ Python code for kids is a clearly written summary of the Python language written in accessible language: www.pythondictionary.code-it.co.uk/ ‘Python in 10 minutes’ is a quick run through of the basic concepts: www.korokithakis.net/tutorials/python/ ‘Quintin Cutts – Too much doing, not enough understanding’ is a 20-minute video containing useful ideas and concepts on how to teach programming: www.youtube.com/watch?v=Pim4aYfiZiY

	Take input and create output Find and fix runtime errors Use primitive data types (integer, real, char, string) Translate a flowchart into code			

Computer Science - Year 10 Term 1.2 CT 7 - 12

What are we learning?	What knowledge, understanding and skills will we gain?	What does mastery look like?	How does this build on prior learning?	What additional resources are available?
Two's complement 2 Logical binary shifts Arithmetic binary shifts Hexadecimal ASCII	**Knowledge**: Signed denary numbers Two's complement binary numbers Define what is meant by the term 'hexadecimal'. Define what is meant by the term 'character set' **Understanding:** Determine the range of values that can be represented in two's complement by a binary number of a given length. Explain why a number may be less precise after a binary shift right has been applied. Describe how an arithmetic right shift differs from a logical right shift Explain why hexadecimal notation is used Describe how characters are represented in 7-bit ASCII **Skills:** Convert between signed denary numbers and two's complement binary numbers. Apply logical left and right shifts to binary integers. Use logical binary shifts to multiply and divide unsigned binary integers by powers of 2. Apply arithmetic left and right shifts to signed binary numbers. Convert between hexadecimal	Apply a logical binary shift left to a positive 8-bit binary number Apply a logical shift right to a positive 8-bit binary number Explain why a binary number may become less accurate after a binary shift right Apply an arithmetic binary shift right to a two's complement number Give the hexadecimal equivalent of an 8-bit binary number Give the binary equivalent of a hexadecimal number Explain why hexadecimal is used	8.2 Binary Bits and Bobs Unit CT1-6	Seneca BBC Bitesize cs4fn, Queen Mary, University of London (www.cs4fn.org). Computer Science Unplugged (http://csunplugged.org/ *BCS Glossary of Computing and ICT*, 13th edition (ISBN 9781780171500) A range of articles on teaching coding: www.edsurge.com/guide/teaching-kids-to-code **Python resources** Official Python documentation (also available through help in IDLE): www.docs.python.org/3/ Python summer school from Anglia Ruskin University is an excellent resource with videos and programming challenges: http://www.pythonschool.net/ Python code for kids is a clearly written summary of the Python language written in accessible language: www.pythondictionary.code-it.co.uk/

	and binary. Given the ASCII code for one character derive the code for another Outline the shortcomings of ASCII and how encoding systems that use more bits overcome them.	Describe how characters are encoded in ASCII Derive the code for an ASCII character from that of another Describe the limitations of ASCII		'Python in 10 minutes' is a quick run through of the basic concepts: www.korokithakis.net/tutorials/python/ 'Quintin Cutts – Too much doing, not enough understanding' is a 20-minute video containing useful ideas and concepts on how to teach programming: www.youtube.com/watch?v=Pim4aYfiZiY

Computer Science - Year 10 Term 1.2 P 7 - 12

What are we learning?	What knowledge, understanding and skills will we gain?	What does mastery look like?	How does this build on prior learning?	What additional resources are available?
String manipulation, string methods if, if else, relational operators if elif else, readability Boolean operators Repetition (while)	Knowledge: What is string manipulation Describe string methods Define 'AND', 'NOT' and 'OR' Understanding: The purpose of string manipulation Use flowchart decision symbol Identify parts of code (variables, constants, selection, repetition) Construct truth tables for Boolean operators and combinations Use repetition (condition-controlled loops) in algorithms Skills: Use string manipulation functions (index, left, right, upper, lower, isalpha, …, etc.) Use relational operators in flowchart and code Use 'if' and 'if else' in code Use comments, white space, meaningful identifiers, and indentation in code	Use flowcharts to represent selection and repetition Identify parts of a program Solve problems using code Use repetition in code Use selection in code	7.1 Introduction to Scratch 7.6 Micro:Bit Madness 8.3 From Scratch to Python 9.1 More Python 9.6 Computational Thinking Unit P1-6	Seneca. BBC Bitesize. cs4fn, Queen Mary, University of London (www.cs4fn.org). Computer Science Unplugged (http://csunplugged.org/ BCS Glossary of Computing and ICT, 13th edition (ISBN 9781780171500) A range of articles on teaching coding: www.edsurge.com/guide/teaching-kids-to-code Python resources Official Python documentation (also available through help in IDLE): www.docs.python.org/3/ Python summer school from Anglia Ruskin University is an excellent resource with videos and programming challenges: http://www.pythonschool.net/ Python code for kids is a clearly written summary of the Python language written in accessible language: www.pythondictionary.code-it.co.uk/

	Use relational operators in flowchart and code Use repetition (condition-controlled loops) in code			‘Python in 10 minutes’ is a quick run through of the basic concepts: www.korokithakis.net/tutorials/python/ ‘Quintin Cutts – Too much doing, not enough understanding’ is a 20-minute video containing useful ideas and concepts on how to teach programming: www.youtube.com/watch?v=Pim4aYfiZiY

Computer Science - Year 10 Term 2.1 CT 13 - 18

What are we learning?	What knowledge, understanding and skills will we gain?	What does mastery look like?	How does this build on prior learning?	What additional resources are available?
Stored program concept Fetch-decode-execute Secondary storage	**Knowledge**: What is the ‘stored program concept’. Hardware components. Von Neumann architecture. What is secondary storage **Understanding:** Define what is meant by the ‘stored program concept’. Describe the hardware components used in the von Neumann architecture and explain their role in the fetch-decode-execute cycle. Explain how the speed of the clock impacts on performance. Explain how pipelining improves the performance of the CPU. Explain the relationship between the width of the address bus and the number of memory locations that can be addressed. Explain why secondary storage is needed Describe how data are stored on magnetic, optical and solid-state media. **Skills:** Calculate the number of addressable memory locations provided by an address bus of a specified width. Compare the capacity, speed and portability of magnetic, optical and solid-state storage devices. Select an appropriate	Define what is meant by the term ‘stored program concept’ Describe what is stored in main memory when a program is running Explain what happens during the fetch-decode-execute cycle and the role of specified components Label and complete a diagram of the inside of a computer Explain the need for secondary storage Describe how data are stored on a solid-state drive	8.2 Binary Bits and Bobs Unit CT1-6 Unit CT7-12	Seneca BBC Bitesize cs4fn, Queen Mary, University of London (www.cs4fn.org). Computer Science Unplugged (http://csunplugged.org/ *BCS Glossary of Computing and ICT*, 13th edition (ISBN 9781780171500) A range of articles on teaching coding: www.edsurge.com/guide/teaching-kids-to-code **Python resources** Official Python documentation (also available through help in IDLE): www.docs.python.org/3/ Python summer school from Anglia Ruskin University is an excellent resource with videos and programming challenges: http://www.pythonschool.net/ Python code for kids is a clearly written summary of the Python language written in accessible language: www.pythondictionary.code-it.co.uk/

	type of storage for a particular purpose. Construct an expression to calculate data storage requirements.			'Python in 10 minutes' is a quick run through of the basic concepts: www.korokithakis.net/tutorials/python/ 'Quintin Cutts – Too much doing, not enough understanding' is a 20-minute video containing useful ideas and concepts on how to teach programming: www.youtube.com/watch?v=Pim4aYfiZiY

Computer Science - Year 10 Term 2.1 P13 - 18

What are we learning?	What knowledge, understanding and skills will we gain?	What does mastery look like?	How does this build on prior learning?	What additional resources are available?
One-dimensional lists for loops, range function Procedures Functions Subprograms	Knowledge: Define the terms 'array' and 'list' Define the term 'procedure' Define the term 'parameter' Define the term 'function' Define the term 'return value' Understanding: Explain that the range() function generates a sequence of numbers Use 'separation of concerns' Skills: Access each item in a list using indexing Create, append, delete items from a list Use iteration 'for' to process every item in a one-dimensional data structure Create procedures	Use 'lists' Use 'range()' Use 'for' Create procedures Create functions	7.1 Introduction to Scratch 7.6 Micro:Bit Madness 8.3 From Scratch to Python 9.1 More Python 9.6 Computational Thinking Unit P1-6 Unit P7-12	Seneca BBC Bitesize cs4fn, Queen Mary, University of London (www.cs4fn.org). Computer Science Unplugged (http://csunplugged.org/ BCS Glossary of Computing and ICT, 13th edition (ISBN 9781780171500) A range of articles on teaching coding: www.edsurge.com/guide/teaching-kids-to-code **Python resources** Official Python documentation (also available through help in IDLE): www.docs.python.org/3/ Python summer school from Anglia Ruskin University is an excellent resource with videos and programming challenges: http://www.pythonschool.net/ Python code for kids is a clearly written summary of the Python language written in accessible language: www.pythondictionary.code-it.co.uk/ 'Python in 10 minutes' is a quick run through of the basic concepts: www.korokithakis.net/tutorials/python/ 'Quintin Cutts – Too much doing, not enough understanding' is a 20-minute video containing useful

	Create functions			ideas and concepts on how to teach programming: www.youtube.com/watch?v=Pim4aYfiZiY

Computer Science - Year 10 Term 2.2 CT 19 - 24				
What are we learning?	**What knowledge, understanding and skills will we gain?**	**What does mastery look like?**	**How does this build on prior learning?**	**What additional resources are available?**
Operating systems OS: file management OS: process management OS: peripherals & user management Utility software	**Knowledge**: What is an operating system Identify tasks carried out by an OS Define what is meant by the term 'peripheral' Define what is meant by the term 'access control' Define what is meant by the term 'utility software' Identify different types of utility software **Understanding:** Describe the role of the operating system in a computer system. Describe how the OS organises files and allocates space on a hard drive. Describe how file permissions are used to control access to files. Describe how an OS uses scheduling to give each active process a share of CPU time. Describe the features of the round-robin scheduling algorithm. Describe how the OS uses a paging algorithm to swap programs in and out of main memory. Describe how the OS uses drivers to communicate with and manage peripherals. Explain the purpose of a user interface and describe features of a user interface.Describe commonly used methods of authentication. Describe the purpose of: – file repair/recovery software – backup/recovery software – file compression software	Define what is meant by the term 'operating system' Describe how files are organised Select appropriate permissions for specified users Define what is meant by the term 'process' Describe how an OS allocates each active process a share of CPU time Explain the role of a device driver	8.2 Binary Bits and Bobs Unit CT1-6 Unit CT7-12 Unit CT13-18	Seneca BBC Bitesize cs4fn, Queen Mary, University of London (www.cs4fn.org). Computer Science Unplugged (http://csunplugged.org/ *BCS Glossary of Computing and ICT*, 13th edition (ISBN 9781780171500) A range of articles on teaching coding: www.edsurge.com/guide/teaching-kids-to-code **Python resources** Official Python documentation (also available through help in IDLE): www.docs.python.org/3/ Python summer school from Anglia Ruskin University is an excellent resource with videos and programming challenges: http://www.pythonschool.net/ Python code for kids is a clearly written summary of the Python language written in accessible language: www.pythondictionary.code-it.co.uk/

	– disk defragmentation software **Skills:** Construct an expression to calculate the number of blocks of space on a hard drive needed to store a file of a given size. Select an appropriate level of file access (read, write, delete, none) for a user. Select suitable access right for specified individuals. Select which utility software tool to use for a particular task.	Describe features of a GUI user interface Select a utility tool for a specified job.		Python in 10 minutes' is a quick run through of the basic concepts: www.korokithakis.net/tutorials/python/ 'Quintin Cutts – Too much doing, not enough understanding' is a 20-minute video containing useful ideas and concepts on how to teach programming: www.youtube.com/watch?v=Pim4aYfiZiY

Computer Science - Year 10 Term 2.2 P 19 - 24				
What are we learning?	**What knowledge, understanding and skills will we gain?**	**What does mastery look like?**	**How does this build on prior learning?**	**What additional resources are available?**
String.format() Two-dimensional lists Validation Linear search (one-dimensional) Linear search (two-dimensional)	Knowledge: Define the term 'array'. Define the term 'list'. Understanding: Give characteristics of one-dimensional and two-dimensional data structures. Apply a linear search to a one-dimensional list (paper). Complete a linear search algorithm in a flowchart. Apply a linear search to a two-dimensional list (paper) Complete a linear search algorithm in a flowchart. Skills: Format output to meet requirements. Format output suitable for the end user. Use indexing to access any item in a two-dimensional structure. Use 'for' to iterate over every item in a two-dimensional structure. Use 'while' to find a row in a two-dimensional structure. Validate input using presence check, length check, range check, pattern check. Write a linear search for a single item in a one-dimensional list (code). Write a linear search for a single record in a two-dimensional list (code)	Use one-dimensional and two-dimensional lists Find a single item in a one-dimensional list Find a single record and file in a two-dimensional list	7.1 Introduction to Scratch 7.6 Micro:Bit Madness 8.3 From Scratch to Python 9.1 More Python 9.6 Computational Thinking Unit P1-6 Unit P7-12 Unit P13-18	Seneca. BBC Bitesize. cs4fn, Queen Mary, University of London (www.cs4fn.org). Computer Science Unplugged (http://csunplugged.org/ BCS Glossary of Computing and ICT, 13th edition (ISBN 9781780171500) A range of articles on teaching coding: www.edsurge.com/guide/teaching-kids-to-code Python resources Official Python documentation (also available through help in IDLE): www.docs.python.org/3/ Python summer school from Anglia Ruskin University is an excellent resource with videos and programming challenges: http://www.pythonschool.net/ Python code for kids is a clearly written summary of the Python language written in accessible language: www.pythondictionary.code-it.co.uk/ 'Python in 10 minutes' is a quick run through of the basic concepts: www.korokithakis.net/tutorials/python/

	.			'Quintin Cutts – Too much doing, not enough understanding' is a 20-minute video containing useful ideas and concepts on how to teach programming: www.youtube.com/watch?v=Pim4aYfiZiY

Computer Science - Year 10 Term 3.1 CT 25 - 30				
What are we learning?	**What knowledge, understanding and skills will we gain?**	**What does mastery look like?**	**How does this build on prior learning?**	**What additional resources are available?**
Malware & anti-malware Hackers Social engineering Data-level protection	**Knowledge**: Define what is meant by the term 'cyberattack'. Define what is meant by the term 'hacker'. Define what is meant by the term 'social engineering'. **Understanding:** Describe the financial, reputational and legal damage that a cyberattack can cause. Describe the characteristics of and threat posed by different types of malware. Describe how anti-malware works. Explain why it is important to keep anti-malware up-to-date. Explain why unpatched software is a target for hackers. Explain the function of a firewall. Explain how ethical hacking and penetration testing help identify vulnerabilities. Describe commonly used social engineering tactics (phishing, pretexting, baiting, quid pro quo) used by hackers. Explain the purpose of an acceptable use policy and what it typically includes. Explain how data are protected by encryption. Describe how backup and recovery procedures protect against data loss. Explain how access control helps to protect systems and data. Explain how a hacker can exploit a code vulnerability. Describe examples of bad coding	Identify a type of malware. Describe how anti-malware protects digital systems and data. Explain how backup and recovery procedures would help an organisation withstand a ransomware attack. Explain the security threat posed by unpatched software. Describe the purpose of an acceptable use policy. Describe two bad programming	8.2 Binary Bits and Bobs Unit CT1-6 Unit CT7-12 Unit CT13-18 Unit CT 19-24	Seneca. BBC Bitesize. cs4fn, Queen Mary, University of London (www.cs4fn.org). Computer Science Unplugged (http://csunplugged.org/ *BCS Glossary of Computing and ICT*, 13th edition (ISBN 9781780171500) A range of articles on teaching coding: www.edsurge.com/guide/teaching-kids-to-code **Python resources** Official Python documentation (also available through help in IDLE): www.docs.python.org/3/ Python summer school from Anglia Ruskin University is an excellent resource with videos and programming challenges: http://www.pythonschool.net/ Python code for kids is a clearly written summary of the Python language written in accessible language: www.pythondictionary.code-it.co.uk/ 'Python in 10 minutes' is a quick run through of the basic concepts: www.korokithakis.net/tutorials/python/ 'Quintin Cutts – Too much doing, not enough understanding' is a 20-minute video containing useful ideas and concepts on how to teach programming: www.youtube.com/watch?v=Pim4aYfiZiY

	practices and secure coding practices Explain how code reviews and audit trails help to identify vulnerabilities. **Skills:** Identify a type of malware. Describe how anti-malware protects digital systems and data. Explain how backup and recovery procedures would help an organisation withstand a ransomware attack. Explain the security threat posed by unpatched software. Describe the purpose of an acceptable use policy. Describe bad programming practices that could make software vulnerable to attack.	practices that could make software vulnerable to attack.		

Computer Science - Year 10 Term 3.1 P 25 - 30

What are we learning?	What knowledge, understanding and skills will we gain?	What does mastery look like?	How does this build on prior learning?	What additional resources are available?
Merge sort Reading files String processing Writing files Authentication	Knowledge: Describe the merge sort algorithm Define the term 'authentication' Understanding: Create a flowchart for algorithm Skills: Merge two sorted lists (paper, code) Open files for reading Read lines from text files Close a file Split lines on commas Store items in lines as records in two-dimensional structure Open files for writing Construct comma-separated value line from record in two-dimensional structure	Validate input Read and write files Iterate over all records in a two-dimensional structure	7.1 Introduction to Scratch 7.6 Micro:Bit Madness 8.3 From Scratch to Python 9.1 More Python 9.6 Computational Thinking Unit P1-6 Unit P7-12 Unit P13-18 Unit P19-24	Seneca BBC Bitesize cs4fn, Queen Mary, University of London (www.cs4fn.org). Computer Science Unplugged (http://csunplugged.org/ BCS Glossary of Computing and ICT, 13th edition (ISBN 9781780171500) A range of articles on teaching coding: www.edsurge.com/guide/teaching-kids-to-code Python resources Official Python documentation (also available through help in IDLE): www.docs.python.org/3/ Python summer school from Anglia Ruskin University is an excellent resource with videos and programming challenges: http://www.pythonschool.net/ Python code for kids is a clearly written summary of the Python language written in accessible language: www.pythondictionary.code-it.co.uk/

	Write comma separated text (records) to a file Implement authentication using a two-dimensional structure with at least two columns			'Python in 10 minutes' is a quick run through of the basic concepts: www.korokithakis.net/tutorials/python/ 'Quintin Cutts – Too much doing, not enough understanding' is a 20-minute video containing useful ideas and concepts on how to teach programming: www.youtube.com/watch?v=Pim4aYfiZiY

Computer Science - Year 10 Term 3.2 CT 31 - 36				
What are we learning?	**What knowledge, understanding and skills will we gain?**	**What does mastery look like?**	**How does this build on prior learning?**	**What additional resources are available?**
LANs and WANs Network speed Connectivity Wired vs wireless Network topologies	**Knowledge**: LAN and WAN. Define the meanings of the terms 'bandwidth' and 'latency'. Differentiate between wired and wireless connectivity. Define the term 'topology'. **Understanding:** Give reasons why computers are connected on a network. Differentiate between a LAN and a WAN. Explain the benefits to organisations of a WAN. Explain why protocols are needed on a network Describe the purpose of an IP address. Explain how bandwidth and latency affect the performance of a network. Use bits per second (bps) to describe network speed. Explain how data are transmitted along copper and fibre-optic cables. Compare the performance of copper and fibre-optic cables and give examples of their use. Describe how high-speed broadband is delivered Describe how devices are connected on a wireless network. Describe the characteristics of bus, star and mesh network topologies. **Skills:** Categorise tasks according to the type of network used to carry them out. Construct expressions involving file size, transmission rate and time. Compare the performance of wired and wireless LANs and give examples of situations where one is preferable to the other. Summarise the	Give three reasons for connecting devices in a network . Explain how a LAN differs from a WAN. Define the term 'internet backbone'. Describe the function of a router. Explain how data are transmitted on a fibre-optic cable. State *two* advantages and *two* disadvantages of using wireless to connect devices on a LAN rather than cable. Construct an expression to calculate the time	8.2 Binary Bits and Bobs Unit CT1-6 Unit CT7-12 Unit CT13-18 Unit CT19-24 Unit CT25-30	Seneca. BBC Bitesize. cs4fn, Queen Mary, University of London (www.cs4fn.org). Computer Science Unplugged (http://csunplugged.org/ *BCS Glossary of Computing and ICT*, 13th edition (ISBN 9781780171500) A range of articles on teaching coding: www.edsurge.com/guide/teaching-kids-to-code **Python resources** Official Python documentation (also available through help in IDLE): www.docs.python.org/3/ Python summer school from Anglia Ruskin University is an excellent resource with videos and programming challenges: http://www.pythonschool.net/ Python code for kids is a clearly written summary of the Python language written in accessible language: www.pythondictionary.code-it.co.uk/

	characteristic of Wi-Fi, Bluetooth, RFiD, Zigbee and NFC and give examples of their use Draw and label a diagram of each topology Match descriptions to network topologies. Match descriptions of what they do to internet components (backbone, POP, NAP, router).	needed to transmit a file over a network. Explain why protocols are needed on a network.		‘Python in 10 minutes’ is a quick run through of the basic concepts: www.korokithakis.net/tutorials/python/ ‘Quintin Cutts – Too much doing, not enough understanding’ is a 20-minute video containing useful ideas and concepts on how to teach programming: www.youtube.com/watch?v=Pim4aYfiZiY