Computing

KS3 Curriculum Overview

Year 7

Year 7	Term 7.1	Term 7.2	Term 7.3	Term 7.4	Term 7.5	Term 7.6
Job Opportunities	**School Councillor, Human Resources, IT Security, School IT Technicians**	**Software Programmers in: Games, Operating Systems, databases, Application software, Mobile applications, Robotics, Telemetry analyst data in car racing**	**Web-Developer, Social Media Influencer, Journalist, Media Manager, Events Manager, Vlogger, Journalism**	**Financial Analyst, Administrative Assistant, Project Manager, Business Analysts, Data Journalist, Accountant, CAD Technician**	Entrepreneur, GCHQ, Cybersecurity, Cryptologist	**Software application developer, Web developer, Computer programmer, Software Engineer, Cloud Engineer, Machine Learning Engineer, Games Designer**
Topic	**Using computers safely, effectively and responsively**	**Programming data structures – using Flowol**	**Using Media**	**Modelling Data with Spreadsheets**	**History of Computing**	**Visual Programming essentials (Scratch)**
BIG Question	How do we use computers responsibly in the real-world?	How do programmers use flowcharts?	How can I use ICT support my school work? How can I use ICT effectively?	How to use spreadsheets to manage information?	Technology develops at an incredible rate – what were the starting points and what did they lead to?	How do we program a game?
Overall Aim (inc strand)	Identifying how to use online collaboration tools respectively. An introduction to the computing lab.	To develop logical reasoning and problem-solving skills, develop programming skills and explore the world of automatic, autonomous systems and robots	To develop core ICT skills which will support core curricula skills.	To develop an understanding of using numerical data to analyse its use and purpose	To use industry standard specialist software to enhance future career in IT	To develop basic programming techniques using blocks of code
Module Overview	Learning how to use the school network/Office365 appropriately, and learning how to collaborate online respectfully - building on their understanding of online dangers and how to stay safe online from Primary school.	Learners use Flowol to learn many programming elements; Sequences of instructions; Branching using decisions and Loops (infinite, or based on a condition or count). Programming visually with a flowchart allows the student to focus on the logic of their solution rather than the syntax of a written program.	Learners will develop their understanding of information technology and digital literacy skills whilst creating a digital product for a real-world problem. They will also explore concerns surrounding the use of other people's work, including licensing and legal issues	Learners will be introduced to basic spreadsheet techniques to create and format a simple financial model. Learners will experiment with "what if" scenarios and macros.	Learners are introduced to some of the key people who are responsible for some of the innovations we take for granted today. Learners will also conduct activities such as problem solving, code breaking and Boolean logic.	Building learners' confidence and knowledge of the key programming constructs. Learners will begin to understand basic concepts: sequencing, variables, selection, and count-controlled iteration. Putting it all together in a game.
Content Declarative Knowledge – 'Know What'	We have a responsibility to protect ourselves and others when using online environments • **File Management (files/folders) • Cloud (teams, Brightspace etc) • Email (send, cc, bcc, receive etc) • MS Office Skills • ESafety - Cyberbullying • Digital Footprint • Privacy • Identity Theft • Online sites, Chat rooms & social media • Presenting to an audience	Problems can be broken down into smaller, simpler pieces and that flowcharts control the flow of instructions • Computation thinking (decomposition) • Flowchart symbols (start, stop, process, input, output, decision) & interface • Sequencing (importance) • Sensors (light, heat, movement) • Sub-routines (use, advantages) • Actuators (motors) • Variables (created, edited, used)	That different software applications have different purposes • Word (fonts, paragraphs, tables) • Copyright & Creative Commons (differences) • Reliability & bias (Hyperlinks, Authors, Source, Information motive or purpose, date, common facts on other sites etc) • Plagiarism (citation, paraphrase) • Referencing (sourcing assets)	An introduction to spreadsheets and the concept of cell referencing. Learners will collect, analyse, and manipulate data, before turning it into graphs and charts • BIDMAS • Autofill (replicating) • Advanced formulae (sum, count, max, min, average, counta, if) • IF Statement • Conditional Formatting • Simple formulas (+ - * /) • Primary (collecting data yourself) & Secondary (other's collecting data) Research • Data (raw facts) & Information	Understand the key points in computer History, identifying the key people and explaining what they did to help in the development of computers. Understand why searching & sorting algorithms are needed and how they differ. • Cryptography – Alan Turing • Boolean Logic – George Boole • 1^{ST} Computers – Charles Babbage & Ada Lovelace • The Internet – Tim Berners-Lee • Searching Algorithms (Linear, Binary) • Sorting Algorithms (Insertion, Bubble, Merge)	That algorithms are instructions that can be developed and followed to solve a problem or to perform tasks. Developing an understanding of the sequence and selection constructs using block coding • Sequence • Selection • Count-controlled Iteration • Problem solving
Skills Procedural Knowledge – 'Know How'	Knowing how: **L1:** Know how to log on, create a secure password, create folders and organise work. Know how to follow the rules to stay safe.	Knowing how: **L1:** Identify everyday situations where computer control is used. **L2:** Identify flowchart symbols and understand how they are used to break down problems.	Knowing how: **L1:** Understand that different software applications have different purposes. Use a use word processing software to explore a range of formatting tools.	Knowing how: **L1:** Know the concept of spreadsheets and why spreadsheets are useful. Learn how to navigate a spreadsheet via its rows and columns and become familiar with the cell referencing system.	Knowing how: **L1:** Summarise who Alan Turing was? Describe the role of cryptography & how its used in history and its need for today.	Knowing how: **L1:** To understand the precise nature of instructions that computers need to execute.

	L2: Become familiar with common applications (accessing software/school network) and their personal work areas. Know how to Communicate respectfully via emails **L3:** Work successfully when collaborating online. How to make positive contributions to their online community. **L4:** Plan an effective presentation for an audience. Understanding the term 'cyberbullying' and the effects of cyberbullying **L5:** How to identify our friends online and how to spot impersonations	**L3:** Produce flowchart-based solutions for control systems that include sequences and loops. **L4:** Produce control solutions for problems that include subroutines. **L5:** Produce control solutions for problems that include variables	**L2:** Apply relevant formatting techniques to documents. Learners will be introduced to copyright law and learn about Creative Commons licensing. **L3**: Learners will look at techniques to use to help determine the credibility of a source, and then apply these skills by writing an article that could be real or fake. **L4:** Learners will be introduced to the concept of plagiarism. Learners will spend time looking at blog posts to evaluate their layout and content, then they will research their own cause and justify the credibility of their sources L5: Apply referencing techniques that credit authors appropriately. Construct a blog using appropriate software	**L2:** Learn how to perform calculations on data using basic formulas and cell references. Learn how to use the autofill tool to duplicate cells and continue a linear pattern **L3:** Discover the difference between data and information, and between primary and secondary sources of data. **L4:** Learn how to use functions to analyse data in a spreadsheet. As well as learning how to automatically create charts from data, learners will be introduced to four functions: SUM, MAX, MIN, and COUNTA **L5:** Understand three more functions — COUNTIF, AVERAGE, and IF — and to how they can sort and filter a spreadsheet. **L6**: Discover how to use conditional formatting, whereby the appearance of a cell changes automatically depending on the data it contains, according to rules that have been set	**L2:** Describe the role George Boole played in Computing. Complete truth tables linked to the laws of logic. **L3:** Describe the roles of Charles Babbage and Ada Lovelace played in Computing. Describe the need for sorting algorithms and explain how they work. **L4:** Describe the role of Tim Berners-Lee played in the Web. Describe the need for searching algorithms and explain how they work.	**L2:** Learners will be introduced to variables as well as the opportunity to get more confident with sequences. **L3:** Learners are introduced to the concept of selection statements and how they can be used to control the flow of a program. **L4:** Learners will be introduced to logical and comparison operators. Then build a Scratch program by adding new questions to subroutines. **L5:** Learners will be introduced to the concept of iteration and debugging.
Prior Domains	KS2 E-safety lessons Use of Common application Software	Logical thinking & problem-solving skills Present data effectively? (7.1) Programming constructs (sequence) Arithmetic & relational operations	Opportunities to share KS2 Computing lessons Searching the web (7.1)	Opportunities to share KS2 Computing lessons Basic Maths Skills	Opportunities to share KS2 Computing lessons Flowcharts (7.2) Problem solving, resilience and computational thinking skills	Programming constructs and problem-solving skills from KS2 Sequencing (7.2) Computational Thinking (7.2) Resilience

Year 8

Year 8	Term 1	Term 2	Term 3	Term 4	Term 5	Term 6
Job Opportunities	**Computer Engineer, System Admin, Back-end Software developer, Hardware designer, Research and Development Leaders,**	Computer Engineer, System Admin, Back-end Software developer, Hardware designer, Electronics	Software application developer, Web developer, Computer programmer, Software Engineer, Cloud Engineer, Machine Learning Engineer	Graphic Designer, Animator	Mobile app development, mobile app design	Machine Learning Engineer, Python Developer, Data Scientist, DevOps Engineer, FullStack Developer
Topic	Computer Systems	Bits & Binary	Visual Programming Cont..	Working with Digital Images	Algorithms – App Development	Text-based Programming
BIG Question	How do computers talk to each other?	Why is Binary so important?	How do we program a computer?	What are vector images?	How do we create an online gaming app?	How do we program a computer?
Aim	Explore fundamental elements that make up a computer system	Represent numbers and text using binary	Using subroutines to solve a problem	Use objects, layering and path manipulation to create vector graphics	Use event-driven programming to create a gaming app	Introduce students to the key computing principles and theory.
Module Overview	Learners will study what computer systems; the components.	Learners will study how data is stored on a computer as well as the logic that is required to make them work. Learners will be introduced to binary digits and how they can be used to represent text and numbers	Learners will build on their understanding of the control structures' sequence, selection, and iteration and develop their problem-solving skills. Learners will learn how to create their own subroutines, develop their understanding of decomposition, learn how to create and use lists, and build upon their problem-solving skills by working through a larger project at the end of the unit	Learners are introduced to graphics and graphic file types; exploring how bitmap and vector images are represented and stored by the computer.	Learners will play the role of designer to project manager to developer in order to create their own mobile app	Learners will build on their understanding of control structures' sequence, selection, and iteration and develop their problem-solving skills. Learners will build on their understanding of an algorithm from Year 7.

Content Declarative Knowledge – 'Know What'	That computers are made up of digital and many physical components • Input and Output devices • Internal/External devices • Storage devices • Embedded systems • Role of CPU	What binary is and why data must be stored in binary format. • Units of data (Bit, nibble, byte, KB, MB, GB) • Binary Basics – transistors, bases • Binary conversion • Binary addition • ASCII	• Subroutines • Condition-controlled loops • Lists • Decomposition	Different forms that digital graphics can take, and that different graphics can be used for different purposes	Problems can be broken down into smaller, simpler pieces • Decomposition • Event handling • Sequencing • Variables • Selection • Operators	Know that programming languages can be used to create programs for computers to perform certain tasks
Skills Procedural Knowledge – 'Know How'	Knowing how: • Software differs from hardware • How different hardware components of a PC work • Input and Output devices differ • Data is stored **L1:** Explain the difference between a general-purpose computing system and a purpose-built device. **L2:** Describe the function of the hardware components used in computing systems and how hardware components used in computing systems work together in order to execute programs. **L3:** Define what an operating system is and recall its role in controlling program execution. **L4:** Describe the NOT, AND, and OR logical operators, and how they are used to form logical expressions. **L5:** Provide broad definitions of 'artificial intelligence' and 'machine learning' **L6:** Explain the implications of sharing program code	Knowing how: • Computers can use binary input in complex ways • Logic gates work • To convert denary numbers to binary numbers and vice versa. • To add two binary numbers together and explain when an overflow error might occur • Characters are represented in binary. • List examples of representations • Recall that representations are used to store, communicate, and process information • Provide examples of how different representations are appropriate for different tasks • Recall that characters can be represented as sequences of symbols and list examples of character coding schemes • Measure the length of a representation as the number of symbols that it contains • Provide examples of how symbols are carried on physical media Explain what binary digits (bits) are, in terms of familiar symbols such as digits or letters • Measure the size or length of a sequence of bits as the number of binary digits that it contains • Describe how natural numbers are represented as sequences of binary digits terms of familiar symbols such as digits or letters • Measure the size or length of a sequence of bits as the number of binary digits that it contains • Describe how natural numbers are represented as sequences of binary digits • Convert a decimal number to binary and vice versa • Convert between different units and multiples of representation size • Provide examples of the different ways that binary digits are physically represented in digital devices •	Knowing how: **L1:** Define a subroutine and decomposition. **L2:** Identify where condition-controlled iteration can be used in a program and implement condition-controlled iteration in a program. **L3:** Evaluate which type of iteration is required in a program **L4:** Describe the need for lists. Define a list as a collection of related elements that are referred to by a single name. **L5:** To decompose a larger problem into smaller sub problems. To apply appropriate constructs to solve a problem	Knowing how: • Bitmap and Vector images are structured and their differences between them • To combine different images to make a logo • To combine images, graphics and text • To use a graphics editor to a fundamental degree	Knowing how: **L1:** Establish user needs when completing a creative project. Identify when a problem needs to be broken down (decomposition) **L2:** Use variables in an event-driven programming environment. **L3:** Identify and fix common coding errors. Establish user needs when completing a creative project. **L4:** Apply decomposition to break down a large problem into more manageable steps. Use a block-based programming language to create a sequence. **L5**: Be able to reflect and react to user feedback	Knowing how: **L1:** Write simple Python programs that display messages, assign values to variables, and receive keyboard input. **L2:** Use simple arithmetic expressions in assignment statements to calculate values **L3:** Use relational operators to form logical expressions. Use binary selection (if, else statements) to control the flow of program execution **L4:** Use multi-branch selection (if, elif, else statements) to control the flow of program execution. Describe how iteration (while statements) controls the flow of program execution **L5:** Use iteration (while loops) to control the flow of program execution.
Prior Domains	KS2 knowledge (input/output) Boolean Operators (7.3, 7.5, 8.1)	Interpret binary (7.4) Boolean Logic (7.5) Maths Skills	Programming constructs (7.2, 7.6) Problem solving & logical thinking skills	Copyright & Creative Commons (7.3)		All the programming concepts they used in previous units

Year 9

Year 9	Term 1	Term 2	Term 3	Term 4	Term 5	Term 6
Job Opportunities	**Cyber Security Analyst, Penetration Tester, Cyber Security Consultant, Technical Account Manager - Cyber Security**	**Sound Designers, Multimedia Artists, Digital Photography, Hardware Sales Representative, Hardware Designers, Electrical Engineers**	Network Service Technician, Network Systems Manager, Network Programmer/Analyst, Network Systems Engineer, Network Administrator, Network engineer	**Web developer, web designer, front end developer, Interface designer, software development**		**SQL Developer, SQL Database Administrator, Database Tester, SQL Data Analyst, Business/Financial Data Analyst, SQL Data Scientist, Data Modeler**
Baseline Assessment	**Cyber Security**	**Data Representation**	**Intro 2 Networks**	**HTML & Web Development**	**Python Programming with sequences of data**	**Database Development**
BIG Question	Why is our data so valuable to hackers?	How do media companies use Computers to construct their products?	How does the internet work?	How are webpages created?	How can we solve problems with efficient problems?	What do companies use databases for?
Aim	Identifying how users and organisations can protect themselves from cyber attacks	Represent images and sound using binary digits. Use graphics package to manipulate images to create a movie poster. Use a sound editing program to experiment with sound.	Recognising networking hardware and explaining how networking components are used for communication	Using HTML & CSS to create functional webpage(s)	Manipulating strings and lists	Introduction to database fundamentals. Build and interrogate your own software system
Module Overview	Learners will look at the issues surrounding the use of personal data/sensitive data and understand ways of keeping safe.	Learners will focus on making digital media such as images and sounds, and discover how media is stored as binary code.	Learners will build on their understanding of the internet from their daily lives. Learners begin by defining a network and addressing the benefits of networking, before covering how data is transmitted across networks using protocols, including gaining an understanding about the hardware required.	Learners will explore the technologies that make up the internet and World Wide Web. Exploring the building blocks of the World Wide Web, HTML, and CSS, they will investigate how websites are catalogued and organised for effective retrieval using search engines.	Learners will understand how data can be represented and processed in sequences, such as lists and strings.	Learners create and use of a single-table database and a simple relational database involving two tables in a one-to-many relationship. Learners will add records and make queries.
Content Declarative Knowledge – 'Know What'	Know techniques that cybercriminals use to steal data, disrupt systems, and infiltrate networks • What data is collected & how's it used • Privacy polices & GDPR • Social engineering & vulnerabilities • Hacking & exploitation • Attacks (brute force, DDOS etc) • Malware & bots • Risks (common threats) & Prevention methods	Know the underlying principles of digital representations that are applied in real settings and how to combine and manipulate images to create a graphical product for a purpose • Pixels • Resolution & colour depth • Image manipulation tools • Altitude (sampling rate, sample size) • Image quality • Sound quality	Know how networks have changed how we learn, play, work and communicate • Computer networks (LAN, WAN, PAN) and protocols • Hardware • Connection methods • Wired & Wireless • Performance factors • Security • WWW & internet services	Know how websites are planned, structured, designed and styled • HTML • CSS • Design • Development • Web forms • Testing	Know how to operations on sequences of data, that range from accessing an individual element to manipulating the entire sequence • Adding, removing, modifying items • Locating, counting occurrences • While loops • For loops	Identify databases used by organisations which are accessible to the public via the Internet • Tables • Queries • Input forms • Reports
Skills Procedural Knowledge – 'Know How'	Knowing how: **L1:** Explain the difference between data and information. Critique online services in relation to data privacy. **L2**: Recognise how human errors pose security risks to data. Implement strategies to minimise the risk of data being compromised	Knowing how: **L1:** Describe how an image can be represented as a sequence of bits. **L2:** Define key terms such as 'pixels', 'resolution', and 'colour depth'. **L3:** Describe how colour can be represented as a mixture of red, green, and blue, with a sequence of bits representing each colour's intensity. **L4**: Calculate representation size for a given digital sound, given its attributes.	Knowing how: **L1:** State that the Internet is a wide area network and the world wide web is part of the Internet **L2:** To understand examples of LANs and WANs. **L3:** State why some transmissions are encrypted, and use a simple algorithm to encrypt and decrypt a message **L4:** Understand advantages and	Knowing how: **L1:** Understand how web pages are constructed using HTML tags, and how they can be modified to start to resemble the websites they are accustomed to. **L2:** Display images within a web page. Apply HTML tags to construct a web page structure from a provided design **L3:** Describe what CSS is and use CSS to style static web pages. Assess the benefits of using CSS to style pages instead of in-line	Knowing how: **L1:** Use selection (ifelif-else statements) to control the flow of program execution. Locate and correct common syntax errors **L2:** Perform common operations on lists or individual items. **L3:** Use iteration (while statements) to control the flow of program execution.	Knowing how: **L1:** Create a database table using several fields with different data types. **L2:** Query the database using more than one criterion to find answers to user queries. **L3:** Create a frontend application menu with buttons linking to a form and a report. **L4:** Add features to an input form to make it

	through human error. **L3:** To Define hacking in the context of cyber security. **L4:** List the common malware threats Examine how different types of malware causes problems for computer systems **L5:** Compare security threats against probability and the potential impact to organisations	**L5:** Define 'compression', and describe why it is necessary.	disadvantages of different network topologies **L5:** Know the difference between peer-to-peer networks and client-server networks.	formatting. Use CSS to style static web pages instead of in-line formatting **L4:** Describe what a search engine is. Explain how search engines 'crawl' through the World Wide Web and how they select and rank results. **L5:** Discuss the impact of search technologies and the issues that arise by the way they function and the way they are used	**L4:** Use iteration (for statements) to iterate over list items. **L5:** Use variables to keep track of counts and sums. Combine key programming language features to develop solutions to meaningful problems	more user-friendly. Fully customise their input forms and reports. **L5:** Create the relationship between two linked tables and create a complex query which uses two tables in a relational database
Prior Domains	Email scams (7.1) Personal data (7.1) Copyright (7.3, 8.3)	Digital Images (8.3)	Cyber Security (9.1)	Networks (9.3) Searching the web (7.3, 7.5, Boolean 8.5)	Visual programming (7.2) (7.6) (8.2) Text-based Programming (8.6) Abstraction, Logic and algorithms – Computational thinking (7.2, 7.6, 8.2, 8.6)	Spreadsheet modelling (7.4)
	L1. Identify input/output/storage (HS&CS) L3. Using email (UCSER) L5. Communicate different ideas CG) L? (UCSER)		L2. Licencing (UM) L4. Searching the Web (UM)		Programming constructs (P) (CP) Lessons (CP, AD, CT, CP)	L1. Datatypes (AP) L2. Datatypes (AD)

Year 10

Year 10	Autumn Term	Spring Term	Summer Term
Topic	**Boolean Logic & Data Representation**	**Algorithms & Programming**	**Programming languages and Integrated Development Environments**
BIQ Question	How can we solve problems with efficient programs? Can everything be represented by 0s and 1s?	How can we think more like a computer? How can we solve problems with efficient programs?	How do high-level languages differ from low-level languages and why are they needed? What is an operating system and why do devices need them?
Overview	Learners will develop their knowledge about data that can be represented within a computer. Exploring Binary, Denary and Hexadecimal conversions, before moving onto image and sound representation. Learners will also begin to develop their skills using the Python 3 programming language. Learning about the essential programming foundations of Sequence, Selection and Iteration and how to apply these skills across a variety of different programming scenarios.	Learners explore the term algorithm and what different algorithms exists within the field of Computer Science to carry out a variety of different operations. They will learn how to represent these algorithms as flowcharts and ERL and will learn how to hand-trace through algorithms using a trace table Learners will continue to develop their programming skills by looking at the use of a variety of data types and data structures including the use of Arrays. Also, learning how to use subroutines and functions, ad understanding their differences. Students will learn how to apply these skills across a series of different, more advances programming scenarios. Learners will also develop their understanding of how databases can be used to store information in an organised way; how different entities must be stored in appropriate data tables and how keys can be used to identify and relate these tables together. Exploring Structured Query Language (SQL) to carry out database operations on SQL databases.	Learners dive into programming language classifications including translators and low-level languages. Testing and error handling is covered using practical examples, including the use of the common tools and functions of an IDE. Learners will also learn about the different categories of software and some of the key functions of the operating system
Content Declarative Knowledge – 'Know What'	**2.4 – Boolean Logic** **2.4.1 – Boolean Logic** • Knowledge of the truth tables for each logic gate • To create, complete or edit logic diagrams and truth tables for given scenario • Use more than 1 logic gate	**2.1 – Algorithms** **2.1.1 – Computational Thinking** • Understand the concepts of Abstraction, Decomposition **2.1.2 – Designing, creating and refining algorithms** • Complete, write or refine an algorithm to solve problems • Identify syntax/logic errors in code and suggest fixes	**2.5 – Programming Languages and Integrated Development Environments** **2.5.1 – Languages** • Characteristics and purpose of different levels of High level and low-level programming languages: • The purpose of translators • Translation of source code through interpreter and compilation,

	1.2 Memory and Storage **1.2.3 – Units:** • How data needs to be converted into a binary format to be processed by a computer • Data capacity and calculation of data capacity requirements **1.2.4 – Data Storage** **Numbers:** • How to convert between the different number bases i.e. binary, denary and hexadecimal numbers. • Understand the effect of a binary shift (both left or right) on a number. **Character:** • Character set' using ASCII and UNICODE **Images:** • Pixels and their representation of a colour binary code representation. • The effect on image size and quality when changing colour depth and resolution • Metadata **Sound:** • Analogue sounds must be stored in binary; sound is sampled in hertz. Know • that the number of bits required to store each sample is the bit rate. **1.2.5 – Compression:** • Advantages and disadvantages of each type of compression • Effects on the file for each type of compression **2.2 – Programming fundamentals** **2.2.1 – Programming fundamentals** • The use of variables, constants, operators, inputs, outputs and assignments • The use of the three basic programming constructs used to control the flow of a program using Sequence, Selection and Iteration (count- and condition-controlled loops) **2.2.2 – Data types** • The common arithmetic operators along with the common Boolean operators AND, OR and NOT.	• Create and use trace tables to follow an algorithm **2.1.3 – Searching and sorting algorithms** • The main algorithms to sort and search data, bubble, insertion and merge sort, linear and binary searches. **2.2.3 – Additional programming techniques** • The use of basic string manipulation • The use of basic file handling operations: • Open, Read, Write and Close • The use of records to store data • The use of arrays (or equivalent) when solving problems, including both one-dimensional and two-dimensional arrays • How to use sub programs (functions and procedures) to produce structured code • Random number generation **2.2.3 – Additional programming techniques** • The use of records to store data • The use of SQL to search for data **2.3 – Robust programs** **2.3.1 – Defensive design considerations:** • Anticipating misuse • Authentication • Input validation • Maintainability: use of sub programs, naming conventions, Indentation and commenting	**2.5.2 – The Integrated Development Environment (IDE)** • Common tools and facilities available in an • Integrated • Development Environment (IDE): Editors, • Error diagnostics, Run-time environment • Translators • Techniques used to trap errors, types of errors. Breakpoints, stepping variable watches. **1.5 – Systems Software** **1.5.1 – Operating systems** **The purpose and functionality of operating systems:** • User interface, Memory management and multitasking. • Peripheral management and drivers, user management and file management **1.5.2 Utility software** • The utility system software on a computer system understanding typical routines such as defragmentation, firewall and backup **2.3.2 – Testing** • The purpose of testing • The different types of tests, i.e. iterative and final/terminal • Understand what is meant be a syntax error and a logic error. • Know and understand the different test types, i.e. boundary, invalid, valid and erroneous.
Skills Procedural Knowledge – 'Know How'	**2.4 Boolean Logic** • How to use and create simple logic diagrams using the operators AND, OR and NOT Truth tables • Combine Boolean operators using AND, OR and NOT • Apply logical operators in truth tables to solve problems **1.2 Memory and Storage** **1.2.3 – Units:** • How and why data must be stored in binary format • Calculate capacity of devices, required capacity for different types of files i.e. text, sound, image. **1.2.4 – Data Storage** **1.2. Number:** • How to convert positive denary whole numbers to binary numbers (1 byte) and vice versa • How to add two binary integers together (up to and including 8 bits) and explain overflow errors which may occur • How to convert positive denary whole numbers into 2-digit hexadecimal numbers and vice versa • How to convert binary integers to their hexadecimal equivalents and vice versa • Binary shift	**2.1 – Algorithms** **2.1.1 – Computational Thinking** • How problems can be broken down into small tasks and begin to develop skills in programming • How to define and refine problems **2.1.3 – Searching and sorting algorithms** • Understand how the main sort and search algorithms are applied and in what circumstances they are useful. • Demonstrate the order in which they can reorganise data and why this might be useful in a given set of • circumstances **2.1.2 – Designing, creating and refining algorithms** • Know how to Identify the inputs, processes, and outputs for a problem • Understand pseudocode and flow diagrams to interpret, correct or complete algorithms • Abstract and decompose a problem to its simplest level, where code can be designed and written. • Test and validate inputs and outputs from the program • Identify common errors	**Programming languages and Integrated Development Environments** **2.5.1. Languages** • The differences between high- and low-level programming languages • The need for translators • The differences, benefits and drawbacks of using a compiler or an interpreter **2.5.2 The Integrated Development Environment (IDE)** • Using their own practical knowledge of the tools that an IDE provides • How each of the tools and facilities can be used to help a programmer develop a program **Systems software** **1.5 Operating systems and 1.5.2 Utility software** • What each function of the operating system does their purpose and how they can be configured to maintain computer systems • How data is transferred between devices and the processor • Purpose of utility software and why they are required to maintain and repair computer systems.

	Character: • How characters are represented in binary • How the number of characters stored is limited by the bits available • The differences between and impact of each character set and that character sets are logically ordered **Images:** • Pixels represented in binary • The effect of colour depth and resolution on the quality and size of an image file. **Sound:** • How sound can be sampled and stored in digital form • The effect of sample rate, duration and bit depth on the playback quality and the size of a sound **file** **1.2.5 – Compression:** • How and why there is a need for compression • The differences between lossy and lossless compression. **2.1- Designing, creating and refining algorithms** • Know how to Identify the inputs, processes, and outputs for a problem • Understand pseudocode and flow diagrams to interpret, correct or complete algorithms • Abstract and decompose a problem to its simplest level, where code can be designed and written. • Test and validate inputs and outputs from the program • Identify common errors • Use trace tables to predict the logic flow in an algorithm. **2.2 – Programming fundamentals** **2.2.1 – Programming fundamentals** • How to use Python to solve problems using a range of operators, data structures and calling functions. • Understand pseudocode and flow diagrams to interpret, correct or complete algorithms • Abstract and decompose a problem to its simplest level, where code can be designed and written. • Test and validate inputs and outputs from the program **2.2.2 Data Types** • Switch between datatypes for given scenario • Identify where this is useful and why	• Use trace tables to predict the logic flow in an algorithm **2.1.3 – Searching and sorting algorithms** • Understand how the main sort and search algorithms are applied and in what circumstances they are useful. • Demonstrate the order in which they can reorganise data and why this might be useful in a given set of • Circumstances. **2.2.3 – Additional programming techniques** • Practical use of the additional programming techniques in a high-level language with given problems. • Within their program code know how to manipulate strings using concatenation and slicing. • Design and write program code to manipulate both 1D and 2D array data structures to solve problems. Effectively use slicing on the arrays. • Know and understand the difference between functions and procedures and effectively use these in their program code solutions. **2.2.3 – Additional programming techniques** • Using SQL commands SELECT FROM WHERE to search data **2.3 – Robust programs** **2.3.1 – Defensive Design** • Understanding of the issues a programmer should consider to ensure that a program caters for all likely • input values including invalid data • Authentication to confirm the identity of a user • Practical experience of designing input validation and simple authentication • Understand why commenting is useful and apply this appropriately	**Paper 2 – Algorithms and programming** • Learners will know how to design a solution to a given set of problems that will vary in their complexity • They will abstract and decompose a problem to its simplest level, where code can be designed and /written. • Test and validate inputs and outputs from the program. **2.3.2 – Testing** • Know how to write an effective test plan using suitable test data. • The difference between testing modules of a program during development and testing the program at the end of production
Prior domains	7.2 Sequencing with Flowol, 7.5 History of Computing, 7.6 Visual Programming, 8.2 Bits and Binary, 8.3 Visual Programming, 8.6 Text-based Programming, 9.2 Images and Sound and 9.5 Python Programming and Autumn Term- Bebras Test	7.1 UCSER, 7.2 Sequencing with Flowol, 7.5 History of Computing, 7.6 Visual Programming, 8.3 Visual Programming, 8.6 Text-based Programming, 9.4 HTML Development and 9.5 Python Programming.	7.6 Visual Programming, 8.3 Visual Programming, 8.6 Text-based Programming and 9.5 Python Programming

Year 11

Year 11	Autumn Term	Spring Term	Summer Term
Topic	**Systems Architecture, Memory & Storage**	**Networks, connections & Protocols**	
BIQ Question	How can we design the fastest computer in the world? How does primary memory and secondary memory differ?	Why can data travel securely across a network? Can all data be hacked when it travels across the internet?	Do computers really make the world a better place?
Overview	Learners explore the essential hardware and software required to create a functional computer system. Learning about key components including the Motherboard, The CPU, RAM, ROM and Secondary Storage.	Learners start to build an understanding of how computers can be networked together to perform tasks and to share data and resources. Building an understanding of different network topologies and developing their understanding of how the internet works including the different stages of the TCP/IP protocol stack.	Learners build an understanding of the different ethical issues which surround the use of computers including the digital divide and appropriate use of computers. They will also learn how this links into specific legislation regarding the safe and legal; use computers e.g. GDPR, The Computer Misuse Act. Learners will also explore the environmental issues surrounding the

		Learners will also explore cyber security threats which computer users face on a daily basis. Learning about the different categories of threats and about how organisations and businesses can directly address and prepare for these threats during their system development. They will also learn about social engineering techniques used to gain access to users' personal data and some of the techniques used to avoid these.	development of technology from the mining or raw materials up to the consumption of energy used to charge and use these devices By this time, learners will be preparing for their summer assessments looking specifically at exam technique and utilising existing resource such as past exam papers to develop their skills and knowledge across the breadth of their specification
Content Declarative Knowledge – 'Know What'	**1.1 - Systems Architecture** **1.1.1 - Architecture of the CPU** • Von Neuman architecture i.e. MAR (Memory Address Register), MDR (Memory Data Register) Program Counter and Accumulator • The purpose of the CPU – The FDE (Fetch Decode Execute) cycle. • Students should be able to take a series of simple low-level instructions and understand how the processor uses the various general-purpose registers to do calculations during the FDE cycle. • How RAM and ROM interact with the CPU and what they contain during operation of • the CPU **1.1.2 - CPU Performance** • How common characteristics of CPUs affect their performance: Clock speed, cache size and number of cores **1.1.3 - Embedded systems** • The purpose and characteristics of embedded systems **1.2 - Memory and storage** **1.2.1 - Primary storage** • Why computers have primary storage RAM and ROM • Key characteristics of RAM and ROM • Why virtual memory may be needed in a system **1.2.2 - Secondary storage** • The need for secondary storage • The differences between each type of storage device/medium • Compare advantages/disadvantages for each storage device **2.2 Algorithms and programming** **2.2 Programming fundamentals, 2.2 Data types, 2.2. Additional programming techniques & 2.3 Testing** • Students will undertake practice their programming skills independently – mini style challenges	**1.3 Computer networks, connections and protocols** **1.3.1 - Networks and topologies** • The types of networks – LAN and WAN • The topologies of a network – Star and Mesh. • The advantages and disadvantages of types of hardware and where they are needed in different • circumstances. • Factors affecting performance such as bandwidth and number of devices. • The different roles of computers in a client-server and a peer-to peer network. • The Internet as a worldwide collection of computer networks **1.3.2 - Wired and wireless networks, protocols and layers** • The differences between wired and wireless connections • Encryption used on a network • IP addressing and MAC addressing • Communication standards and protocols • Layers used in protocols **1.4 Network Security** **1.4.1 Threats to computer systems and networks** • Forms of attack: ○ Malware ○ Social engineering, e.g. phishing, people as the 'weak point' ○ Brute-force attacks ○ Denial of service attacks ○ Data interception and theft ○ The concept of SQL injection **1.4.2 – Identifying threats and vulnerabilities** Common prevention methods: • Penetration testing • Anti-malware software • Firewalls • User access levels • Passwords • Encryption • Physical security **2.2 Algorithms and programming** **2.2 Programming fundamentals, 2.2 Data types, 2.2. Additional programming techniques & 2.3 Testing** • Students will undertake practice their programming skills independently – mini style challenges	**1.6 - Ethical, legal, cultural and environmental impact** **1.6.1 - Ethical, legal, cultural and environmental impact** • The laws that govern how computers, data and information can be used. • The moral and ethical dilemmas of computer use, including environmental impacts, social well-being. • Software licences (i.e. open source and proprietary)
Skills Procedural Knowledge – 'Know How'	**1.1 Architecture of the CPU** • How the FDE cycle is accomplished. The effects of changing any of the common characteristics of CPU's on system performance, either individually or in combination • The links with memory (RAM) and how RAM is used. They will know the purpose and function of the MAR, MDR, PC, CU, ALU, Accumulator and be able to know how each register works during the FDE cycle.	**1.3 Computer networks, connections and protocols** **1.3.1 - Networks and topologies** • When building a network, what hardware and software is required and the weaknesses of the different types • Factors such as bandwidth and number of devices can impact network performance.	**1.6 - Ethical, legal, cultural and environmental impact** **1.6.1 - Ethical, legal, cultural and environmental impact** • Social, ethical and moral issues regarding environment, health and wealth. • Under what circumstances the law can be applied and the sort of activities they are intended to combat. • The general problems encountered with the blind application of technology.

	• The difference between storing data and an address • How RAM interacts with the CPU, how RAM and ROM is measured, their various uses **1.1 CPU performance** • How cache size, clock speed and cores affect the performance of the CPU. • The effects of changing any of the common characteristics on system performance, either individually or in combination. **1.1 Embedded systems** • Typical characteristics of embedded systems **1.2 Primary storage** • Why computers have primary storage RAM and ROM • How virtual memory works. **1.2 Secondary storage** • Understand how secondary storage is used and which secondary storage media is appropriate for a given set of circumstances • To know why computers, have secondary storage and to identify the differences between each type of storage. They will then apply their knowledge in context within scenarios **2.2 Algorithms and programming** **2.2 Programming fundamentals, 2.2 Data types, 2.2. Additional programming techniques & 2.3 Testing** • Learners will know how to design a solution to a given set of problems that will vary in their complexity • They will abstract and decompose a problem to its simplest level, where code can be designed and written. Test and validate inputs and outputs from the program.	• Clients make request for services into a client-server network. • DNS's convert IP addresses • Cloud computing services work along with its advantages and disadvantages **1.3.2 - Wired and wireless networks, protocols and layers** • Compare the benefits and drawbacks of wired versus wireless connection • Encryption and why it is needed on a network. • The principle of encryption to secure data across network connections • How the internet operates through routers, packet switching and IP addresses. • The internet and the protocols that make it work, with particular reference to the TCP/IP model and how each layer in the stack contributes a different protocol that allows for network and internet communication. POP3, HTTPS, HTTP, FTP, SMTP, IMAP **1.4 Network Security** **1.4.1 Threats to computer systems and networks** • How systems are vulnerable to attacks and how each attack is used and why **1.4.2 – Identifying threats and vulnerabilities** • How to use different precautions to reduce vulnerabilities posed by the different threats **2.2 Algorithms and programming** **2.2 Programming fundamentals, 2.2 Data types, 2.2. Additional programming techniques & 2.3 Testing** • Learners will know how to design a solution to a given set of problems that will vary in their complexity They will abstract and decompose a problem to its simplest level, where code can be designed and written. Test and validate inputs and outputs from the program.	• The need to license software and the purpose of a software licence
Prior domains	8.1 Computer Systems,	7.3 Networks, 9.4 HTML Development,	