COMPUTING KNOWLEDGE & SKILLS PROGRESSION – We are Programmers

"Technology will never replace great teachers but technology in the hands of great teachers is transformational." George Couros

We want our children to be highly equipped to respond to the fast moving and rapidly evolving technological world. Our Computing curriculum teaches children to understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation. Computer science and applying its principles into programming is taught from Year 3 to Year 6 as our children learn to analyse problems and evaluate and apply information technology analytically to solve problems. We use *Kapow* to support our teaching of the Computing curriculum and *National Online Safety* to support the teaching of online safety.

We have designed a curriculum that teaches a progression of knowledge, skills and understanding. New learning is taught in each learning programme and specific skills are repeated and practised in subsequent terms to ensure a depth of knowledge. Our programmes of learning are taught with a cross curricular perspective, when appropriate, to allow children to understand the interconnections between technology and other subjects. Science, mathematics and design are strongly connected to the curriculum and children are taught to recognise and learn from these connections.

The curriculum has been designed so that online safety is taught each term and these are joined by a termly programme of study which is consistent across both key stages.

Our pupils learn to:

- understand what algorithms are; how they are implemented as programs on digital devices; and that programs execute by following precise and unambiguous instructions
- create and debug simple programs
- use logical reasoning to predict the behaviour of simple programs
- use technology purposefully to create, organise, store, manipulate and retrieve digital content
- design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts
- use sequence, selection, and repetition in programs; work with variables and various forms of input and output
- use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs
- understand computer networks including the internet; how they can provide multiple services, such as the world wide web; and the opportunities they offer for communication and collaboration
- use search technologies effectively, appreciate how results are selected and ranked, and be discerning in evaluating digital content
- select, use and combine a variety of software (including internet services) on a range of digital devices to design and create a range of programs, systems and content that accomplish given goals, including collecting, analysing, evaluating and presenting data and information

Online Safety

- recognise common uses of information technology beyond school
- use technology safely and respectfully, keeping personal information private;
- identify where to go for help and support when they have concerns about content or contact on the internet or other online technologies.
- use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact.

Key Stage 2 Computing Curriculum

- can understand and apply the fundamental principles and concepts of computer science, including abstraction, logic, algorithms and data representation
- can analyse problems in computational terms, and have repeated practical experience of writing computer programs in order to solve such problems
- can evaluate and apply information technology, including new or unfamiliar technologies, analytically to solve problems
- are responsible, competent, confident and creative users of information and communication technology.

Pupils should be taught to:

- design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts
- use sequence, selection, and repetition in programs; work with variables and various forms of input and output
- use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs
- understand computer networks including the internet; how they can provide multiple services, such as the world wide web; and the opportunities they offer for communication and collaboration
- use search technologies effectively, appreciate how results are selected and ranked, and be discerning in evaluating digital content
- select, use and combine a variety of software (including internet services) on a range of digital devices to design and create a range of programs, systems and content that accomplish given goals, including collecting, analysing, evaluating and presenting data and information
- use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact

A journey through our Computing curriculum

	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer2
Year 3	**Computing systems and networks 1** Networks and the internet Introduction to the concept of networks, learning how devices communicate. Identifying components, learning how information is shared and exploring examples of real-world networks. (3 lessons)	**Programming** Programming Scratch Using the more advanced computer-based application called 'Scratch', learning to use repetition or 'loops' and building upon skills to program; an animation, a story and a game. (2 lessons)	**Computing systems and networks 2** Emailing Learning how to send emails with attachments and how to be a responsible digital citizen by thinking about the contents of what is sent. (2 lessons)	**Computing systems and networks 3** Journey inside a computer. Assuming the role of computer parts and creating paper versions of computers helps to consolidate an understanding of how a computer works, as well as identifying similarities and differences between various models. (2 lessons)	**Creating media** Video trailers (Option 2: Using ipads). Developing filming and editing video skills through storyboarding and creation of book trailers. (3 lessons)	**Data handling** Comparison cards databases Using the theme of a 'Comparison cards game' (based on the popular game, Top Trumps), to understand what a database is by learning the meanings of records, fields and data. Further exploration will lead to the development of the ideas of sorting and filtering.
Year 4	**Computing systems and networks** Collaborative learning. Working collaboratively in a responsible and considerate way as well as looking at a range of collaborative tools. (3 lessons)	**Programming 1** Further coding with Scratch. Using variables in coding. Options for both Google and Microsoft schools. (2 lessons)	**Creating media** Website design and HTML Children develop their research, word processing, and collaborative working skills whilst learning how web pages and web sites are created, exploring how to change layouts, embed images and videos and link between pages. This unit has options for both Google and Microsoft-based devices. Editing the HTML and CSS of a web page to change the layout of a website and the text and images. (4 lessons)		**Programming 2** Computational thinking Plugged and unplugged activities to develop the four areas of computational thinking. (3 lessons)	**Data Handling** Investigating weather. Option 2: Microsoft Office 365). Researching and storing data using spreadsheets; designing a weather station that gathers and records data; learning how weather forecasts are made and using green screen technology to present a weather forecast. Options for both Google and Microsoft schools. (2 lessons)
Year 5	**Computing systems and networks** Search engines.	**Programming 1** Programming music Option 1 Sonic pi	**Data Handling & Programming 2** **Mars Rover 1** Data transfer and binary code.		**Creating media** Stop motion animation.	**Skills showcase** 3D design skills Mars Rover 2

	Using keywords and phrases, identifying inaccurate information, learning page rank works as well. These lessons are available for both Microsoft and Google schools. (3 lessons)	Option 2 Scratch Applying programming skills to create sounds and melodies leading to a battle of the bands performance. (2 lessons)	**Micro:bit** The meaning and purpose of programming. (4 lessons total)		Storyboarding ideas, taking photographs and editing to create a video animation. (3 lessons)	3D design skills. (2 lessons)
Year 6	**Computing systems and networks** Bletchley Park Code breaking and password hacking. (3 lessons: 1-3)	**Programming** Intro to Python Using the programming language of Python. (2 lessons: 1-4)	**Data Handling** Big data 1. Barcodes, QR codes and RFID. (2 lessons)	**Creating media** History of computers. Children write, record and edit radio plays set during WWII, look back in time at how computers have evolved and design a computer of the future. Options for schools that use Google or Microsoft. (Option 2: Microsoft Office 365) (2 lessons)	**Data Handling** Big data 2. Data usage and smart schools. (2 lessons)	**Skills showcase** Inventing a product Designing a product, pupils: evaluate, adapt and debug code to make it suitable and efficient for their needs; use a software program to design their products; create their own websites and video adverts to promote their inventions. (3 lessons)

Vocabulary

	Year 3	Year 4	Year 5	Year 6
Autumn				
Vocabulary	• Cables • Component • Connection • Corrupted • Data • Desktop • Device • DSL • Fibre • File • Internet • Laptop	• Animations • Average • Bar chart • Collaboration • Comment • Contribution • Data • Edited • Email account • Format • Freeze • Icon	• Algorithm • Appropriate • Copyright • Correct • Credit • Data leak • Deceive • Fair • Fake • Inappropriate • Incorrect • Index	• Acrostic Code • Brute force hacking • Caesar cipher • Chip and pin system • Cipher • Code • Combination • Contribute • Convince • Date shift cipher • Discovery • Hero

	• Network • Network map • Network switch • Packets • Radio waves • Router • Server • Submarine cables • Tablet • Text map • The Cloud • Web server • Website • Website trackers • WiFi • Wired • • Algorithm • Animation • Application • Code • Code block • Coding application • Debug • Decompose • Interface • Game • Loop • Predict • Program • Remixing code • Repetition code • Review • Scratch • Sprite • Tinker	• Images • Insert • Link • Multiple choice • Numerical data • Pie chart • Presentations • Resolved • Reviewing comments • Share • Slides • Software • Spreadsheets • Suggestions • Survey • Teamwork • Broadcast block • Code blocks • Conditional • Coordinates • Decomposition • Features • Game • Information • Negative numbers • Orientation • Parameters • Position • Program • Project • Script • Sprite • Stage • Tinker • Variables	• Information • Keywords • Network • Privacy • Rank • Real • Search engine • TASK • Web crawler • Website • Error • Instructions • Loop • Melody • Mindmap • Music • Output • Performance • Pitch • Plan • Play • Predict • Programming • Repeat • Rhythm • Scratch • Soundtrack • Spacing • Tempo • Timbre • Tinker • Tutorials	• Invention • Nth Letter Cipher • Password • Pig Latin • Pigpen cipher • Present • Scrambled • Secret • Secure • Technological advancement • Trial and error • Algorithm • Code • Command • Design • Import • Indentation • Input • Instructions • Loop • Output • Patterns • Random • Remix • Repeat • Shape
Spring				
Vocabulary	• Attachment • Bcc (Blind carbon copy) • Cc (Carbon copy) • Compose • Content	• Assessment • Audience • Checklist • Collaboration • Content	• 8-bit binary • Addition • ASCII • Binary code • Boolean	• Algorithms • Barcode • Binary • Boolean • Brand

	• Cyberbullying • Document • Domain • Download • Email • Email account • Email address • Emoji • Emotions • Fake • Font • Genuine • Hacker • Icons • Inbox • Information • Link • Log in • Log out • Negative language • Password • Personal information • Positive language • Algorithm • Assemble • CPU (central processing unit) • Data • Decompose • Desktop • Disassemble • GPU (graphics processing unit) • Hard drive • HDD (hard disk drive) • Infinite loop • Input • Keyboard • Laptop • Memory • Microphone • Monitor • Mouse • Output • Photocopier	• Contribution • Create • Design • Embed • Evaluate • Features • Google Sites • Hobby • Homepage • Hyperlinks • Images • Insert • Online • Plan • Progress • Published • Record • Review • Style • Subpage • Tab • Theme • Web page • Code • Component • Content • Copyright • CSS • End tag • Fake news • Hacking • Heading • Headline • Hex code • HTML • Input • Internet browser • Output • Paragraph • Permission • Remixing • Script • Start tag	• Byte • Communicate • Construction • CPU • Data transmission • Decimal numbers • Design • Discovery • Distance • Hexadecimal • Input • Instructions • Internet • Mars Rover • Moon • Numerical data • Output • Planet • Radio signal • RAM • Research • Scientist • Sequence • Connection • Create • Debug • Decompose • Designing • Desktop • Device • Download • Images • Input • Instructions • Laptop • Load • Loop • Micro:bit • Outputs • Pairing • Pedometer • Polling • Predict	• Chips • Commuter • Contactless • Data • Encrypted • Infrared • Magic Band • Privacy • Proximity • QR code • QR scanner • Radio waves • RFID • Signal • Systems/data analyst • Transmission • Wireless • Background noise • Byte • Computer • Devices • File • FX • Gigabyte • Graphics • Hard drive • Hardware • Kilobytes • Megabyte • Memory storage • Mouse • Operating system • Overlay • Play • Processor • Radio play • RAM • Raspberry Pi • Record • Reverb • ROM • Script • Smartphone

	• Program • QR Code • RAM (random access memory) • ROM (read only memory) • Storage • Tablet device • Technology • Touchscreen	• Tags • Text • URL • Webpage	• Program • Repetition	• Sound • Sound effects
Summer				
Vocabulary	• Application • Camera angle • Clip • Cross blur • Cross fade • Cross zoom • Desktop • Digital device • Dip to black • Directional wipe • Edit • Film • Film editing software • Graphics • Import • Key events • Laptop • Music • Photo • Plan • Recording • Sound effects • Storyboard • Time code • Trailer • Transition • Video • Voiceover • Categorise • Category • Chart • Data	• Abstraction • Algorithm • Code • Computational thinking • Decomposition • Input • Logical reasoning • Output • Pattern recognition • Script • Sequence • Variable • Accurate • Backdrop • Climate zone • Cold • Collaboration • Condensation • Cylinder • Degrees • Evaporation • Extreme weather • Forecast • Heat sensor • Lightning • Measurement • Pinwheel • Presenter • Rain • Satellite • Script • Sensitive	• Animation • Animator • Background • Character • Decomposition • Design • Digital device • Edit • Evaluate • Flip book • Fluid movement • Frames • Model • Moving images • Onion skinning • Still images • Stop motion • Storyboard • Thaumatrope • Zoetrope • 3D • Algorithm • Binary image • CAD • Compression • CPU • Data • Drag and drop • Fetch, decode, execute • ID card • Input • JPEG	• Big Data • Bluetooth • Corrupted • Data • Energy • GPS • Improve • Infrared • Internet of Things • Personal • Privacy • QR codes • Revolution • RFID • SIM • Simulation • Smart city • Smart school • Stop motion • Threat • WiFi • Wireless • Design • Edit • Electronic • Evaluate • Facts • Image rights • Images • Influence • Information • Inputs

	• Database • Excel • Fields • Filter • Graph • Information • Interpret • PDF • Questionnaire • Record • Representation • Sort • Spreadsheet	• Sensor data • Solar panel • Tablet/Digital camera • Temperature • Thermometer • Tornado • Warm • Weather	• Memory • Online community • Operating system • Output • Pixels • RAM • Responsible • RGB • ROM • Safe	• Loops • Manipulation • Opinions • Output • Photos • Product • Program • Repetition • Screenshot • Search engine • Selection • Sequence

What our pupils have experienced - a journey through MIANS KS1 Computing Curriculum

	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
Year 1	Create simple sequence algorithms using symbols (e.g. Bee-Bot cards).	Programme (code) a computer using a number of steps in order before pressing the 'run' button. Identify where in the code or algorithm bug/ problem occurs.	Give explanations for what a programme will do.	Use a range of digital devices to create and store content e.g taking a photo, videoing, artwork. Create original content using digital devices.	Show awareness of how IT is used for communication beyond school.	How to keep safe when using digital technology. Explain what to do to keep safe. To know that information on the internet can be seen by others. What to do if disturbing content is seen online at school or home.
Year 2	Read and follow written sequence algorithms (forward 3, right 90)	Create a simple programme on screen with a particular goal or purpose in mind. Independently debug any errors in own code.	Give logical explanations for what a programme will do.	Use a range of digital devices to create store and retrieve content. Edit original digital content using a range of technologies.	Name a number of ways IT is used beyond school e.g. sharing work on online communities, web, email, analyse data.	How to keep safe on the internet and explain this. Show respect to others while using digital technology. Respect others' rights, privacy and intellectual property. Know not to share personal information online. Have a range of strategies for dealing with concerns over content or contact online.

ASPIRATIONS FOR THE FUTURE

Pupils develop an understanding of how subjects and specific skills are linked to future jobs. Here are some of the jobs you could aspire to do in the future as a programmer:

Cyber security analyst, Software engineer, Systems analyst, Web designer/developer, UX designer, Game designer/developer, Forensic computer analyst, Database administrator, IT consultant.

Our feeder high school snapshot Computing curriculum:

Jane Austin

Year	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
7	Online Safety	Computer Systems	Computational Thinking and Algorithms	Computational Thinking and Algorithms	CAD/CAM	CAD/CAM
8	Cyber Security	Data Handling	Programming Fundamentals	Programming Fundamentals	DT Project	DT Project
9	Networking	Data Representation	Data Representation	Programming with confidence	DT Project	DT Project
10	Data Representation	Programming Fundamentals	Software & Secondary Storage	Wired & Wireless Networks and Network Security	Computational Logic	Systems Architecture
11	Impacts of Technology	Translators and facilities of language	Producing robust programs	Extended Programming project	Revision and Exam Practice	Revision and Exam Practice

Open Academy

Key Stage 3 Computing

We have developed a curriculum across Key Stage 3 that follows the National Computing curriculum. It develops and challenges our students and includes but is not limited to: E-safety, sound editing, digital imaging using Adobe software. Desktop publishing, word processing, spreadsheet modelling and formal electronic communication using Microsoft Office. 2D/3D Animation using Alice/ Sketchup /Blender and both visual and text based programming in Scratch/Alice, MIT App Inventor and Small Basic/C# respectively.

Sprowston Community Academy

Computer Science at Key Stage 3

At Key Stage 3, pupils are placed in mixed-ability classes and receive a one-hour weekly lesson. Our Key Stage 3 curriculum lasts for three years and covers all areas of the National Curriculum. It has been carefully sequenced so that the core topics are revisited yearly, and we use a spiralled curriculum approach so that a deeper knowledge can be developed as topics are revisited each year. The core topics that are covered each year including Computational Thinking, Python Programming, Data Representation, Networking and Computer Systems. In addition, pupils will cover units throughout Key Stage 3, including Physical Computing (using the BBC Micro:Bit), Spreadsheet Modelling, Sound & Video Production, Mobile App Development and Web Development.

The Hewitt Academy

Hewett Academy Curriculum Summary 2022-23 Computer Science KS3 & Yr10

	Autumn I	Autumn II	Spring I	Spring II	Summer I	Summer II
Year 7	Rotation		Rotation		Online Safety	Computer Systems
Year 8	Rotation		Rotation		Cyber Security	Spreadsheets
Year 9	Rotation		Rotation		Networks	Data Representation
Year 10	Computer data representation + algorithms	Python programming (selection and iteration) + computer memory	Computer hardware design + function programming in Python	Programming with Arrays + software	Networking + programming using validation and error testing	Programming project: Dice Roll project. + Sorting and searching algorithms

CNS

City Academy

INTRODUCTION

As we live in a digital age, most industries rely on data and software programmes. Computer Science & IT impacts everything, from scientific research to health development, transport, banking, communications, you name it.

Computer Science	Algorithms	Programming & Development	Data & Data Representation	Communications & Networks	Hardware & Processing	Information Technology
Beginning	With support: • I design solutions (algorithms) that use repetition and two-way selection i.e. if, then and else. • I use diagrams to express solutions. • I use logical reasoning to predict outputs, showing an awareness of inputs.	With support: • I create programs that implement algorithms to achieve given goals. • I declare and assign variables. • I use post-tested loops e.g. 'until', and a sequence of selection statements in programs, including an if, then and else statement.	With support: • I know the difference between data and information. • I know why sorting data in a flat file can improve searching for information. • I use filters or can perform single criteria searches for information.	With support: • I know that computers collect data from various input devices, including sensors and application software. • I know the difference between hardware and application software, and their roles within a computer system.	With support: • I know the difference between the internet and internet service e.g. World Wide Web. • I show an awareness of, and can use a range of internet services e.g. VOIP. • I know what is acceptable and unacceptable behaviour when using technologies and online services.	With support: • I collect, organise and present data and information in digital content. • I create digital content to achieve a given goal through combining software packages and internet services to communicate with a wider audience e.g. blogging. • I make appropriate improvements to solutions based on feedback received, and can comment on the success the solution.
Developing	With prompting: • I show an awareness of tasks best completed by humans or computers. • I design solutions by decomposing a problem and create a sub-solution for each of these parts (decomposition). • I know that different solutions exist for the same problem.	With prompting: • I know the difference between, and appropriately I can use if and if, then and else statements. • I use a variable and relational operators within a loop to govern termination. • I design, write and debug modular programs using procedures. • I know that a procedure can be used to hide the detail with sub-solution (procedural abstraction).	With prompting: • I perform more complex searches for information e.g. using Boolean and relational operators. • I analyse and evaluate data and information, and I know that poor quality data leads to unreliable results, and inaccurate conclusions.	With prompting: • I know why and when computers are used. • I know the main functions of the operating system. • I know the difference between physical, wireless and mobile networks.	With prompting: • I know how to effectively use search engines, and I know how search results are selected, including that search engines use 'web crawler programs'. • I select, combine and I use internet services. • I show responsible use of technologies and online services, and I know a range of ways to report concerns.	With prompting: • I make judgements about digital content when evaluating and repurposing it for a given audience. • I know the audience when I am designing and creating digital content. • I know the potential of information technology for collaboration when computers are networked. • I use criteria to evaluate the quality of solutions and can identify improvements making some refinements to the solution, and future solutions.
Secure	• I know that iteration is the repetition of a process such as a loop. • I know that different algorithms exist for the same problem. • I represent solutions using a structured notation. • I identify similarities and differences in situations and can use these to solve problems (pattern recognition).	• I know that programming bridges the gap between algorithmic solutions and computers. • I have practical experience of a high-level textual language, including using standard libraries when programming. • I use a range of operators and expressions e.g. Boolean, and apply them in the context of program control. • I select appropriate data types.	• I know that digital computers use binary to represent all data. • I know how bit patterns represent numbers and images. • I know that computers transfer data in binary. • I know the relationship between binary and file size (uncompressed). • I define data types; real numbers and Boolean. • I query data on one table using a typical query language.	• I know the function of the main internal parts of basic computer architecture. • I know the concepts behind the fetch-execute cycle. • I know that there is a range of operating systems and application software for the same hardware.	• I know how search engines rank search results. • I know how to construct static web pages using HTML and CSS. • I know about data transmission between digital computers over networks, including the Internet i.e. IP addresses and packet switching.	• I evaluate the appropriateness of digital devices, internet services and application software to achieve given goals. • I recognise ethical issues surrounding the application of information technology beyond school. • I design criteria to critically evaluate the quality of solutions, I can use the criteria to identify improvements and make appropriate refinements to the solution.
Confident	• I know a recursive solution to a problem repeatedly applies the same solution to smaller instances of the problem. • I know that some problems share the same characteristics and use the same algorithm to solve both (generalisation). • I know the notion of performance for algorithms and I know that some algorithms have different performance characteristics for the same task.	• I use nested selection statements. • I know the need for, and can write, custom functions including use of parameters. • I know the difference between procedures and functions and I use them appropriately. • I use negation with operators. • I use and manipulate one dimensional data structures. • I can find and corrects syntactical errors.	• I know how numbers, images, sounds and character sets use the same bit patterns. • I perform simple operations using bit patterns e.g. binary addition. • I know the relationship between resolution and colour depth, including the effect on file size. • I distinguish between data used in a simple program (a variables) and the storage structure for that data.	• I know the von Neumann architecture in relation to the fetch-execute cycle, including how data is stored in memory. • I know the basic function and operation of location addressable memory.	• I know names of hardware e.g. hubs, routers, switches, and the names of protocols e.g. SMTP, IMAP, POP, FTP, TCP/IP, associated with networking systems. • I use technologies and online services securely, and I know how to identify and report inappropriate conduct.	• I justify the choice of and independently combine and use multiple digital devices, internet services and application software to achieve given goals. • I evaluate the trustworthiness of digital content and consider the usability of visual design features when designing and creating digital artefacts for known audience. • I design criteria for users to evaluate the quality of solutions, and use the feedback from users to identify improvements and make appropriate refinements to the solution. • I identify and explain how the use of technology can impact on society.
Exceptional	• I know that the design of an algorithm is distinct from its expression in a programming. • I evaluate the effectiveness of algorithms and models for similar problems. • I know where information can be filtered out in generalising problem solutions (abstraction). • I use logical reasoning to explain how an algorithm works. • I represent algorithms using a structured language.	• I know the effect of the scope of a variable e.g. a local variable can't be accessed from outside its function. • I know and apply parameter passing. • I know the difference between, and I can use, both pre-tested e.g. 'while', and post-tested e.g. 'until' loops. • I apply a modular approach to error detection and correction.	• I know the relationship between data representation and data quality. • I know the relationship between binary and electrical circuits, including Boolean logic. • I know how and why values are data typed in many different languages when manipulated within programs.	• I know that processors have instruction sets and that these relate to low-level instructions carried out by a computer.	• I know the purpose of the hardware and protocols associated with networking computer systems. • I know the client-server model including how dynamic web pages use server-side scripting and that web servers process and store data entered by users. • I know that persistence of data on the internet requires careful protection of online identity and privacy.	• I undertake creative projects that collect, analyse, and evaluate data to meet the needs of a known user group. • I effectively design and create digital artefacts for a wider or remote audience. • I consider the properties of media when importing them into digital artefacts. • I document user feedback, the improvements identified and the refinements made to the solution. • I explain and justify how the use of technology impacts on society, from the perspective of social, economical, political legal, ethical and moral issues.
Beyond	• I design a solution to a problem that depends on solutions to smaller instances of the same problem (recursion). • I know that some problems cannot be solved computationally.	• I design and write nested modular programs that enforce reusability utilising sub-routines wherever possible. • I know the difference between 'While' loop and 'For' loop, which I can use a loop counter. • I use two dimensional data structures.	• I perform operations using bit patterns e.g. conversion between binary and hexadecimal, binary subtraction etc. • I know and can explain the need for data compression, and performs simple compression methods. • I know what a relational database is, and I know the benefits of storing data in multiple tables.	• I have practical experience of a small (hypothetical) low level programming language. • I know and can explain Moore's Law. • I know and can explain multitasking by computers.	• I know the hardware associated with networking computer systems, including WANs and LANs, I know their purpose and how they work, including MAC addresses.	• I know the ethical issues surrounding the application of information technology, an existance of legal frameworks governing its use e.g. Data Protection Act, Computer Misuse Copyright etc.

KS3

Computer science at KS3 covers: computational abstraction, key algorithms, block and text based programming, Boolean logic, hardware, software, binary representation, creativity within ICT and Internet safety.

Ormiston Victory Academy

Key Stage 3

In Key Stage 3 we follow the National Curriculum for Computing. During Year 7, 8 and 9, students will develop skills relating to both Computer Science and ICT, to prepare them for when they make an informed choice in Key Stage 4.

In Year 7, students will study E-Safety through using Internet Searching, PowerPoint, Word and Desktop Publishing in autumn term. In spring term, students will create website using HTML. In Summer, students will study Python Turtle Programming.

In Year 8, students will study Spreadsheets and Databases in Autumn Term. From Spring, students will move on to more Computer Science based topics including algorithms, flowcharts and pseudocode. This will prepare them for summer term, where they will learn Basic Python Programming.

Sewell Park Academy

Computing KS3

Computing wants to develop student's computer skills across a range of software that can be used in all jobs when they leave school. We want students who will explore technology and its enjoyment and impact on society and business. Through a range of programing E-Safety and software units, we will create an environment to explore, make mistakes and learn new skills.

Hellesdon High School

ICT - Education and experience

We learn about E-Safety, Graphics, Programming, Website Design, Computational Thinking, Database Design, Animation, Spreadsheets, Multimedia Presentation and much more!

By the time they leave school, students will have a good foundation of skills that will allow them to make informed choices about the areas they would like to specialize in.

As well as practical skills, students will have developed important skills for the workplace, such as planning and organization of work, problem-solving and presentation, identifying users' needs, producing solutions and evaluating what has been developed.

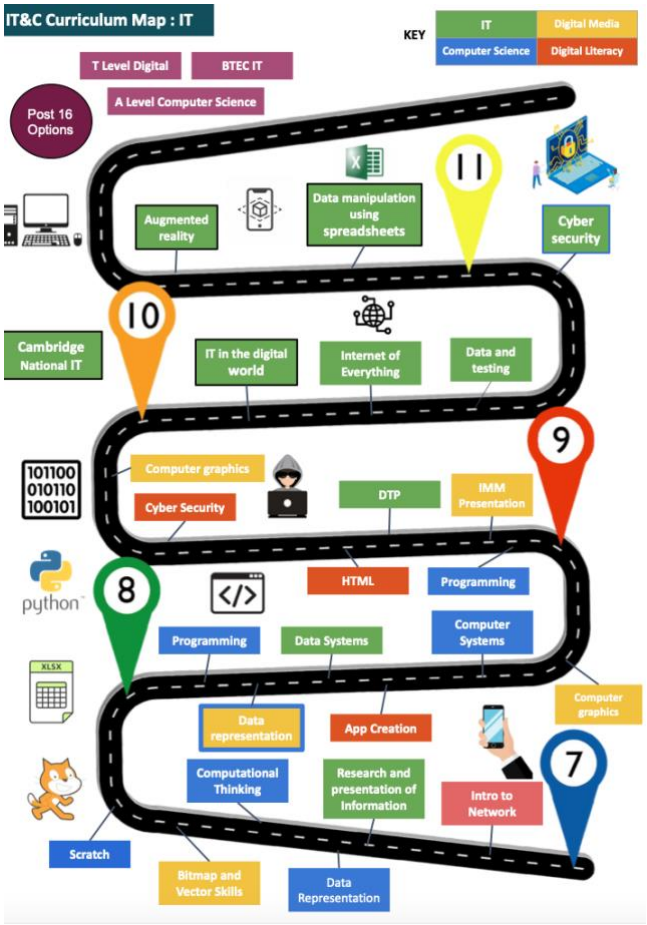
Notre Dame High School

	Year 7	Year 8	Year 9
Aut.1	PC basics (unplugged)	E-safety: Website design – Hot Topics	E-Safety: Text based e-safety game coded in Python
Aut.2	E-safety: quiz in Scratch	E-safety: Website design – Hot Topics	E-Safety: Text based e-safety game coded in Python
Spr.1	Scratch: Heli-cave and Rock, Paper, Scissors	Intro to Python	Cyber Security
Spr.2	Creating your own game in Scratch	Programming with Kodu – The Maze and Side Scroller	Dynamic website creation using HTML5, JavaScript and CSS
Sum.1	Programming in Python	Creating your own game in Kodu	Dynamic website creation using HTML5, JavaScript and CSS
Sum.2	Programming with the Micro:bit	Python Challenges	Set 1 –coding in C# Unity. Set 2 - Kodu

Thorpe St Andrew School

Curriculum Intent Statement:

We aim to develop students who are problem solvers and computational thinkers, who are also aware of the constantly evolving technological landscape. To develop students skills, knowledge and understanding throughout the three strands of the curriculum; Digital Literacy, IT and Computer Science. We will prepare them for their options at Key Stage 4 and 5, and their digital lives beyond.



Norwich School

Our computer scientists will learn to value computational thinking, helping pupils to develop the skills necessary to solve problems and to design systems that do so.

These skills will represent the best possible preparation for pupils who wish to study at a higher level or pursue a career in the Computer Science and IT industry. These qualification will also provide a good grounding for other subjects that require logical thinking, creative and analytical skills, particularly Mathematics.