

Limehurst Academy Computing Curriculum

Technology is everywhere and will play a pivotal part in students' lives. Therefore, at Limehurst Academy we want to model and educate our students on how to use technology positively, responsibly and safely. We have a broad curriculum that encompasses computing, information technology and digital literacy. We want our students to understand that there is always a choice with using technology and as a school we utilise technology to model positive use.

Knowledge and understanding

We recognise that technology can allow students to share their learning in creative ways. We also understand the accessibility opportunities technology can provide for our students. Our knowledge rich curriculum must be balanced with the opportunity for students to apply their knowledge creatively which will in turn help our students become skilful computer scientists.

We want our students to be fluent with a range of tools to best express their understanding and hope by the end of Key stage three, students have the independence and confidence to choose the best tool to fulfil the tasks and challenge set by their teachers.

Skills

Due to the broad nature of the Computing curriculum students will gain practical ICT skills which are transferrable to other subject areas and the world of work. Students will be confident in using a range of different software packages to solve problems.







Many of the ICT and Computing units require students to recall and build upon their mathematical skills. In Computing particularly students will need to have a logical mindset, where they will be required to design and program algorithms.

Literacy plays a key role in the Computing curriculum. Students will develop their evaluative skills as this is a key element of the 'Project Life Cycle', that needs to be completed so that reflection on the end solution can take place.

The Future





















The subject-specific knowledge developed in our Computing lessons equip students with experiences which will benefit them in further education and future workplaces. From research methods, use of presentation and creative tools and critical thinking, Computing at Limehurst Academy gives students the building blocks that enable them to pursue a wide range of interests and vocations in the next stage of their lives.







KS3 Computing Road Map – Links to the Curriculum














					Links to the National Curriculum						Link to GCSE Computer Science?	Link to OCR CAM NAT IT
					Algorithms	Programming & Development	Data & Data Representation	Hardware & Processing	Communication & Networks	Information Technology		
Unit	Unit Names	Prior topic requirement	# of lessons	Assessment Strands								
												
Year 7	1	Using Computers Safely	N/A	8					✓	✓	✓	✓
	2	Spreadsheet Modelling	N/A	8		✓				✓		✓
	3	Python Programming in Edublocks	N/A	6	✓	✓					✓	
	4	The Internet and Networks	N/A	7				✓	✓		✓	
	5	HTML and Web Development	N/A	4	✓	✓				✓		
	6	Flowcharts with Flowol	N/A	6	✓						✓	✓
Year 8	1	Spreadsheet Modelling	Year 7 Unit 1	8		✓				✓		✓
	2	Understanding Computers	N/A	8			✓	✓			✓	✓
	3	Cyber Security	Year 7 Unit 1	6			✓		✓	✓	✓	✓
	4	Computational thinking and logic	N/A	8	✓	✓	✓				✓	
	5	Introduction to Small Basic	Year 7 Unit 3	6	✓	✓					✓	
	6	Introduction to Python	Year 7 Unit 3	4	✓	✓					✓	
Year 9	1	Spreadsheet Modelling	Year 8 Unit 1	8		✓				✓		✓
	2	Data Representation (Images and Sound)	Year 8 Unit 4	8			✓	✓		✓	✓	



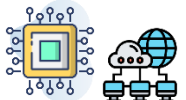





3	Modelling in Small Basic	Year 8 Unit 5	6	✓	✓					✓	
4	Further programming in Python	Year 8 Unit 6	8	✓	✓					✓	
5	AI and Machine Language	N/A	7	✓					✓	✓	✓
















Computing Road Map – All Year Groups



	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
Year 7	Using Computers Effectively, Safely and Responsibly 	Spreadsheets Harry Potter 	Programming in Python using block-based programming (Edublocks) 	The Internet and Networks 	HTML and Web Development 	Control Systems with Flowol 
Year 8	Spreadsheets Chimp Andy 	Understanding Computers 	Cyber Security 	Computational thinking and Logic 	First Steps in Small Basic 	Introduction to Python 
Year 9	Spreadsheet Modelling 	Data representation (Images and Sound) 	Modelling in Small Basic 	Python Next Steps 	AI and Machine Learning 	
	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
Year 10 Cam Nat	R050: TA3 - Data & Testing (3.1 Information & data, 3.2 Data use) 	R050: TA2 Human Computer Interface in Everyday Life 	Practical Spreadsheet Skills (planning, creating, testing, and evaluating) Practice Project	R060 Spreadsheet NEA 	R060: TA4 - Evaluating the spreadsheet solution	R070: TA1 - Augmented Reality (AR) (1.1 Purpose and uses of AR, 1.2 Types of AR and user interaction, 1.3 Devices used with AR)

	<p>R050: TA1 - Design tools (1.1 Types of design tools)</p> 	<p>R060: TA2 Creating the spreadsheet solution (2.1.1 Data handling & manipulation, 2.1.2 Techniques to generate the outputs, 2.1.3 User interface)</p> 	<p>R060: TA1.2 HCI design conventions and principles (1.2.1 Functionality, 1.2.2 Types of</p> <p>R060: TA2 Creating the spreadsheet solution (2.1.1 Data handling & manipulation, 2.1.2 Techniques to generate the outputs, 2.1.3 User interface)</p> <p>R050: Recap TA3 – 3.5.2 Testing</p> <p>R060: TA3 - Testing the spreadsheet solution (3.1 Test the user interface and technical aspects of the spreadsheet solution)</p> 		<p>R060: Spreadsheet NEA – Finish and submit</p> 	<p>Blippar skill building</p> <p>R070: TA3 - Creating and AR model prototype (3.1 AR model prototype, 3.2 Triggers, 3.3 Layers/ user interaction, 3.4 Information output)</p> 
<p>Year 11 Cam Nat</p>	<p>Practice AR Project (planning, creating, testing, and evaluating) on XR Studio +</p> <p>R070: Recap - TA2 - Designing an AR model prototype (2.1 Planning</p>	<p>R050: Recap TA3 – 3.5.2 Testing</p> <p>R070: TA4 - Testing and reviewing (4.1 Testing, 4.2 Reviewing the process of creating the AR model prototype)</p>	<p>R050: TA5 - Digital communications (5.1 Types, 5.2 Software, 5.3 Digital devices, 5.4 Distribution channels, 5.5 Audience demographics)</p>	<p>R050: TA4 – Cyber security & legislation4.1. Threats. 4.2 impact of attacks, 4.3 Prevention measures, 4.4 Legislation related</p>	<p>R050 Exam Revision (TA1, TA2, TA3,TA4,TA5 and TA6)</p> 	

	<p>and design consideration, 2.2 Design tools)</p> <p>R070: Recap - TA3 - Creating and AR model prototype (3.1 AR model prototype, 3.2 Triggers, 3.3 Layers/ user interaction, 3.4 Information output)</p> <p>R070: Augmented Reality NEA</p> 	<p>R070: Augmented Reality NEA</p> 	<p>R050: TA6 - Internet of Everything (IoE) (6.1 Use of IoE, 6.2 Application areas in everyday life)</p> 	<p>to the use of IT systems)</p>  <p>R050 – Recap TA1, TA2, TA3 revision</p>		
	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
Year 10 Computing	<p>2.2 Programming Fundamentals</p>  <ul style="list-style-type: none"> • Input, Output and Structure • Variables and Constants • Data types, Casting & Concatenation • Procedures & Functions • Arithmetic • Relational/Comparison Operators & If statements <p>1.1 System Architecture</p>  <ul style="list-style-type: none"> • CPU Components 	<p>2.2 Programming Fundamentals</p>  <ul style="list-style-type: none"> • Else, Elif statements & Boolean operators • Match case statements • While loops • Looping forever • Break statements • Floor division, modulus & exponential <p>1.2 Primary & Secondary Storage</p>	<p>2.2 Programming Fundamentals</p>  <ul style="list-style-type: none"> • Importing modules & Random number generation • Random Seeds • String formatting • Finding the length of a list • List indexing & Choosing from a list <p>1.3 Computer networks, connections and protocols</p>	<p>2.2 Programming Fundamentals</p>  <ul style="list-style-type: none"> • Storing and reversing lists • Shuffling lists • Modifying & copying lists • Checking if something is in a list <p>1.4 Network Security</p>  <ul style="list-style-type: none"> • Dos, Brute Force & SQL Injection • Human errors 	<p>2.2 Programming Fundamentals</p>  <ul style="list-style-type: none"> • For loops • Nested for loops • Time module • String classification <p>1.5 System Software</p>  <ul style="list-style-type: none"> • Operating systems • Utility software <p>1.6 Ethical, Cultural, Legal and Environmental</p>	<p>2.2 Programming fundamentals</p>  <ul style="list-style-type: none"> • String indexing and slicing • String splitting and joining • Making replacements & stripping strings • String multiplication & working with Unicode • Exception handling • File handling <p>1.6 Ethical, cultural, legal and</p>







	<ul style="list-style-type: none"> • Von Neumann Architecture • FDE Cycle • CPU Performance • Embedded Systems <p>1.2 Primary & Secondary Storage</p>  <ul style="list-style-type: none"> • Primary Storage • Virtual Memory • The need for and types of secondary storage 	 <ul style="list-style-type: none"> • Secondary storage characteristics • Units and converting between them • Data rep - Numbers (binary conversion, binary addition, shifts & Hex) • Data rep - Images • Data rep - Characters • Data rep - Sound • Compression <p>1.3 Computer networks, connections and protocols</p>  <ul style="list-style-type: none"> • Types of networks (LAN & WAN) • Network Performance 	 <ul style="list-style-type: none"> • P2P & Client server model • LAN hardware • The internet • Star & Mesh networks • Modes of connection • Wireless encryption • IP & MAC Addressing • Cloud Computing • Standards & Protocols • Layers 	<ul style="list-style-type: none"> • Malware & Social Engineering • Identifying & preventing vulnerabilities 	<p>Issues in Computing</p>  <ul style="list-style-type: none"> • Investigating technology • Privacy issues • Cultural issues 	<p>environmental issues in computing</p>  <ul style="list-style-type: none"> • Environmental issues • Impacts of digital technology on wider society • Legislation • Open source vs proprietary software
Year 11 Computing	<p>1.1 System Architecture Revision</p> 	<p>2.2 Programming fundamentals</p>  <ul style="list-style-type: none"> • SQL queries 	<p>1.2 Primary & Secondary storage</p>	<p>1.3 Computer networks, connections and protocols</p>	Smart revise revision (tailored to students' strengths & weaknesses)	

<ul style="list-style-type: none"> • CPU Components • Von Neumann Architecture • FDE Cycle • CPU Performance • Embedded Systems <p>1.2 Primary & Secondary Storage revision</p>  <p>2.1 Algorithms</p> <ul style="list-style-type: none"> • Abstraction • Decomposition • Algorithmic Thinking • Flowcharts  <p>2.2 programming techniques</p>  <ul style="list-style-type: none"> • Variables & Constants • Data Types • Data Structures <p>2.3 Producing robust programs</p>  <ul style="list-style-type: none"> • Authentication • Validation • Anticipating Misuse • Maintainability 	<p>2.4 Boolean Logic</p>  <p>2.1 Algorithms</p>  <ul style="list-style-type: none"> • Linear search • Binary search • Bubble sort • Merge sort • Insertion sort <p>2.3 Producing robust programs</p>  <ul style="list-style-type: none"> • Testing & Test data <p>2.5 Programming languages and integrated development environments</p>   <ul style="list-style-type: none"> • Low level languages • High level languages • Interpreters • Compilers • Integrated development environment features 	  <ul style="list-style-type: none"> • Data rep (binary, hex, shifts & addition) • Data rep (images, sounds and compression) <p>1.3 Computer networks, connections and protocols</p>  <ul style="list-style-type: none"> • Cloud computing <p>2.1 Algorithms</p>  <ul style="list-style-type: none"> • Flow charts • Algorithmic thinking <p>2.2 Programming fundamentals</p> 	 <ul style="list-style-type: none"> • Types of networks (LAN & WAN) • P2P & Client server model • Topologies • The internet • Modes of connection & wireless encryption • IP & MAC addressing <p>2.1 Algorithms revision</p> <p>2.2 Programming fundamentals revision</p> <p>2.3 Producing robust programs revision</p> <p>2.4 Boolean logic revision</p> <p>2.5 Programming languages & IDE revision</p>			
--	--	---	---	--	--	--

		<p>1.6 Ethical, Legal, Cultural & Environmental Issues</p>  <ul style="list-style-type: none">• Exam strategy & QWC <p>1.2 Primary & Secondary storage</p>  <ul style="list-style-type: none">• Converting between units				
--	--	--	--	--	--	--

Year 7

Unit 1: Using Computers Effectively and Safely and Responsibly

Links to the National Curriculum					
Algorithms 	Programming & Development 	Data & Data representation 	Hardware & Processing 	Communication & Networks 	Information Technology 

This is a theoretical unit covering the necessary basic knowledge to use computers safely, effectively, and responsibly. Students begin by looking at file management and security. The unit then moves on to e-safety (cyber-bullying, phishing etc.), and online profiles to give students a better understanding and awareness of using social media.

Intent

Knowledge acquired:

1. Use file management techniques to create folders, save, copy, move, rename and delete files and folders and make backup copies of files
2. Recognise extensions for common file types such as .doc or .docx, .ppt, .jpg etc
3. Keep their files in well organised and appropriately named folders
4. Explain what constitutes a "strong" password for an online account
5. Describe a code of conduct
6. List the dangers and drawbacks of social networking sites
7. List responses to cyberbullying

Skills developed:

- File management skills will be developed that the students will hopefully use whenever documents need to be saved onto the computer.
- Skills and an understanding of staying safe when online will be developed.

Understanding:

- Understanding how to respond to cyber bullying.
- Understanding how to use advanced features of a search engine.
- Understanding why the information they find may not be accurate.

Links to National Curriculum KS2:

- use search technologies effectively, appreciate how results are selected and ranked, and be discerning in evaluating digital content
- use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact.

Links to future study:

Year 8: Unit 2 - Understanding Computers
Unit 3 - Cyber Security

KS4: GCSE OCR Computer Science:
Unit 1.4 – Network Security
Unit 1.6 – Ethical, legal, cultural and environmental impacts in computing

OCR Cambridge Nationals:

LO4 – Understanding factors to be considered when collecting and processing data/information

Implementation







Number of lessons: ≈ 6

Description of lesson content		Written Feedback	
1	First Day routines (passwords/folders)		
2	Routines and File management		
3	Social networking	✓	Checking students' justifications in their work
4	Keeping your data safe		
5	Being smart with your smart phone	✓	Checking key terminology and justifications
6	Revision Lesson & Assessment		

<u>Main Resources</u>	<u>Method of Assessment</u>	<u>Use of Assessment Results</u>
<ul style="list-style-type: none"> PG Online Worksheets Teacher prepared PowerPoint & Activities 	<ul style="list-style-type: none"> Recall Quizzes Self-Assessed and Peer assessed Written feedback (L3 & L5) End of Unit Assessment 	<ul style="list-style-type: none"> Used to identify areas of weakness Used to set WWW and EBI based on Assessment

Year 7

Unit 2: Spreadsheet Modelling

Links to the National Curriculum					
Algorithms 	Programming & Development 	Data & Data representation 	Hardware & Processing 	Communication & Networks 	Information Technology 

This is a practical, skills-based unit covering the principles of creating and formatting basic spreadsheets to produce and use simple computer models. Students will learn how to use basic Formula and Functions, use different formatting features, and create graphs and charts with data. They will also learn to use and link multiple tabs on a spreadsheet.

Intent

Knowledge acquired:

1. Use the / * - + to create simple formulas.
2. Add formatting to text and cells.
3. Work independently to create professional spreadsheets that are formatted to the needs to the required audience.
4. Create a chart using data from a spreadsheet, with appropriate chart titles and axis labels.
5. Use AutoFill to replicate a formula.
6. Use the Sum, MAX, MIN and AVERAGE Functions appropriately.
7. Use the Sort and Filter tool to sort the data in ascending order.
8. Format the text to appropriately use text wrap, Merge Cells and resize cells .
9. I know when it is appropriate to format the cells to currency.
10. I can add an image to my spreadsheet.
11. Understand that Some Charts will need Data labels (E.G Percentages). And Colours on a chart can be changed.

Skills developed:

- Be able to use MS Excel to create spreadsheet models to solve real life problems.
- Use spreadsheet features effectively to create spreadsheet models.

Understanding:

- Understand the uses of a spreadsheet, and where in industry they are useful.

Links to National Curriculum KS2:

- 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.

Links to future study:

Year 8: Unit 1 - Spreadsheet Modelling – Chimp Andy

Year 9: Unit 1 - Further spreadsheet modelling – TBNT

KS4: OCR Cambridge Nationals:
L05- to be able to import and manipulate data to develop a solution to meet an identified need.

Implementation







Number of lessons: ≈ 7

	Description of lesson content		Written Feedback
1	Shopping list - Using basic formulas and formatting	✓	<ul style="list-style-type: none"> • Checking students can add data into a spreadsheet and format their data. • Checking students can use formulas and functions (multiplying and sum)
2	Hogwalk's houses – basic functions, entering data and formatting features	✓	<ul style="list-style-type: none"> • Checking students formatting skills • Checking students can use functions: Sum, Max, Min, Average • Checking students can create a bar chart
3	Test scores – formatting cells and using functions		
4	Spell costs – working on multiple sheets, sorting data and setting cells to currency		
5	Stationary shop – Planning and creating a spreadsheet		
6	Delivering letters – Planning and creating a spreadsheet		
7	Revision and Assessment		

Main Resources	Method of Assessment	Use of Assessment Results
<ul style="list-style-type: none"> • Teacher prepared Worksheets. • Teacher prepared PowerPoint. • Teacher prepared example Spreadsheets for modelling purposes 	<ul style="list-style-type: none"> • Self-Assessed and Peer assessed. • Written Feedback (L1 & L2) • End of Unit Assessment 	<ul style="list-style-type: none"> • Used to identify areas of weakness. • Used to set WWW and EBI based on Assessment

Year 7

Unit Three: Edublocks

Links to the National Curriculum					
Algorithms 	Programming & Development 	Data & Data representation 	Hardware & Processing 	Communication & Networks 	Information Technology 

This unit begins by securing students understanding of key terms: algorithms and sequence. Students are introduced to the Edublocks platform which is a block-based programming tool used to support students transition between Scratch (primarily used at primary school) and the Python programming language. Further programming fundamentals are then explored such as variables, selection (if statements), operators and count controlled iteration.

Intent

Knowledge acquired:

1. Know the term 'algorithm'
2. Know the term 'sequence'
3. Know how to create a python program using Edublocks
4. How to create algorithms
5. Know the term 'variable'
6. Trace the values of variables within a sequence
7. How to create a sequence using variables
8. Know what is meant by the term 'selection'
9. Create a program that will use selection
10. Know the operators used in Python
11. Creating conditions that use comparison and logical operators
12. Know the term iteration
13. Explain why we need iteration in code
14. Implement count-controlled iteration in a program

Skills developed:

- Computational thinking, algorithms and programming

Understanding:

- The unit will establish a foundation understanding of the main programming fundamentals sequence, selection & iteration
- The unit will establish an understanding of variables and how they are used to refer to data within our programs

Links to National Curriculum KS2:

- design, write and debug programs that accomplish specific goals
- 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

Links to future study:

- Year 8: Unit 4 – Computational thinking and logic
Unit 5 – Introduction to programming in Small basic
Unit 6 – Introduction to programming in Python
- Year 9: Unit 5 – Further programming in small basic
Unit 6 – Further programming in Python
- KS4: OCR Computer Science:
Unit 2.1 Algorithms
Unit 2.2 Programming fundamentals
Unit 2.3 Programming robust programs

Implementation







Number of lessons: ≈ 6

Description of lesson content		Written Feedback	
1	Algorithms & Sequences		
2	Variables	✓	<ul style="list-style-type: none"> • Checking that students understand the term 'variable' • Checking students can create and use variables within their code
3	Selection	✓	<ul style="list-style-type: none"> • Checking that students understand the term 'selection' • Checking students can use if statements within their code
4	Operators		
5	Count-controlled Iteration		
6	Revision and Assessment		

<u>Main Resources</u>	<u>Method of Assessment</u>	<u>Use of Assessment Results</u>
<ul style="list-style-type: none"> • Worksheets • PowerPoints • EduBlocks 	<ul style="list-style-type: none"> • Self-Assessed and Peer assessed. • Written Feedback (L2 & L3) • End of Unit Assessment 	<ul style="list-style-type: none"> • Used to identify areas of weakness. • Used to set WWW and EBI based on Assessment

Year 7

Unit 4: The Internet and Networks

Links to the National Curriculum					
Algorithms 	Programming & Development 	Data & Data representation 	Hardware & Processing 	Communication & Networks 	Information Technology 

This unit begins by defining a network and addressing the benefits of networking, before covering how data is transmitted across networks using protocols. The types of hardware required are explained, as is wired and wireless data transmission. Learners will develop an understanding of the term's 'internet' and 'World Wide Web', and of the key services and protocols used. Practical exercises are included throughout to help strengthen understanding.

Intent

Knowledge acquired:

1. Define what a computer network is and explain how data is transmitted between computers across networks.
2. Define 'protocol' and provide examples of non-networking protocols.
3. List examples of the hardware necessary for connecting devices to networks
4. Compare wired to wireless connections and list examples of specific technologies currently used to implement such connections.
5. Define 'bandwidth', using the appropriate units for measuring the rate at which data is transmitted.
6. Define what the internet is
7. Explain how data travels between computers across the internet.
8. Describe key words such as 'protocols', 'packets', and 'addressing'.
9. Explain the difference between the internet, its services, and the World Wide Web.
10. Describe how services are provided over the internet.
11. List some of these services and the context in which they are used.
12. Explain the term 'connectivity' as the capacity for connected devices ('Internet of Things') to collect and share information about me with or without my knowledge (including microphones, cameras, and geolocation)
13. Describe how internet-connected devices can affect me.
14. Describe components (servers, browsers, pages, HTTP and HTTPS protocols, etc.) and how they work together.

Skills developed:

- This unit progresses students' knowledge and understanding of networks and associated hardware.

Understanding:

- The unit will establish a foundation understanding of how data is transmitted across networks, as well as exploring the factors that can affect performance.
- The unit will spend time focussing on the internet and services provided over the internet.

Links to National Curriculum KS2:

- Understand the hardware and software components that make up computer systems, and how they communicate with one another and with other systems.

Links to prior study:

Year 7: Unit 1 – Using computers effectively and safely

Links to future study:

Year 7: Unit 5 – HTML Coding

Year 8: Unit 3 – Cyber Security

KS4: OCR Computer Science:
Unit 3 - Computer networks, connections, and protocols
OCR Cambridge Nationals:
TA6 Internet of Everything (IoE)

Implementation







Number of lessons: ≈ 7

	Description of lesson content	Written Feedback	
1	Computer networks and protocols		
2	Networking hardware		
3	Wired and wireless networks		
4	The internet		
5	Internet services		
6	The world wide web		
7	Revision and Assessment		

<u>Main Resources</u>	<u>Method of Assessment</u>	<u>Use of Assessment Results</u>
<ul style="list-style-type: none"> • NCCE Worksheets • NCCE PowerPoint 	<ul style="list-style-type: none"> • Self-Assessed and Peer assessed. • End of Unit Assessment 	<ul style="list-style-type: none"> • Used to identify areas of weakness. • Used to set WWW and EBI based on Assessment

Year 7

Unit 5: HTML and Web Development

Links to the National Curriculum					
Algorithms 	Programming & Development 	Data & Data representation 	Hardware & Processing 	Communication & Networks 	Information Technology 

Students will learn the basics of HTML and CSS. They will learn how to create text styles and add content, including text and graphics, in a specified position on a page, as well as navigation links to other pages on their website and to external websites. The basics of good design are covered, and with the help of worksheets, students will develop their own templates in a text editor such as Notepad. They will decide on a Unit for their websites, document their designs and collect suitable text and images. They will then use their HTML templates to create their websites, including a web form.

Intent

Knowledge acquired:

1. Write HTML code to create a simple web page and display it in a browser.
2. Write CSS to define the styles used in a web page.
3. Create a simple navigation system using HTML.
4. Use a design to create a template for a web page using HTML.
5. Insert text, images and links on their web pages.
6. Use a range of HTML tags to create well laid out web pages.
7. Use the template to design a multi-page website with a consistent look and feel to each page.
8. Create a simple web form to collect user data.

Skills developed:

- Students will learn the skills to code websites in HTML and Serif Web Plus.

Understanding:

- Understand the basic HTML Tags that can be used to create a webpage.
- Understand that Small Basic –Web Plus is a Web authoring software.
- Understand how to change HTML code for a website that they have created in Web Plus.

Links to National Curriculum KS2:

- 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.

Links to prior study:

Year 7: Unit 1 – Using computers effectively and safely
Unit 4 – The internet and networks

Links to future study:

Year 8: Unit 3 – Cyber security

KS4: OCR Cambridge Nationals:
LO7 to be able to select and present information in the development of the solution to meet an identified need.

Implementation







Number of lessons: ≈ 5

	Description of lesson content	Written Feedback
1	Happy birthday e-card	
2	Tell a story comic	
3	Wanted Poster	
4	Recipe	
5	Revision and Assessment	

<u>Main Resources</u>	<u>Method of Assessment</u>	<u>Use of Assessment Results</u>
<ul style="list-style-type: none"> PG Online Worksheets Teacher prepared PowerPoint. Teacher prepared example Spreadsheets for modelling purposes 	<ul style="list-style-type: none"> Self-Assessed and Peer assessed. End of Unit Assessment 	<ul style="list-style-type: none"> Used to identify areas of weakness. Used to set WWW and EBI based on Assessment

Year 7

Unit 6: Flowcharts with Flowol

Links to the National Curriculum					
Algorithms 	Programming & Development 	Data & Data representation 	Hardware & Processing 	Communication & Networks 	Information Technology 

This is a practical unit covering the principles of producing control and monitoring solutions using a flowchart-based interface (Flowol). Students will start by producing systems that use simple loops and basic outputs and then move on to look at systems that have multiple inputs and outputs. They will refine their solutions using subroutines and variables.

Intent

Knowledge acquired:

1. Identify everyday situations where computer control is used.
2. Identify common types of sensors used by control systems.
3. Identify control flowchart symbols and understand how they are used to break down problems.
4. Produce flowchart-based solutions for control systems that include sequences and loops.
5. Explain why control systems might fail and how this might impact on safety.
6. Produce control solutions for problems that include subroutines.
7. Produce control solutions for problems that include variables.

Skills developed:

- Use Flowol to create flowcharts, that can be linked to mimics to test.
- Build skills to create routines and subroutines with flowcharts.

Understanding:

- Understand the control flowchart symbols and how they are used to break down problems.
- Understand that flowcharts are used a planning tool by professional coders in industry

Links to National Curriculum KS2:

- 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

Links to prior study:

Year 7: Unit 3 – Python programming with Edublocks

Links to future study:

Year 8: Unit 4 – Computational thinking and logic

Unit 5 - First Steps in Small Basic

Unit 6 – Introduction to Python

Year 9: Unit 3 – Modelling in Small Basic

Unit 4 – Further programming in Python

KS4: GCSE OCR Computer Science:

Unit 2.1- Algorithms

Unit 2.2 – Programming fundamentals

Unit 2.3 – Creating robust programs

Implementation







Number of lessons: ≈ 6

	Description of lesson content	Written Feedback
1	Solving problems	
2	Sequencing	
3	Sensors	
4	Subroutines	
5	Actuators	
6	Revision and Assessment	

<u>Main Resources</u>	<u>Method of Assessment</u>	<u>Use of Assessment Results</u>
<ul style="list-style-type: none"> • PG Online Worksheets • Teaching PowerPoint • Flowol Mimics • Mimic Solutions • Software: Flowol 	<ul style="list-style-type: none"> • Self-Assessed and Peer assessed • End of Unit Assessment 	<ul style="list-style-type: none"> • Used to identify areas of weakness • Used to set WWW and EBI based on Assessment

Year 8

Unit 1: Spreadsheet Modelling

Links to the National Curriculum					
Algorithms 	Programming & Development 	Data & Data representation 	Hardware & Processing 	Communication & Networks 	Information Technology 

This is a practical Spreadsheet modelling unit where students will understand how Spreadsheets are used. Students will learn basic formula and functions to solve problems. They will learn to create professional looking sheets with formatting to meet the needs of the end user. They will learn to create charts and graphs from datasets, and work between multiple sheets.

Intent

Knowledge acquired:

1. Format a spreadsheet, by resizing column widths, merging cells and centre the title and wrap text labels.
2. Font style, size and colours need to be consistent and professional.
3. Images used are appropriate and of a good quality and formatted correctly.
4. Change data types appropriately
5. Appropriately use formula and functions to solve problems.
6. Use advanced features of the software to sort and filter data.
7. Able to amend IF statements to reflect changes of certain rules.
8. Make proper use of absolute cell referencing and conditional formatting.
9. Create meaningful charts with appropriate Data Labels
10. Use Goal Seek to solve 'What If' problems.

Skills developed:

- Build skills to be able to independently create professional spreadsheets with formulae and functions.
- Create meaningful charts and graphs with data within a spreadsheet.

Understanding:

- Understand when it is appropriate to use spreadsheet software, and the types of problems that can be solved.
- Understand where in industry Spreadsheet models are used.

Links to National Curriculum KS2:

- 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

Links to Prior study:

Year 7: Unit 2 - Spreadsheet Modelling – Harry Plotter

Links to future study:

Year 9: Unit 1 - Further spreadsheet modelling – TBNT

KS4: OCR Cambridge Nationals:

L05- to be able to import and manipulate data to develop a solution to meet an identified need.

Implementation

Number of lessons: ≈ 5







	Description of lesson content	Written Feedback
1	Chimp Andy's Resources (adding and formatting data)	
2	Wrestling Match (conditional formatting, basic functions and formatting)	<ul style="list-style-type: none"> • Checking to see if students can use conditional formatting • Checking to see if students can use count If

			<ul style="list-style-type: none"> • Checking to see if students are able to create charts
3	Operation Catnip (formatting text and data, creating charts)		
4	Mobile Phone Comparisons (formatting text and data, creating charts)		
5	Right hand man competition (formatting text and data, creating charts)		
6	Test scores	✓	<ul style="list-style-type: none"> • Checking to see if students can calculate percentages using formulas • Checking to see that students can use functions average, max and min • Checking to see if students can use VLOOKUP
7	Revision and Assessment		

<u>Main Resources</u>	<u>Method of Assessment</u>	<u>Use of Assessment Results</u>
Original resources from Teach-ICT Teacher prepared PowerPoint & worksheet	Self-Assessed and Peer assessed Written feedback (L2 & L6) End of Unit Assessment	Used to identify areas of weakness Used to set WWW and EBI based on Assessment

Year 8

Unit 2: Understanding Computers

Links to the National Curriculum					
Algorithms 	Programming & Development 	Data & Data representation 	Hardware & Processing 	Communication & Networks 	Information Technology 

This is a theoretical unit covering the basic principles of computer architecture and use of binary. Students will revise some of the theory on input and output covered in previous learning and continue to look at the Input-Process-Output sequence and the Fetch-Decode-Execute cycle through practical activities. Students will then look at some simple binary to decimal conversion and vice versa and learn how text characters are represented using the ASCII code. This will be followed by some simple binary addition. Students will learn more in depth how storage devices represent data using binary patterns and physically save these patterns. Finally, they will look at a brief history of communication devices, how new technologies and applications are emerging and the pace of change.

Intent

Knowledge acquired:

1. Distinguish between hardware and software
2. CPU, input, output and storage devices
3. Name different types of permanent storage device
4. Suggest appropriate input and output devices for a simple scenario
5. Explain what RAM and ROM are used for
6. Show how numbers and text can be represented in binary
7. Explain the impact of future technologies
8. Perform simple binary arithmetic
9. State strengths and weaknesses of different storage devices
10. Describe briefly how data is stored on a CD
11. Identify input and output devices for more complex scenarios
12. Explain how characters are encoded using the ASCII system
13. Use an ASCII reference chart to convert a character into binary and its decimal equivalent

Skills developed:

- Develop theoretical computing skills about the basic principles of computer architecture and use of binary.
- Develop an understanding of the Input-Process-Output sequence and the Fetch-Decode-Execute cycle.
- Develop skills to use binary numbers and how storage devices represent data using binary patterns and physically save these patterns.
- Understand the hardware and software components that make up computer systems, and how they communicate with one another and with other systems
- Understand how instructions are stored and executed within a computer system; understand how data of various types (including text, sounds, and pictures) can be represented and manipulated digitally, in the form of binary digits; be able to convert between binary and decimal and perform simple binary arithmetic.

Links to National Curriculum KS2:

- 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

Links to prior study:

Year 7: Unit 6 – Flowcharts with Flowol

Links to future study:

Year 8: Unit 4 – Computational thinking and logic

Year 9: Unit 2 – Data representation (images and sound)

KS4: GCSE OCR Computer Science:

Unit 1.1 Systems Architecture

Unit 1.2 Memory and Storage

Unit 1.6 Ethical, legal, cultural and environmental impacts of computing

Implementation







Number of lessons: ≈ 7

	Description of lesson content		Written Feedback
1	Elements of a computer	✓	<ul style="list-style-type: none"> Check that students can categorise hardware into input, output and storage Check students' justifications
2	The central processing unit		
3	Understanding Machine code	✓	<ul style="list-style-type: none"> Check that students are able to convert between base 10 (denary) and base 2 numbers (binary)
4	Binary mathematics		
5	Secondary storage devices		
6	Convergence and new technology		
7	Revision and assessment		

<u>Main Resources</u>	<u>Method of Assessment</u>	<u>Use of Assessment Results</u>
<ul style="list-style-type: none"> PG Online Worksheets PG Online PowerPoint 	<ul style="list-style-type: none"> Self-Assessed and Peer assessed Written Feedback (L1 & L3) End of Unit Assessment 	<ul style="list-style-type: none"> Used to identify areas of weakness Used to set WWW and EBI based on Assessment

Year 8

Unit 3: Cyber Security

Links to the National Curriculum					
Algorithms 	Programming & Development 	Data & Data representation 	Hardware & Processing 	Communication & Networks 	Information Technology 

This unit is a follow on from Year 7 where they studied 'Using computers effectively and safely'. Students will explore why their data is valuable to others and why it is important to keep it safe. In this unit, different types of cybercrimes are explained including social engineering and malware. Finally, students will explore different ICT Laws and how the laws protect us from hackers and help keep our data safe.

Intent

Knowledge acquired:

- To identify the common malware threats
- To identify what happens to data that is entered online
- To define hacking in the context of cyber security
- To explain the need for data protection act
- To explain the need for the computer misuse act
- To independently explain the difference between data and information

Skills developed:

- To identify strategies to reduce the chance of a brute force attack being successful
- To implement strategies to minimise the risk of data being compromised through human error
- To identify the most effective methods to prevent cyber attacks

Understanding:

- To explain how malicious bots can have an impact on societal issues
- To compare threats against their probability and their potential impact to organisations
- To examine how different types of malware cause problems for computer systems
- To explain how networks can be protected from common security threats
- To explain how DDOS attack works and how it impacts online services
- To critique online services in relation to data privacy
- To recognise how human errors pose security risks to data

Links to National Curriculum KS2:

- Understanding computer networks and the internet – Year 8 criteria explaining DDOS attacks, network protection, malware threats build on KS2 foundations of how networks function
- Use technology safely, respectfully, and responsibly – KS2 introduces concepts of online safety, data privacy, and ethical use. This unit expands this by critiquing online services for data privacy, explaining the data protection act and computer misuse act and recognising human error as a security risk.
- Identify acceptable/unacceptable behaviour and know how to report concerns – KS2 pupils learn about cyberbullying safe communication and reporting misuse. Year 8 builds on this with defining hacking in a cybersecurity context and explaining societal impacts of malicious bots
- Select, use and combine software on a range of digital devices – KS3 focuses on basic software use, year 8 introducing implementing strategies to reduce brute force attacks and minimising data compromise which requires applying tools and techniques.

Links to prior study:

Year 7: Unit 1 – Using computers effectively and safely
Unit 4 – The internet and networks

Links to future study:

Year 9: Unit 5 - AI and Machine Learning

KS4: GCSE OCR Computer Science:
 Unit 1.3 Computer networks
 Unit 1.4 Network Security
 Unit 1.6 Ethical, legal, cultural and environmental impacts of computers
 Unit 2.3 Creating robust programs

OCR Cambridge National in IT:
 TA4 Cyber Security and Legislation

Implementation







Number of lessons: ≈ 6

Description of lesson content		Written Feedback	
1	You and your data	✓	Checking student justifications based on recommended Items
2	Social engineering		
3	Script Kiddies	✓	Check student justifications for each scenario in relation to the Computer Misuse Act
4	Malware and Viruses		
5	Cyber Security Prevention		
6	Revision and Assessment		

<u>Main Resources</u>	<u>Method of Assessment</u>	<u>Use of Assessment Results</u>
<ul style="list-style-type: none"> • NCCE worksheets • NCCE PowerPoints which have been adapted by the teacher • 	<ul style="list-style-type: none"> • Self-Assessed and Peer assessed • Written feedback • End of Unit Assessment 	<ul style="list-style-type: none"> • Used to identify areas of weakness • Used to set WWW and EBI based on Assessment

Year 8

Unit 4: Computational thinking and Logic

Links to the National Curriculum					
Algorithms 	Programming & Development 	Data & Data representation 	Hardware & Processing 	Communication & Networks 	Information Technology 

This unit introduces to students to key problem-solving techniques used by computer scientists, including abstraction, decomposition, and logical reasoning. It uses unplugged activities to make these concepts accessible and engaging.

Intent

Knowledge acquired:

- Boolean logic: Understanding AND, OR and NOT gates and their symbols
- Truth tables: How logical operations produce outputs
- Computational thinking vocabulary: Terms like abstraction decomposition, algorithm and logic
- Problem types: Recognising different computational problems and they're approached
- Basic flow chart symbols: Used to represent algorithms visually

Skills developed:

- Creating and interpreting truth tables for logic gates
- Designing algorithms using flowcharts or pseudocode
- Breaking down problems into smaller steps (decomposition)
- Identifying patterns and removing unnecessary detail (abstraction)
- Applying logical reasoning to solve puzzles and predict outcomes
- Evaluating algorithms for efficiency and clarity

Understanding:

- Why abstraction is useful: Understanding how to be simplifying problems helps focus on what matters
- How logic gates underpin computing: realising that digital decisions are based on binary logic
- How algorithms solve real-world problems: connecting classroom tasks to real applications
- The importance of structure in problem-solving: recognising how decomposition and logic make complex tasks manageable
- How computational thinking applies beyond computing: seeing its relevance in maths, science and everyday decision making

Links to National Curriculum KS2:

- 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 logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs.

Links to prior study:

Year 7: Unit 3 – Python programming in Edublocks
 Unit 6 – Flowcharts with Flowol

Year 8: Unit 2 – Understanding computers

Links to future study:

Year 8: Unit 5 – Turtle using Small basic
Unit 6 – Introduction to python programming

Year 9: Unit 2 – Data Representation (Images and Sound)
Unit 3 – Modelling in small basic
Unit 4 – Further programming in Python
Unit 5 – AI and Machine Learning

KS4: GCSE OCR Computer Science:
Unit 2.1 Algorithms
Unit 2.2 Programming fundamentals
Unit 2.4 Boolean Logic

OCR Cambridge National in IT:
TA2 Data and information
TA3 Human-computer interaction

Implementation







Number of lessons: ≈ 7

	Description of lesson content	Written Feedback
1	Logical thinking	
2	Logic gates	
3	Algorithmic thinking 1	
4	Algorithmic thinking 2	
5	Abstraction	
6	Decomposition	
7	Revision and Assessment	

<u>Main Resources</u>	<u>Method of Assessment</u>	<u>Use of Assessment Results</u>
<ul style="list-style-type: none"> PG Online Worksheets Teacher prepared PowerPoint & Activities 	<ul style="list-style-type: none"> Self-Assessed and Peer assessed Written feedback End of Unit Assessment 	<ul style="list-style-type: none"> Used to identify areas of weakness Used to set WWW and EBI based on Assessment

Year 8

Unit 5: Small Basic

Links to the National Curriculum					
Algorithms 	Programming & Development 	Data & Data representation 	Hardware & Processing 	Communication & Networks 	Information Technology 

In this unit, students explore text-based programming using Microsoft Small Basic, with a focus on Turtle Graphics – a visual and engaging way to learn coding fundamentals.

Intent

Knowledge acquired:

- Basic small basic syntax and commands (e.g. turtle.move, turtle.turn)
- Programming structures: sequence, iteration (for loops) and selection (if...then)
- Coordinate systems and angles for movement and direction
- Common programming errors and how to identify them
- Vocabulary: syntax, loop, variable, debug, output etc.

Skills developed:

- Writing and editing small basic programs using Turtle graphics
- Using loops to create patterns and reduce repetition
- Applying angles and coordinates to control turtle movement
- Debugging code to fix syntax and logic errors
- Designing creative outputs using structured code

Understanding:

- How instructions are executed in order (sequence and control flow)
- Why loops are useful for efficiency and pattern creation
- What mathematical concepts like angles and coordinates apply to programming
- The importance of precision and logic in coding
- How visual programming connects to real-world applications and future languages like python

Links to National Curriculum KS2:

Design, write and debug programs that accomplish specific goals... and use logical reasoning to explain how some simple algorithms work.

Links to prior study:

Year 7: Unit 3 – Python Programming with Edublocks
 Unit 5 – HTML coding
 Unit 6 – Flowcharts with Flowol

Year 8: Unit 4 – Computational thinking and logic

Links to future study:

Year 8: Unit 6 – Introduction to python programming

Year 9: Unit 3 – Modelling in small basic
 Unit 4 – Further programming in python

KS4: GCSE OCR Computer Science:
 Unit 2.1 - Algorithms

Implementation







Number of lessons: ≈ 7

	Description of lesson content	Written Feedback	
1	Introduction to Small Basic and turtle		
2	Using the for loop 1		
3	Using the for loop 2		
4	The text window		
5	Using variables		
6	Conditions and branching		
7	Revision and Assessment		

<u>Main Resources</u>	<u>Method of Assessment</u>	<u>Use of Assessment Results</u>
<ul style="list-style-type: none"> PG Online Worksheets Teacher prepared PowerPoint & Activities 	<ul style="list-style-type: none"> Self-Assessed and Peer assessed Written feedback End of Unit Assessment 	<ul style="list-style-type: none"> Used to identify areas of weakness Used to set WWW and EBI based on Assessment

Year 8

Unit 6: Introduction to Python

Links to the National Curriculum					
Algorithms 	Programming & Development 	Data & Data representation 	Hardware & Processing 	Communication & Networks 	Information Technology 

This is an introduction to Python, a powerful but easy-to-use high-level programming language. Python is an object-oriented language. The focus is on getting students to understand the process of developing programs, the importance of writing correct syntax, being able to formulate algorithms for simple programs and debugging their programs. Python is the language that we use at GCSE level, so this unit give the students a flavour for GCSE Computing.

Intent

Knowledge acquired:

- Execute Python programs in both Interactive and Script modes
- Use pseudocode to plan algorithms before coding
- Work with different data types (e.g. strings, integers, floating point)
- Apply assignment statements and arithmetic operators correctly
- Identify and correct syntax and logic errors
- Use relational operators and control structures (,) to manage program flow
- Add comments to explain and document their code
- Describe how a binary search algorithm works
- Compare binary and linear search methods for ordered data
- Design algorithms to solve moderately complex problems (e.g. binary search)
- Handle user input errors gracefully, including error messages and program termination

Skills developed:

- Write structured Python programs using selection and iteration
- Formulate and implement algorithms using pseudocode and Python syntax
- Debug and test their code to ensure reliability and accuracy
- Develop well-documented programs with clear logic and user interaction
- Apply computational thinking to solve real-world problems

Understanding:

- Use Python to solve a range of computational challenges
- Understand and apply key algorithms that demonstrate computational thinking
- Make appropriate use of data types, control structures, and modular design
- Appreciate the importance of planning, testing, and documenting code effectively

Links to National Curriculum KS2:

- 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

Links to prior study:

Year 7: Unit 3 – Python Programming with Edublocks
 Unit 5 – HTML coding
 Unit 6 – Flowcharts with Flowol

Year 8: Unit 4 – Computational thinking and logic

Unit 5 – Small Basic

Links to future study:

Year 9: Unit 3 – Modelling in small basic
 Unit 4 – Further programming in python

KS4: GCSE OCR Computer Science:
 Unit 2.1 - Algorithms
 Unit 2.2 - Programming Fundamentals

Implementation







Number of lessons: ≈ 5

	Description of lesson content	Written Feedback	
1	Strings and variables		
2	Numbers and arithmetic		
3	Selection		
4	Writing Algorithms		
5	Revision and Assessment		

<u>Main Resources</u>	<u>Method of Assessment</u>	<u>Use of Assessment Results</u>
<ul style="list-style-type: none"> • PG Online Worksheets • Teacher prepared PowerPoint • Teacher prepared example Spreadsheets for modelling purposes 	<ul style="list-style-type: none"> • Self-Assessed and Peer assessed • Written feedback • End of Unit Assessment 	<ul style="list-style-type: none"> • Used to identify areas of weakness • Used to set WWW and EBI based on Assessment

Year 9

Unit 1: Spreadsheet Modelling

Links to the National Curriculum					
Algorithms 	Programming & Development 	Data & Data representation 	Hardware & Processing 	Communication & Networks 	Information Technology 

This is a practical, skills-based unit covering the principles of creating and formatting basic spreadsheets to produce and use simple computer models. The unit is centred around creating a financial model for a TV show. Students start by looking at different types of models and then use basic spreadsheet techniques to create and format a simple financial model to calculate the expected income from viewers' voting. The model is then extended to include sales from merchandising, with the introduction of "what if" scenarios. Finally, the students create a seating plan, book seats and calculate income from seat sales.

Intent

Knowledge acquired:

- Charts
- Formulas
- Conditional Formatting
- Data Validation
- Macros

Skills developed:

- Microsoft Excel Skills

Understanding:

- How to input data into a spreadsheet
- How to use this data for formulas and charts
- How to make custom buttons

Links to National Curriculum KS2:

- Understanding simple Boolean Logic
- Creative Projects involving selection, and usage

Links to prior study:

Year 7: Unit 1 - Spreadsheet Modelling – Harry Plotter

Year 8: Unit 1 – Spreadsheet Modelling – Chimp Andy

Links to future study:

KS4: OCR Cambridge Nationals:
L05- to be able to import and manipulate data to develop a solution to meet an identified need.

Implementation







Number of lessons: ≈ 7

Description of lesson content		Written Feedback	
1	Computer modelling	✓	<ul style="list-style-type: none"> • Check to see that students are able to use functions such as sum • Check that students understand absolute cell referencing • Check that students can create charts
2	Financial Model		
3	Financial Model – charts and Macros		
4	What if scenarios		
5	Seat booking	✓	<ul style="list-style-type: none"> • Check that students can use conditional formatting • Check that students can calculate percentages • Check that students can use count If
6	Seat booking – charts and Macros		
7	Revision and Assessment		

<u>Main Resources</u>	<u>Method of Assessment</u>	<u>Use of Assessment Results</u>
<ul style="list-style-type: none"> • Original resources from Teach-ICT • Teacher prepared PowerPoint & Workbook • Teacher prepared example Spreadsheets for modelling purposes 	<ul style="list-style-type: none"> • Self-Assessed and Peer assessed • End of Unit Assessment 	<ul style="list-style-type: none"> • Used to identify areas of weakness • Used to set WWW and EBI based on Assessment

Year 9

Unit 2: Data Representation

Links to the National Curriculum					
Algorithms 	Programming & Development 	Data & Data representation 	Hardware & Processing 	Communication & Networks 	Information Technology 

This unit builds on from the computational thinking unit students study in year 8. Students will learn about how digital images and sounds are created and stored on computers. They will discover how pictures are made up of tiny squares called pixels and how they can measure their quality with resolution. They will explore how colours are made using RGB, and how computer save images using binary. They will get to see how sound is stored in similar way.

Intent

Knowledge acquired:

- What digital images are made of
- Definition of a pixel, resolution and colour depth
- How images are represented in binary
- How colour is formed using red, green, and blue (RGB) values
- Definition of sample, sampling frequency/rate, and sample size
- How sound is represented as binary data

Skills developed:

- Calculate the file size of digital images based on resolution and colour depth
- Create bitmap images and animations using pixel-based tools
- Calculate the number of bits required to represent sound files
- Investigate and compare sound file quality and its impact on file size

Understanding:

- How binary data underpins digital media (images and sound)
- The relationship between image resolution, colour depth, and file size
- How sampling affects sound quality and storage requirements
- Why different media formats vary in quality and efficiency
- How digital representations balance quality with storage constraints

Links to National Curriculum KS2:

"Pupils should be taught to 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."

While this statement focuses on networks, it introduces the idea that computers process and transmit digital data, which is foundational to understanding how images and sound are represented in binary. The Year 9 unit builds on this by teaching students how digital media is stored, measured, and manipulated using binary systems — deepening their understanding of how computers handle different types of information

Links to prior study

Year 8: Unit 2 – Understanding computers
 Unit 4 – Computational thinking and Logic

Links to future study:

Year 9: Unit 5 – AI and machine learning

KS4: OCR GCSE Computer Science:
Unit 1.2 – Memory and Storage
Unit 2.4 – Boolean Logic

Implementation







Number of lessons: ≈ 5

	Description of lesson content		Written Feedback
1	Binary Mosaic	✓	<ul style="list-style-type: none">Understanding of key terminology 'pixel', 'resolution', 'colour depth' and how they affect image quality
2	Image File size	✓	<ul style="list-style-type: none">Can students calculate image file sizes?
3	Images – putting theory into practice		
4	Good vibrations		
5	Representation size (sound)		
6	Investigating file size		
7	Revision and Assessment		

<u>Main Resources</u>	<u>Method of Assessment</u>	<u>Use of Assessment Results</u>
<ul style="list-style-type: none">Adapted worksheets and Powerpoints from NCCE	<ul style="list-style-type: none">Self-Assessed and Peer assessedWritten feedback (L1 & L2)End of Unit Assessment	<ul style="list-style-type: none">Used to identify areas of weaknessUsed to set WWW and EBI based on Assessment

Year 9

Unit 3: Modelling in Small Basic

Links to the National Curriculum					
Algorithms 	Programming & Development 	Data & Data representation 	Hardware & Processing 	Communication & Networks 	Information Technology 

The unit covers Modelling, using Small Basic. Students start to write small programs which will eventually become subroutines in their Lemonade Stand model. The students will discuss some of the different areas in which models are used, from climate change to business enterprises, and the limitations of their own model. The unit moves through the concepts of selection and iteration, giving practice in each of these, before covering subroutines and modular programming.

Intent

Knowledge acquired:

- Programming Syntax
- Small Basic Language
- Decomposition
- Abstraction
- Algorithmic design

Skills developed:

- Small Basic Programming
- Computational Thinking
- Problem Solving
- Troubleshooting

Understanding:

- How to program simple programs
- How to break a big problem down into smaller more manageable steps

Links to National Curriculum KS2:

- Use of Computational Abstractions
- Understand how instructions are stored and executed in a computer
- 1st half of using at least 2 programming languages

Links to prior study:

Year 7: Unit 3 – Python programming using Edublocks
Unit 6 – Flowcharts with Flowol

Year 8: Unit 5 – Introduction to small basic
Unit 6 – Introduction to python programming

Links to future study:

Year 9: Unit 4 – Further programming in Python

KS4 GCSE OCR Computer Science:
Unit 2.1 – Algorithms
Unit 2.2 – Programming Fundamentals
Unit 2.3 – Creating robust programs

OCR Cambridge Nationals:

L05- to be able to import and manipulate data to develop a solution to meet an identified need. (Some elements of Spreadsheet skills are transferable)

Implementation







Number of lessons: ≈ 7

	Description of lesson content		Written Feedback
1	Introduction to small basic		
2	Data types, string concatenation and variables	✓	Checking that students can apply variables, concatenation and formatting
3	Calculations – Lemonade stand		
4	Calculations – Milles per gallon	✓	Checking that students can use code to model calculations.
5	Selection		
6	Selection		
7	Revision and assessment		

<u>Main Resources</u>	<u>Method of Assessment</u>	<u>Use of Assessment Results</u>
<ul style="list-style-type: none"> PG Online Worksheets Teacher prepared PowerPoint Teacher prepared code of Small Basic 	<ul style="list-style-type: none"> Self-Assessed and Peer assessed End of Unit Assessment 	<ul style="list-style-type: none"> Used to identify areas of weakness Used to set WWW and EBI based on Assessment

Year 9

Unit 4: Further programming in python

Links to the National Curriculum					
Algorithms 	Programming & Development 	Data & Data representation 	Hardware & Processing 	Communication & Networks 	Information Technology 

The first lesson has a series of tasks designed to revisit the basic skills that were covered in year 8. Students then use for loops and compare their use with while loops, before moving on to arrays (lists), which are introduced as a new data structure and are used in conjunction with for loops. Procedures and functions with parameters are covered to help students understand the concept and benefits of modular programming. This unit is designed to take students right up to a point where a GCSE in Computing can pick up from and should provide ample experience of programming in order to confirm any decision to pursue Computing as a GCSE option.

Intent

Knowledge acquired:

- Programming Syntax
- Python language
- Decomposition
- Abstraction
- Algorithmic design

Skills developed:

- Python Programming
- Computational Thinking
- Problem Solving
- Troubleshooting

Understanding:

- How to program simple programs
- How to break a big problem down into smaller more manageable steps

Links to National Curriculum KS2:

- Use of Computational Abstractions
- Understand how instructions are stored and executed in a computer
- 2nd half of using at least 2 programming languages

Links to prior study:

Year 7: Unit 3 – Python programming using Edublocks
Unit 6 – Flowcharts with Flowol

Year 8: Unit 4 – Computational thinking and logic
Unit 5 – Small Basic
Unit 6 – Introduction to programming

Year 9: Unit 3 – Modelling in small basic

Links to future study:

KS4: OCR GCSE Computer Science:
Unit 2.1 – Algorithms
Unit 2.2 – Programming fundamentals
Unit 2.3 – Creating robust programs

Implementation







Number of lessons: ≈ 7

	Description of lesson content	Written Feedback	
1	Introduction to python – recall		
2	Casting and if statements		
3	Iteration		
4	Lists		
5	Procedures		
6	Functions		
7	Revision and Assessment		

<u>Main Resources</u>	<u>Method of Assessment</u>	<u>Use of Assessment Results</u>
<ul style="list-style-type: none"> • PG Online Worksheets • Teacher prepared PowerPoint & Activities • Teacher prepared example code to use for modelling purposes 	<ul style="list-style-type: none"> • Self-Assessed and Peer assessed • End of Unit Assessment 	<ul style="list-style-type: none"> • Used to identify areas of weakness • Used to set WWW and EBI based on Assessment

Year 9

Unit 5: AI and Machine Learning

Links to the National Curriculum					
Algorithms 	Programming & Development 	Data & Data representation 	Hardware & Processing 	Communication & Networks 	Information Technology 

This unit allows students to explore how computers learn, make decisions and interact intelligently. Covering key concepts like image recognition, chatbots and the ethics of AI. It introduces students to the fascinating world of Artificial Intelligence and Machine learning through a mix of theory, real-world examples, and hands-on activities.

Intent

Knowledge acquired:

- What artificial intelligence (AI) and machine learning (ML) are, including key definitions and terminology
- The difference between AI, machine learning, and traditional programming
- How AI systems are trained using data and algorithms
- How image recognition and classification systems work
- The concept of the Turing test and how chatbots simulate human conversation
- Ethical considerations of AI, including bias, privacy, and decision-making
- Real-world applications of AI in areas like healthcare, transport, and customer service

Skills developed:

- Analyse and interpret how AI systems make decisions
- Use and evaluate simple AI tools (e.g. image classifiers, chatbots)
- Design and test basic AI models or simulations
- Identify and discuss ethical implications of AI technologies
- Communicate ideas clearly about how AI impacts society and individuals

Understanding:

- How machines learn from data and improve over time
- The role of training data, pattern recognition, and feedback in machine learning
- The limitations and risks of AI, including bias and lack of transparency
- The importance of ethical design and responsible use of intelligent systems
- How AI is shaping the future of work, communication, and decision-making

Links to National Curriculum KS2:

- Use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs.
- By Year 9, students extend this by applying logical reasoning to understand how AI systems are trained, how they make decisions, and how algorithms can reflect bias or error. The unit deepens their understanding of how data and logic underpin modern computing, preparing them for more advanced study in GCSE Computer Science or CNAT IT.

Links to prior study:

Year 7: Python programming in Edublocks

Year 8: Computational thinking and logic

Year 9: Unit 2 – Data representation

Links to future study:

KS4: OCR Computer Science:
 Unit 1.4 Network Security
 Unit 1.6 Ethical, legal, cultural and environmental concerns in computing
 Unit 2.1 algorithms

OCR Cambridge Nationals:

TA1: Design tools – students use logical reasoning and flow-based thinking when exploring how AI systems process data

TA3: Human-Computer Interaction – AI applications like chatbots and image recognition help students understand how users interact with intelligent systems

TA4: Cyber security and legislation – Ethical discussions around AI link to data protection, responsible use, and the implications of automated decision making

Implementation

Number of lessons: ≈ 7

	Description of lesson content	Written Feedback	
1	What is artificial intelligence?		
2	Machine learning		
3	Ethics of artificial intelligence		
4	Image recognition		
5	Turing tests and chat bots		
6	Rate my review		
7	Revision and Assessment		

<u>Main Resources</u>	<u>Method of Assessment</u>	<u>Use of Assessment Results</u>
<ul style="list-style-type: none"> PG Online Worksheets Teacher prepared PowerPoint Teacher prepared example Spreadsheets for modelling purposes Software: MS Access 	<ul style="list-style-type: none"> Self-Assessed and Peer assessed Written Feedback End of Unit Assessment 	<ul style="list-style-type: none"> Used to identify areas of weakness Used to set WWW and EBI based on Assessment