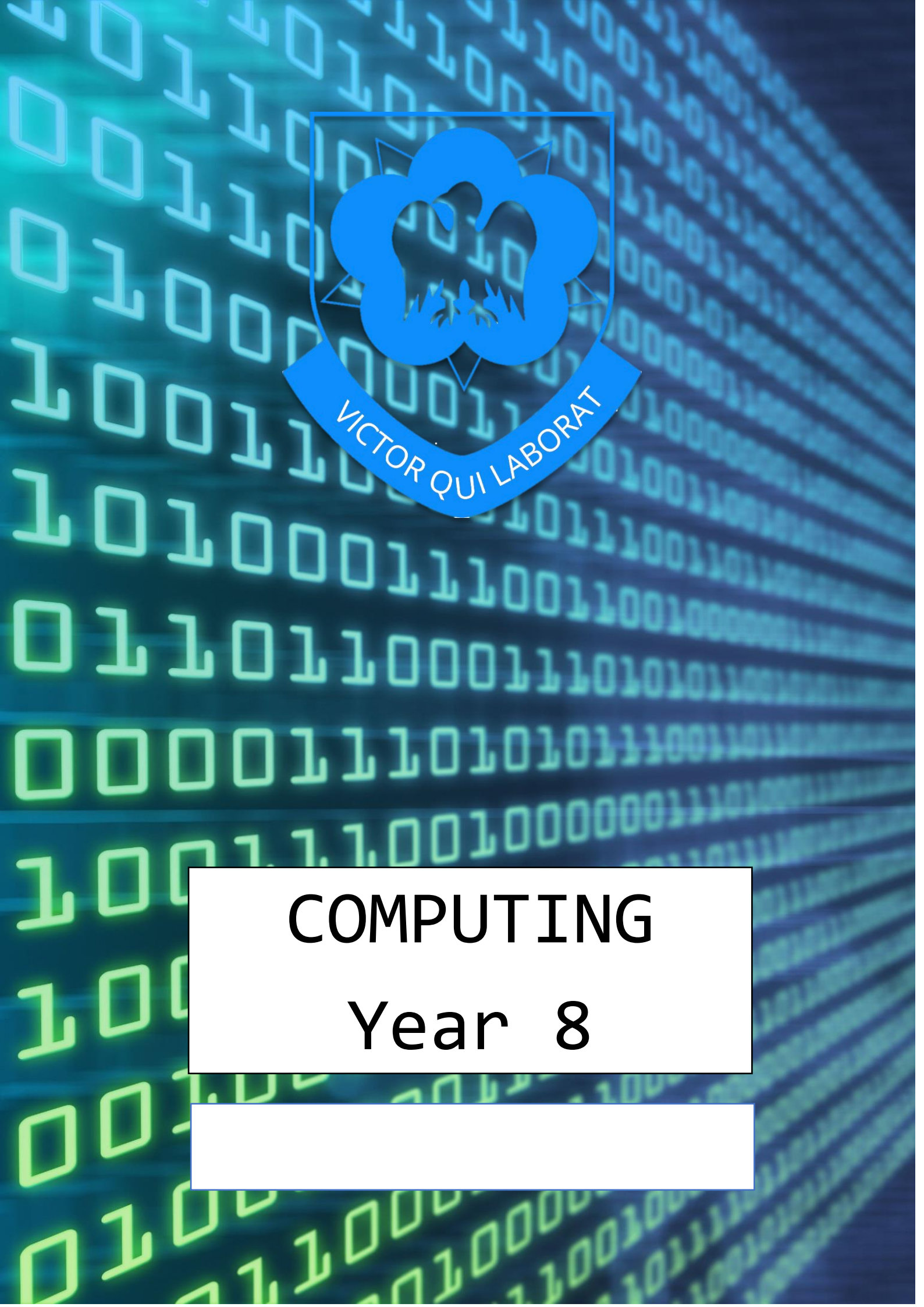

COMPUTING

Year 8

Department for Education

Computing programmes of study: key stages 3 and 4

National curriculum in England

Purpose of study

A high-quality computing education equips pupils to use computational thinking and creativity to understand and change the world. Computing has deep links with mathematics, science, and design and technology, and provides insights into both natural and artificial systems. The core of computing is computer science, in which pupils are taught the principles of information and computation, how digital systems work, and how to put this knowledge to use through programming. Building on this knowledge and understanding, pupils are equipped to use information technology to create programs, systems and a range of content. Computing also ensures that pupils become digitally literate – able to use, and express themselves and develop their ideas through, information and communication technology – at a level suitable for the future workplace and as active participants in a digital world.

Aims

The national curriculum for computing aims to ensure that all pupils:

- 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.

Attainment targets

By the end of each key stage, pupils are expected to know, apply and understand the matters, skills and processes specified in the relevant programme of study.

Schools are not required by law to teach the example content in [square brackets].

Published: September 2013

Subject content

Key stage 3

Pupils should be taught to:

- design, use and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems
- understand several key algorithms that reflect computational thinking [for example, ones for sorting and searching]; use logical reasoning to compare the utility of alternative algorithms for the same problem
- use two or more programming languages, at least one of which is textual, to solve a variety of computational problems; make appropriate use of data structures [for example, lists, tables or arrays]; design and develop modular programs that use procedures or functions
- understand simple Boolean logic [for example, AND, OR and NOT] and some of its uses in circuits and programming; understand how numbers can be represented in binary, and be able to carry out simple operations on binary numbers [for example, binary addition, and conversion between binary and decimal]
- 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
- undertake creative projects that involve selecting, using, and combining multiple applications, preferably across a range of devices, to achieve challenging goals, including collecting and analysing data and meeting the needs of known users
- create, re-use, revise and re-purpose digital artefacts for a given audience, with attention to trustworthiness, design and usability
- understand a range of ways to use technology safely, respectfully, responsibly and securely, including protecting their online identity and privacy; recognise inappropriate content, contact and conduct and know how to report concerns.

Key stage 4

All pupils must have the opportunity to study aspects of information technology and computer science at sufficient depth to allow them to progress to higher levels of study or to a professional career.

All pupils should be taught to:

- develop their capability, creativity and knowledge in computer science, digital media and information technology
- develop and apply their analytic, problem-solving, design, and computational thinking skills
- understand how changes in technology affect safety, including new ways to protect their online privacy and identity, and how to identify and report a range of concerns.

Reference: DFE-00191-2013

Computing Progression Pathways

Pupil Progression	Algorithms	Programming & Development	Data & Data Representation	Hardware & Processing	Communication & Networks	Information Technology
↓	• Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. **(AL)** • Understands that computers need precise instructions. **(AL)** • Demonstrates care and precision to avoid errors. **(AL)**	• Knows that users can develop their own programs, and can demonstrate this by creating a simple program in an environment that does not rely on text e.g. programmable robots etc. **(AL)** • Executes, checks and changes programs. **(AL)** • Understands that programs execute by following precise instructions. **(AL)**	• Recognises that digital content can be represented in many forms. **(AB) (GE)** • Distinguishes between some of these forms and can explain the different ways that they communicate information. **(AB)**	• Understands that computers have no intelligence and that computers can do nothing unless a program is executed. **(AL)** • Recognises that all software executed on digital devices is programmed. **(AL) (AB) (GE)**	• Obtains content from the world wide web using a web browser. **(AL)** • Understands the importance of communicating safely and respectfully online, and the need for keeping personal information private. **(EV)** • Knows what to do when concerned about content or being contacted. **(AL)**	• Uses software under the control of the teacher to create, store and edit digital content using appropriate file and folder names. **(AB) (GE) (DE)** • Understands that people interact with computers. • Shares their use of technology in school. • Knows common uses of information technology beyond the classroom. **(GE)** • Talks about their work and makes changes to improve it. **(EV)**
↓	• Understands that algorithms are implemented on digital devices as programs.**(AL)** • Designs simple algorithms using loops, and selection i.e. if statements. **(AL)** • Uses logical reasoning to predict outcomes. **(AL)** • Detects and corrects errors i.e. debugging, in algorithms. **(AL)**	• Uses arithmetic operators, if statements, and loops, within programs. **(AL)** • Uses logical reasoning to predict the behaviour of programs. **(AL)** • Detects and corrects simple semantic errors i.e. debugging, in programs. **(AL)**	• Recognises different types of data: text, number. **(AB) (GE)** • Appreciates that programs can work with different types of data. **(GE)** • Recognises that data can be structured in tables to make it useful. **(AB) (DE)**	• Recognises that a range of digital devices can be considered a computer. **(AB) (GE)** • Recognises and can use a range of input and output devices. • Understands how programs specify the function of a general purpose computer. **(AB)**	• Navigates the web and can carry out simple web searches to collect digital content. **(AL) (EV)** • Demonstrates use of computers safely and responsibly, knowing a range of ways to report unacceptable content and contact when online.	• Uses technology with increasing independence to purposefully organise digital content. **(AB)** • Shows an awareness for the quality of digital content collected. **(EV)** • Uses a variety of software to manipulate and present digital content: data and information. **(AL)** • Shares their experiences of technology in school and beyond the classroom. **(GE) (EV)** • Talks about their work and makes improvements to solutions based on feedback received.**(EV)**
↓	• Designs solutions (algorithms) that use repetition and two-way selection i.e. if, then and else. **(AL)** • Uses diagrams to express solutions. **(AB)** • Uses logical reasoning to predict outputs, showing an awareness of inputs. **(AL)**	• Creates programs that implement algorithms to achieve given goals. **(AL)** • Declares and assigns variables. **(AB)** • Uses post-tested loop e.g. 'until', and a sequence of selection statements in programs, including an if, then and else statement. **(AL)**	• Understands the difference between data and information. **(AB)** • Knows why sorting data in a flat file can improve searching for information. **(EV)** • Uses filters or can perform single criteria searches for information.**(AL)**	• Knows that computers collect data from various input devices, including sensors and application software. **(AB)** • Understands the difference between hardware and application software, and their roles within a computer system. **(AB)**	• Understands the difference between the internet and internet service e.g. world wide web. **(AB)** • Shows an awareness of, and can use a range of internet services e.g. VOIP. • Recognises what is acceptable and unacceptable behaviour when using technologies and online services.	• Collects, organises and presents data and information in digital content. **(AB)** • Creates digital content to achieve a given goal through combining software packages and internet services to communicate with a wider audience e.g. blogging. **(AL)** • Makes appropriate improvements to solutions based on feedback received, and can comment on the success of the solution. **(EV)**
↓	• Shows an awareness of tasks best completed by humans or computers. **(EV)** • Designs solutions by decomposing a problem and creates a sub-solution for each of these parts. **(DE) (AL) (AB)** • Recognises that different solutions exist for the same problem. **(AL) (AB)**	• Understands the difference between, and appropriately uses if and if, then and else statements. **(AL)** • Uses a variable and relational operators within a loop to govern termination. **(AL) (GE)** • Designs, writes and debugs modular programs using procedures. **(AL) (DE) (AB) (GE)** • Knows that a procedure can be used to hide the detail with sub-solution. **(AL) (DE) (AB) (GE)**	• Performs more complex searches for information e.g. using Boolean and relational operators. **(AL) (GE) (EV)** • Analyses and evaluates data and information, and recognises that poor quality data leads to unreliable results, and inaccurate conclusions. **(AL) (EV)**	• Understands why and when computers are used. **(EV)** • Understands the main functions of the operating system. **(DE) (AB)** • Knows the difference between physical, wireless and mobile networks. **(AB)**	• Understands how to effectively use search engines, and knows how search results are selected, including that search engines use 'web crawler programs'. **(AB) (GE) (EV)** • Selects, combines and uses internet services. **(EV)** • Demonstrates responsible use of technologies and online services, and knows a range of ways to report concerns.	• Makes judgements about digital content when evaluating and repurposing it for a given audience. **(EV) (GE)** • Recognises the audience when designing and creating digital content. **(EV)** • Understands the potential of information technology for collaboration when computers are networked. **(GE)** • Uses criteria to evaluate the quality of solutions, can identify improvements making some refinements to the solution, and future solutions. **(EV)**
↓	• Understands that iteration is the repetition of a process such as a loop. **(AL)** • Recognises that different algorithms exist for the same problem. **(AL) (GE)** • Represents solutions using a structured notation. **(AL) (AB)** • Can identify similarities and differences in situations and can use these to solve problems (pattern recognition). **(GE)**	• Understands that programming bridges the gap between algorithmic solutions and computers. **(AB)** • Has practical experience of a high-level textual language, including using standard libraries when programming. **(AB) (AL)** • Uses a range of operators and expressions e.g. Boolean, and applies them in the context of program control. **(AL)** • Selects the appropriate data types. **(AL) (AB)**	• Knows that digital computers use binary to represent all data. **(AB)** • Understands how bit patterns represent numbers and images. **(AB)** • Knows that computers transfer data in binary. **(AB)** • Understands the relationship between binary and file size (uncompressed). **(AB)** • Defines data types: real numbers and Boolean. **(AB)** • Queries data on one table using a typical query language. **(AB)**	• Recognises and understands the function of the main internal parts of basic computer architecture. **(AB)** • Understands the concepts behind the fetch-execute cycle. **(AB) (AL)** • Knows that there is a range of operating systems and application software for the same hardware. **(AB)**	• Understands how search engines rank search results. **(AL)** • Understands how to construct static web pages using HTML and CSS. **(AL) (AB)** • Understands data transmission between digital computers over networks, including the internet i.e. IP addresses and packet switching. **(AL) (AB)**	• Evaluates the appropriateness of digital devices, internet services and application software to achieve given goals. **(EV)** • Recognises ethical issues surrounding the application of information technology beyond school. • Designs criteria to critically evaluate the quality of solutions, uses the criteria to identify improvements and can make appropriate refinements to the solution. **(EV)**
↓	• Understands a recursive solution to a problem repeatedly applies the same solution to smaller instances of the problem. **(AL) (GE)** • Recognises that some problems share the same characteristics and use the same algorithm to solve both. **(AL) (GE)** • Understands the notion of performance for algorithms and appreciates that some algorithms have different performance characteristics for the same task. **(AL) (EV)**	• Uses nested selection statements. **(AL)** • Appreciates the need for, and writes, custom functions including use of parameters. **(AL) (AB)** • Knows the difference between, and uses appropriately, procedures and functions. **(AL) (AB)** • Understands and uses negation with operators. **(AL)** • Uses and manipulates one dimensional data structures. **(AB)** • Detects and corrects syntactical errors. **(AL)**	• Understands how numbers, images, sounds and character sets use the same bit patterns. **(AB) (GE)** • Performs simple operations using bit patterns e.g. binary addition. **(AB) (AL)** • Understands the relationship between resolution and colour depth, including the effect on file size. **(AB)** • Distinguishes between data used in a simple program (a variable) and the storage structure for that data. **(AB)**	• Understands the von Neumann architecture in relation to the fetch-execute cycle, including how data is stored in memory. **(AB) (GE)** • Understands the basic function and operation of location addressable memory.**(AB)**	• Knows the names of hardware e.g. hubs, routers, switches, and the names of protocols e.g. SMTP, iMAP, POP, FTP, TCP/IP, associated with networking computer systems. **(AB)** • Uses technologies and online services securely, and knows how to identify and report inappropriate conduct. **(AL)**	• Justifies the choice of and independently combines and uses multiple digital devices, internet services and application software to achieve given goals. **(EV)** • Evaluates the trustworthiness of digital content and considers the usability of visual design features when designing and creating digital artifacts for a known audience. **(EV)** • Identifies and explains how the use of technology can impact on society. • Designs criteria for users to evaluate the quality of solutions, uses the feedback from the users to identify improvements and can make appropriate refinements to the solution. **(EV)**
↓	• Recognises that the design of an algorithm is distinct from its expression in a programming language (which will depend on the programming constructs available). **(AL) (AB)** • Evaluates the effectiveness of algorithms and models for similar problems. **(AL) (AB) (GE)** • Recognises where information can be filtered out in generalizing problem solutions. **(AL) (AB) (GE)** • Uses logical reasoning to explain how an algorithm works. **(AL) (AB) (DE)** • Represents algorithms using structured language. **(AL) (DE) (AB)**	• Appreciates the effect of the scope of a variable e.g. a local variable can't be accessed from outside its function. **(AB) (AL)** • Understands and applies parameter passing. **(AB) (GE) (DE)** • Understands the difference between, and uses, both pre-tested e.g. 'while', and post-tested e.g. 'until' loops. **(AL)** • Applies a modular approach to error detection and correction. **(AB) (DE) (GE)**	• Knows the relationship between data representation and data quality. **(AB)** • Understands the relationship between binary and electrical circuits, including Boolean logic. **(AB)** • Understands how and why values are data typed in many different languages when manipulated within programs. **(AB)**	• Knows that processors have instruction sets and that these relate to low-level instructions carried out by a computer. **(AB) (AL) (GE)**	• Knows the purpose of the hardware and protocols associated with networking computer systems. **(AB) (AL)** • Understands the client-server model including how dynamic web pages use server-side scripting and that web servers process and store data entered by users. **(AL) (AB) (DE)** • Recognises that persistence of data on the internet requires careful protection of online identity and privacy.	• Undertakes creative projects that collect, analyse, and evaluate data to meet the needs of a known user group. **(AL) (DE) (EV)** • Effectively designs and creates digital artefacts for a wider or remote audience. **(AL) (DE)** • Considers the properties of media when importing them into digital artefacts. **(AB)** • Documents user feedback, the improvements identified and the refinements made to the solution. **(AB)** • Explains and justifies how the use of technology impacts on society, from the perspective of social, economical, political, legal, ethical and moral issues. **(EV)**
↓	• Designs a solution to a problem that depends on solutions to smaller instances of the same problem (recursion). **(AL) (DE) (AB) (GE)** • Understands that some problems cannot be solved computationally. **(AB) (GE)**	• Designs and writes nested modular programs that enforce reusability utilising sub-routines wherever possible. **(AL) (AB) (GE) (DE)** • Understands the difference between 'While' loop and 'For' loop, which uses a loop counter. **(AL) (AB)** • Understands and uses two dimensional data structures. **(AB) (DE)**	• Performs operations using bit patterns e.g. conversion between binary and hexadecimal, binary subtraction etc. **(AB) (AL) (GE)** • Understands and can explain the need for data compression, and performs simple compression methods. **(AL) (AB)** • Knows what a relational database is, and understands the benefits of storing data in multiple tables. **(AB) (GE) (DE)**	• Has practical experience of a small (hypothetical) low level programming language. **(AB) (AL) (DE) (GE)** • Understands and can explain Moore's Law. **(GE)** • Understands and can explain multitasking by computers. **(AB) (AL) (DE)**	• Understands the hardware associated with networking computer systems, including WANs and LANs, understands their purpose and how they work, including MAC addresses. **(AB) (AL) (DE) (GE)**	• Understands the ethical issues surrounding the application of information technology, and the existence of legal frameworks governing its use e.g. Data Protection Act, Computer Misuse Act, Copyright etc. **(EV)**

Computational Thinking Concept: AB = Abstraction; DE = Decomposition; AL = Algorithmic Thinking; EV = Evaluation; GE = Generalisation

Note: Each of the Progression Pathway statements is underpinned by one-or-more learning outcomes (due for publication in 2014), providing greater detail of what should be taught to achieve each Progression Pathway statement and National Curriculum point of study. Reviewed by Simon Humphreys and Sue Sentance of Computing At School, CAS Master Teachers, and by teachers and academics from the wider CAS community. Computational thinking mapping undertaken by Mark Dorling, Cynthia Selby and John Woollard.

COMPUTING AT SCHOOL
EDUCATE · ENGAGE · ENCOURAGE
In collaboration with BCS, The Chartered Institute for IT

DEPARTMENT Computing

Year 8 and 9

Long-term scheme of learning

Unit/Topic focus	Knowledge being developed	Skills being developed	Assessments (When/what?)	Approximate number of lessons
DESIGN, USE AND EVALUATE COMPUTATIONAL ABSTRACTIONS	develop abstractions to represent physical objects write algorithms for everyday tasks originate useful code in a visual language predict the outcome of statements containing AND, NOT and OR	How to create abstractions Create algorithms and identify the parts Create games and programs using a visual language	End of Topic test and baseline assessment, (6-8 weeks)	16

Unit/Topic focus	Knowledge being developed	Skills being developed	Assessments (When/what?)	Approximate number of lessons
UNDERSTAND ALGORITHMS	use data patterns to represent physical objects identify different algorithms that target the same task originate useful code in a text based language include AND, NOT and OR in information searches follow instructions to produce a software abstraction compare algorithms identify structure in programs identify reasons why some search results are likely to be more important than others	Manipulation of spread sheets, formulas and functions Python construct and Syntax Learn HTML code structures Methods of effectively searching the Internet Names for elements of code and code structure Evaluate solutions to given problems Evaluation and synthesis of data	End of Topic test and baseline assessment, (6-8 weeks)	16

	use software abstractions that model real world systems apply logic to efficiency and effectiveness of algorithms			

Unit/Topic focus	Knowledge being developed	Skills being developed	Assessments (When/what?)	Approximate number of lessons
BE ABLE TO USE PROGRAMMING LANGUAGES	test code relate boolean logic to program flow identify strengths and weaknesses in computational models change variables in an algorithm and predict the effect edit source code to fix a bug use wildcards in searches know how instructions and data are stored choose variable names that aid clarity represent numbers using binary patterns	Learn de bugging skills Flow error correction charts Predict changes and model them adapting code to emulate the outcome Skills to interrogate a database using SQL statements Apply Machine Code instructions to develop programs	End of Topic test and baseline assessment, (6-8 weeks)	9

NAME

TERM 1 Baseline Assessment

Percentage

A2L

DEVELOPING	MEETING	EXCEEDING

BELOW TARGET	ON TARGET	ABOVE TARGET

TERM 2 Binary

	Evidence	Met
Developing	I am aware of binary	
Developing	I require help to convert between binary and denary with ease	
Developing	I can use binary to create a basic image (letter, shape)	
Developing	I can identify one symbol in an algorithm	
Developing	In can create a basic algorithm which contains errors	
Developing	I can explain one of the three search methods	
Developing	With support, I can use some of the topic content to create a themed Binary quiz	
Meeting	I am able to define binary	
Meeting	I sometimes get stuck when converting between binary and denary	
Meeting	I can use binary to create an image	
Meeting	I can identify two symbols in an algorithm	
Meeting	In can create an algorithm and correct any errors	
Meeting	I can explain two of the three search methods	
Meeting	With support for more complex tasks, I can use the topic content to create a themed Binary quiz	
Exceeding	I am able to explain with examples what binary is	
Exceeding	I can convert between binary and denary with ease	
Exceeding	I can use binary to create a complex image and with colour	
Exceeding	I can identify all four symbols in an algorithm	
Exceeding	In can create an algorithm without any errors	
	I can explain three of the three search methods	
Exceeding	Independently I can use the topic content and more to create a themed Binary quiz	

DEVELOPING	MEETING	EXCEEDING

BELOW TARGET	ON TARGET	ABOVE TARGET

What I can do well

What I need to improve

Targets

TERM 3 Spreadsheets

Teacher Assessment based on students evidence in folder

	Evidence	Met
Developing	I can use basic formula with help from others or from my teacher	
Developing	I can use basic functions with help from others or from my teacher	
Developing	I can model basic changes in a spreadsheet	
Developing	I can create a graph	
Developing	I can edit images	
Meeting	I can use formula with little input from others or from my teacher	
Meeting	I can use functions with little input from others or from my teacher	
Meeting	I can make predictions and model changes in a spreadsheet	
Meeting	I can create a graph and label the axis and scales to make it meet the purpose	
Meeting	I can use a range of tools to manipulate images	
Exceeding	I can use formula without input from others or from my teacher	
Exceeding	I can use functions without input from others or from my teacher	
Exceeding	I can make a range of predictions and model changes in a spreadsheet to support my predications	
Exceeding	I can select the most suitable graph and label the axis and scales to make it suitable for the purpose	
Exceeding	I can use a range of tools to manipulate images and produce a quality image	

DEVELOPING	MEETING	EXCEEDING

A2L

BELOW TARGET	ON TARGET	ABOVE TARGET

What I can do well

What I need to improve

Teacher Comment

TERM 4 Websites

Teacher Assessment based the final website that is made

	Evidence	Met
Developing	I can use basic HTML tags to make changes to a website	
Developing	I have completed one or two CSS challenges	
Developing	I created a basic design of my website	
Developing	I can make a website which includes text, images, colour	
Developing	I can create a basic website	
Meeting	I can research and try using HTML tags to make changes to my website	
Meeting	I completed three or more CSS challenges	
Meeting	I created a design of my website with some annotation	
Meeting	I created a website which included with 'media' elements	
Meeting	I create a website where some elements look professional	
Exceeding	Independently I can identify and use appropriate HTML tags to improve a website	
Exceeding	I completed all five CSS challenges	
Exceeding	I planned a detailed design of my website with detailed annotation	
Exceeding	My website uses HTML features to enhance it.	
Exceeding	I can create a website which looks professional business like	

DEVELOPING	MEETING	EXCEEDING

A2L

BELOW TARGET	ON TARGET	ABOVE TARGET

What I can do well

What I need to improve

Teacher Comment

TERM 5 Scratch

Teacher /Pupil Assessment based on the programs

	Evidence	Met
Developing	I know what a variable and a constant are	
Developing	I know what an IF statement is (selection)	
Developing	I know what abstraction, decomposition and algorithms are	
Developing	I know what debugging is	
Developing	I can test my code	
Meeting	I can create variables and a constants	
Meeting	I can use Selection to create two different outcomes	
Meeting	I use elements of abstraction, decomposition and algorithms in my coding	
Meeting	I can use debugging to correct code	
Meeting	I can test and correct my code	
Exceeding	I can create variables and a constants with suitable names	
Exceeding	I can use Selection to create a range of different outcomes	
Exceeding	I use elements of abstraction, decomposition and algorithms in my planning and coding	
Exceeding	I can use a range of debugging skills to correct code	
Exceeding	I can test, solve issues and correct as range of errors	

DEVELOPING	MEETING	EXCEEDING

A2L

BELOW TARGET	ON TARGET	ABOVE TARGET

What I can do well

What I need to improve

Teacher Comment

TERM 6 Databases

Final Examination Percentage:

	Evidence	Met
Developing	I know what a Database is	
Developing	I can use some features of a database	
Developing	With support I can create a query	
Developing	I can use some elements of forms and reports	
Meeting	I can define a Database and give examples	
Meeting	I can use a range of features of a database	
Meeting	I can create a range of queries	
Meeting	I can use elements of forms and reports	
Exceeding	I can define a Database and give every day and business examples	
Exceeding	I can use a wide range of features of a database	
Exceeding	I can select and create the most appropriate queries for the solution	
Exceeding	I can produce, create and use forms and reports appropriately	

DEVELOPING	MEETING	EXCEEDING

A2L

BELOW TARGET	ON TARGET	ABOVE TARGET

What I can do well

What I need to improve

Teacher Comment