Key Stage 3 and 4 Computer Science Curriculum Map 2017/18 - GCSE 9-1

Autumn Term		Spring Term		Summer Term	
Year 7					
Autumn 1	**Autumn 2**	**Spring 1**	**Spring 2**	**Summer 1**	**Summer 2**
Introduction to Computer	**Introduction to Programming TC 1, 5**	**Computer Hardware**	**Introduction to HTML: TC 3, 4**	**Scratch Arcade Game Maker:**	**Micro: Bit Madness: TC 2, 5**
Computer Science at New Mills school Exploring Online Issues: Website Reliability and Quality of Sources of Information Safe & Effective Searching Copyright Issues Online Dangers - Strategies to Stay Safe	language constructs using 'Scratch Blocks' (non-textual approach): Outputs, Variables, Inputs, Selection (IF-ELSE statements), Conditionals	Defining a computer, Identifying the various components of a computer and understanding their function Understanding of how the devices work together (the CPU, RAM, Hard Drive, IO devices) Introduction to the Von-Neumann Architecture.	Basic Tags, Heading, Horizontal Rule, Paragraphs, Fonts, Body (and it's properties), Images, Hyperlinks	Designing Interfaces, Gameplay (and progression) and Algorithms \Code Development, Alpha Testing and Debugging End-User Testing and Evaluations	Understanding the various components of the embedded device Remembering the basics of programming: - Outputs, Variables, Inputs, Selection (IF-ELSE statements) Conditionals, Understanding the concept of compiling and flashing.
Formative Assessment (n/a)	**Summative Assessment (end of term)**	**Summative Assessment (end of half term)**	**Formative Assessment (ongoing)**	**Formative Assessment (1st half)**	**Summative Assessment (end of term)**
Formative tasks covering all aspects of the digital world. These will be carried out throughout the 'My Digital World' unit.	A Summative written assessment (synoptic). Exam style questions covering all learnt aspects of programming. This will be carried out at the end of the term. AO2 - Apply knowledge, understanding and skills AO3 - Analyse and evaluate	A Summative written assessment (synoptic) - exam style question covering the theoretical aspects of hardware. AO1 - Recall, select and communicate AO3 - Analyse and evaluate	An extended HTML project covering Basic Tags, Heading, Horizontal Rule, Paragraphs, Fonts, Body (and it's properties), Images, Hyperlinks	An extended project assessing the progress of the development process of designing and programming a unique, end-user focused, game using the various components of Scratch: Design, Development, Testing, and Evaluation.	A summative extended project assessing the full development process of designing and programming a unique, end-user focused, device making use of the various components of the Micro: Bit computer: Design, Development, Testing, and Evaluation. This will be synoptic. AO2 - Apply knowledge, understanding and skills AO3 - Analyse and evaluate
Year 8					
Introduction to Python	**Binary Bits and Bobs**	**Introduction to Python 2**	**HTML and CSS**	**Scratch Shooter Game Maker:**	**Advanced Scratch**
Outputs, Inputs and Variables, Storage, IF Statements, Problem Solving (Abstraction and Decomposition) Tasks	The Binary Number System, Binary – Denary Conversions, Binary Addition, Binary Representation of Text, Binary Representation of Images, Binary Representation of Sound	Outputs, Inputs and Variables, Storage, IF Statements, Problem Solving (Abstraction and Decomposition) Tasks	HTML Basics, CSS: Text, Images, Divisions, Layout	Designing Interfaces, Gameplay (and progression) and Algorithms Code Development, Alpha Testing and Debugging, End-User Testing and Evaluations	Event Driven Programming, Outputs, Inputs and Variable, Storage, IF Statements, FOR and FOREVER Loops
Formative Assessment (ongoing)	**Summative Assessment (end of term)**	**Formative Assessment (ongoing)**	**Summative Assessment (end of term)**	**Formative Assessment (1st half)**	**Summative Assessment (end of term)**
Formative tasks covering all aspects of Python learnt. These will be carried out throughout the Python unit.	A Summative written assessment (synoptic). Exam style questions covering all aspects of the digital world and binary. AO1 - Recall, select and communicate AO3 - Analyse and evaluate	Formative tasks covering all aspects of Python learnt. These will be carried out throughout the Python unit.	A Summative written assessment (synoptic - exam style question covering the theoretical aspects of Python. AO1 - Recall, select and communicate AO3 - Analyse and evaluate	An extended project assessing the progress of the development process of designing and programming a unique, end-user focused, game using the various components of Scratch: Design, Development, Testing, and Evaluation.	A Summative written assessment (synoptic - exam style question covering the theoretical aspects of programming. AO2 - Apply knowledge, understanding and skills AO3 - Analyse and evaluate
Year 9					
Fundamentals of Computer Systems	**Computing Hardware and Software**	**Computer Networks/Representation of Data in Computer Systems**	**Back to the future HTML, CSS and JavaScript TC 1, 3, 5**	**Scrolling Game Maker Python Programming TC 1,2, 3, 5**	**Python Programming TC 3, 5**
1.1 Computer Systems, 1.2 Importance of Computer Systems in the Modern World, 1.3 Computer System, Reliability 1.4 Professional Standards, 1.5 Ethical, Environmental and Legal Considerations	2.1 The Central Processing Unit (CPU), 2.2 Binary Logic 2.3 Memory, 2.4 Input and Output Devices, 2.5 Secondary Storage, 3.1 Operating Systems, 3.2 Utility Programs, 3.3 Applications Software	Advantages /Disadvantages of Networks Local Area Networks (definition and hardware required), Wide Area Networks (definition and hardware required), Data Packets and the Internet, DNS and the Internet 4 Representation of Data in Computer Systems, 4.1 Units, 4.2 Number, 4.3 Character, 4.4 Images, 4.5 Sound, 4.6 Instructions	Visiting famous CS pioneers and exploring their work: George Boole – Boolean Logic Tim Berners-Lee – HTML and WWW Charles Babbage – The Difference Engine and Problem Solving - Alan Turing – Code Breaking HTML Intermediate, CSS: Text, Images, Divisions, Layout	Designing Interfaces, Gameplay (and progression) and Algorithms, Code Development, Alpha Testing and Debugging, End-User Testing and Evaluations Review of Year 9:, Outputs, Inputs and Variable, Storage IF Statements Advancing Knowledge: FOR Loops, WHILE Loops, Problem Solving (Abstraction and Decomposition) Tasks	7.1 Algorithms, 7.2 Programming constructs, 7.3 Data structures, 7.4 Data Types, 7.5 Programming, Languages, 7.6 Control Flow in Imperative Languages 7.7 Handling Data in Algorithms, 7.8 Security, 7.9 Authentication, 7.10 Testing
Formative Assessment (ongoing)	**Summative Assessment (end of term)**	**Summative Assessment (end of half term)**	**Formative Assessment (ongoing)**	**Formative Assessment (ongoing)**	**Summative Assessment (end of term)**
Formative tasks covering all aspects of Fundamentals of Computer Systems. These will be carried out throughout the unit - Fundamentals of Computer Systems: Weekly Questions and answers.	A Summative written assessment (synoptic). Exam style questions covering all covered aspects of Software and Computing Hardware: This will be carried out at the end of term. AO1 - Recall, select and communicate AO3 - Analyse and evaluate	A Summative written assessment (synoptic - exam style questions covering the theoretical aspects of Representation of Data in Computer Systems and Networks. AO1 - Recall, select and communicate AO3 - Analyse and evaluate	In addition to this there will be an extended HTML project.	An extended project assessing the full development process of coding a computer game: Design, Development, Testing, Evaluation Evidence of testing with resulting improvements documented and a detailed evaluation of the success of the project. , Practical tasks, Peer and self-assessment.	A Summative written assessment (synoptic - exam style questions covering the theoretical aspects of programming. Question and answer - Exam style questions on all topics AO2 - Apply knowledge, understanding and skills AO3 - Analyse and evaluate
Year 10					
Python Programming TC 3, 5	**Python Programming TC 3, 4, 5**	**Python Programming TC 1, 2, 5**	**Python Programming TC 2, 3, 4**	**Networks**	**Networks and Internet**
7.1 Algorithms, 7.2 Programming constructs, 7.3 Data structures, 7.4 Data Types, 7.5 Programming, Languages, 7.6 Control Flow in Imperative Languages 7.7 Handling Data in Algorithms, 7.8 Security, 7.9 Authentication, 7.10 Testing	7.1 Algorithms, 7.2 Programming constructs, 7.3 Data structures, 7.4 Data Types, 7.5 Programming, Languages, 7.6 Control Flow in Imperative Languages 7.7 Handling Data in Algorithms, 7.8 Security, 7.9 Authentication, 7.10 Testing	7.1 Algorithms, 7.2 Programming constructs, 7.3 Data structures, 7.4 Data Types, 7.5 Programming, Languages, 7.6 Control Flow in Imperative Languages 7.7 Handling Data in Algorithms, 7.8 Security, 7.9 Authentication, 7.10 Testing	7.1 Algorithms, 7.2 Programming constructs, 7.3 Data structures, 7.4 Data Types, 7.5 Programming, Languages, 7.6 Control Flow in Imperative Languages 7.7 Handling Data in Algorithms, 7.8 Security, 7.9 Authentication, 7.10 Testing	5.1 Explain the characteristics of networks, 5.2 common network topologies, 5.3 connectivity, 5.4 circuit and packet switching. 5.5 importance/use of contemporary network Protocols, 5.6 TCP/IP packet. Layers and the TCP/IP 5- layer model, methods of routing traffic on a network and calculate routing costs.	5.5 network protocols, Ethernet,Wi-Fi, TCP/IP, HTTP, HTTPS, FTP and email protocols. 5.6 TCP/IP packet. importance of layers and the TCP/IP 5- layer odel.6.1 Domain Name System (DNS) servers and Internet Protocol (IP) addresses work.
Formative Assessment (ongoing)	**Summative Assessment**	**Summative Assessment (continued)**	**Summative Assessment (continued)**	**Formative Assessment (ongoing)**	**Summative Assessment (end of term)**
Mock assessments - Formative tasks covering all aspects of Fundamentals of Python Programming. These will be carried out throughout the unit: Weekly Questions and answers.	Controlled assessment – Unit A452 Candidates carry out a practical investigation of a topic chosen from a set of options supplied by OCR - An investigative task - Approx. 20 hours - 45 marks - 30% of the qualification. AO2 - Apply knowledge, understanding and skills AO3 - Analyse and evaluate	Controlled assessment – Unit A452 Candidates carry out a practical investigation of a topic chosen from a set of options supplied by OCR - An investigative task - Approx. 20 hours - 45 marks - 30% of the qualification. AO2 - Apply knowledge, understanding and skills AO3 - Analyse and evaluate	Controlled assessment – Unit A452 Candidates carry out a practical investigation of a topic chosen from a set of options supplied by OCR - An investigative task - Approx. 20 hours - 45 marks - 30% of the qualification.	Formative tasks covering all aspects of Networks learnt. These will be carried out throughout the Network units. Weekly Questions and answers	A Summative written assessment (synoptic – mock exam style questions covering the theoretical aspects of networking). Question and answer - Exam style questions on all topics. AO1 - Recall, select and communicate AO3 - Analyse and evaluate
Year 11					
Programming TC 1, 2, 3, 4, 5	Programming TC 1, 2, 3, 4, 5	**Fundamentals of Computer Systems**	**Computing Hardware and Software**	**Representation of Data in Computer Systems**	**Exam Prep TC 1, 2, 3, 4, 5**
7.1 Algorithms, 7.2 Programming constructs, 7.3 Data structures, 7.4 Data Types, 7.5 Programming, Languages, 7.6 Control Flow in Imperative Languages, 7.7 Handling Data in Algorithms, 7.8 Security, 7.9 Authentication, 7.10 Testing	7.1 Algorithms, 7.2 Programming constructs, 7.3 Data structures, 7.4 Data Types, 7.5 Programming, Languages, 7.6 Control Flow in Imperative Languages, 7.7 Handling Data in Algorithms, 7.8 Security, 7.9 Authentication, 7.10 Testing	1.1 Computer Systems, 1.2 Importance of Computer Systems in the Modern World, 1.3 Computer System, Reliability 1.4 Professional Standards, 1.5 Ethical, Environmental and Legal Considerations	2.1The Central Processing Unit (CPU) 2.2 Binary Logic 2.3 Memory, 2.4 Input and Output Devices, 2.5 Secondary Storage, 3.1 Operating Systems, 3.2 Utility Programs, 3.3 Applications Software	4 Representation of Data in Computer Systems, 4.1 Units, 4.2 Number, 4.3 Character, 4.4 Images, 4.5 Sound, 4.6 Instructions	All of the units to be covered in revision sessions.
Summative/Formative Assessment	**Summative/Formative Assessment (end of term)**	**Summative/Formative Assessment**	**Summative/Formative Assessment**	**Summative/Formative Assessment**	**Summative Assessment**
Mock assessments - Summative tasks covering all aspects of Fundamentals of Programming. These will be carried out throughout the unit: AO2 - Apply knowledge, understanding and skills AO3 - Analyse and evaluate	Mock assessments - Summative tasks covering all aspects of Fundamentals of Programming. These will be carried out throughout the unit: AO2 - Apply knowledge, understanding and skills AO3 - Analyse and evaluate	Mock assessments - Summative tasks covering all aspects of Fundamentals of Computer Systems. These will be carried out throughout the unit: AO1 - Recall, select and communicate AO3 - Analyse and evaluate	Mock assessments - Summative tasks covering all aspects of Fundamentals of Hardware/Software. These will be carried out throughout the unit: AO1 - Recall, select and communicate AO3 - Analyse and evaluate	Mock assessments - Summative tasks covering all aspects of Fundamentals of Representation of Data. These will be carried out throughout the unit: AO1 - Recall, select and communicate AO3 - Analyse and evaluate	Mock assessments - Exam style questions - Practice papers AO1 - Recall, select and communicate AO2 - Apply knowledge, understanding and skills AO3 - Analyse and evaluate

Key Stage 3 and 4 Computer Science Curriculum Map 2017/18 - GCSE 9-1

	Algorithms	Programming & Development	Data & Data Representation	Hardware & Processing	Communication & Networks	Information Technology
	AO2 - Apply knowledge, understanding and skills **AO3 - Analyse and evaluate**	**AO2 - Apply knowledge, understanding and skills** **AO3 - Analyse and evaluate**	**AO1 - Recall, select and communicate** **AO3 - Analyse and evaluate**	**AO1 - Recall, select and communicate** **AO3 - Analyse and evaluate**	**AO1 - Recall, select and communicate** **AO3 - Analyse and evaluate**	**AO1 - Recall, select and communicate** **AO3 - Analyse and evaluate**
↓	Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. Understands that computers need precise instructions. Demonstrates care and precision to avoid errors.	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. Executes, checks and changes programs. Understands that programs execute by following precise instructions.	Recognises that digital content can be represented in many forms. Distinguishes between some of these forms and can explain the different ways that they communicate information.	Understands that computers have no intelligence and that computers can do nothing unless a program is executed. Recognises that all software executed on digital devices is programmed.	Obtains content from the world wide web using a web browser. Understands the importance of communicating safely and respectfully online, and the need for keeping personal information private. Knows what to do when concerned about content or being contacted.	Uses software under the control of the teacher to create, store and edit digital content using appropriate file and folder names. Understands that people interact with computers. Shares their use of technology in school. Knows common uses of information technology beyond the classroom. Talks about their work and makes changes to improve it.
↓	Understands that algorithms are implemented on digital devices as programs. Designs simple algorithms using loops, and selection i.e. if statements. Uses logical reasoning to predict outcomes. Detects and corrects errors i.e. debugging, in algorithms.	Uses arithmetic operators, if statements, and loops, within programs. Uses logical reasoning to predict the behaviour of programs. Detects and corrects simple semantic errors i.e. debugging, in programs	Recognises different types of data: text, number. Appreciates that programs can work with different types of data. Recognises that data can be structured in tables to make it useful.	Recognises that a range of digital devices can be considered a computer. Recognises and can use a range of input and output devices. Understands how programs specify the function of a general purpose computer	Navigates the web and can carry out simple web searches to collect digital content. 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. Shows an awareness for the quality of digital content collected. Uses a variety of software to manipulate and present digital content: data and information. Shares their experiences of technology in school and beyond the classroom. Talks about their work and makes improvements to solutions based on feedback received
↓	Designs solutions (algorithms) that use repetition and two-way selection i.e. if, then and else. Uses diagrams to express solutions. Uses logical reasoning to predict outputs, showing an awareness of inputs	Creates programs that implement algorithms to achieve given goals. Declares and assigns variables. Uses post-tested loop e.g. 'until', and a sequence of selection statements in programs, including an if, then and else statement.	Understands the difference between data and information. Knows why sorting data in a flat file can improve searching for information. Uses filters or can perform single criteria searches for information	Knows that computers collect data from various input devices, including sensors and application software. Understands the difference between hardware and application software, and their roles within a computer system.	Understands the difference between the internet and internet service e.g. world wide web. 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. Creates digital content to achieve a given goal through combining software packages and internet services to communicate with a wider audience e.g. blogging. Makes appropriate improvements to solutions based on feedback received, and can comment on the success of the solution.
↓	Shows an awareness of tasks best completed by humans or computers. Designs solutions by decomposing a problem and creates a sub-solution for each of these parts (decomposition). Recognises that different solutions exist for the same problem.	Understands the difference between, and appropriately uses if and if, then and else statements. Uses a variable and relational operators within a loop to govern termination. Designs, writes and debugs modular programs using procedures. Knows that a procedure can be used to hide the detail with sub-solution (procedural abstraction).	Performs more complex searches for information e.g. using Boolean and relational operators. Analyses and evaluates data and information, and recognises that poor quality data leads to unreliable results, and inaccurate conclusions.	Understands why and when computers are used. Understands the main functions of the operating system. Knows the difference between physical, wireless and mobile networks.	Understands how to effectively use search engines, and knows how search results are selected, including that search engines use 'web crawler programs'. Selects, combines and uses internet services. 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. Recognises the audience when designing and creating digital content. Understands the potential of information technology for collaboration when computers are networked. Uses criteria to evaluate the quality of solutions, can identify improvements making some refinements to the solution, and future solutions.
↓	Understands that iteration is the repetition of a process such as a loop. Recognises that different algorithms exist for the same problem. Represents solutions using a structured notation. Can identify similarities and differences in situations and can use these to solve problems (pattern recognition).	Understands that programming bridges the gap between algorithmic solutions and computers. Has practical experience of a high-level textual language, including using standard libraries when programming. Uses a range of operators and expressions e.g. Boolean, and applies them in the context of program control. Selects the appropriate data types.	Knows that digital computers use binary to represent all data. Understands how bit patterns represent numbers and images. Knows that computers transfer data in binary. Understands the relationship between binary and file size (uncompressed). Defines data types: real numbers and Boolean. Queries data on one table using a typical query language.	Recognises and understands the function of the main internal parts of basic computer architecture. Understands the concepts behind the fetch-execute cycle. Knows that there is a range of operating systems and application software for the same hardware.	Understands how search engines rank search results. Understands how to construct static web pages using HTML and CSS. Understands data transmission between digital computers over networks, including the internet i.e. IP addresses and packet switching.	Evaluates the appropriateness of digital devices, internet services and application software to achieve given goals. 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.
↓	Understands a recursive solution to a problem repeatedly applies the same solution to smaller instances of the problem. Recognises that some problems share the same characteristics and use the same algorithm to solve both (generalisation). Understands the notion of performance for algorithms and appreciates that some algorithms have different performance characteristics for the same task.	Uses nested selection statements. Appreciates the need for, and writes, custom functions including use of parameters. Knows the difference between, and uses appropriately, procedures and functions. Understands and uses negation with operators. Uses and manipulates one dimensional data structures. Detects and corrects syntactical errors.	Understands how numbers, images, sounds and character sets use the same bit patterns. Performs simple operations using bit patterns e.g. binary addition. Understands the relationship between resolution and colour depth, including the effect on file size. Distinguishes between data used in a simple program (a variable) and the storage structure for that data.	Understands the von Neumann architecture in relation to the fetch-execute cycle, including how data is stored in memory. Understands the basic function and operation of location addressable memory.	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. Uses technologies and online services securely, and knows how to identify and report inappropriate conduct.	Justifies the choice of and independently combines and uses multiple digital devices, internet services and application software to achieve given goals. Evaluates the trustworthiness of digital content and considers the usability of visual design features when designing and creating digital artefacts for a known audience. 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.
↓	Recognises that the design of an algorithm is distinct from its expression in a programming language (which will depend on the programming constructs available). Evaluates the effectiveness of algorithms and models for similar problems. Recognises where information can be filtered out in generalising problem solutions (abstraction). Uses logical reasoning to explain how an algorithm works. Represents algorithms using structured language.	Appreciates the effect of the scope of a variable e.g. a local variable can't be accessed from outside its function. Understands and applies parameter passing. Understands the difference between, and uses, both pre-tested e.g. 'while', and post-tested e.g. 'until' loops. Applies a modular approach to error detection and correction.	Knows the relationship between data representation and data quality. Understands the relationship between binary and electrical circuits, including Boolean logic. Understands how and why values are data typed in many different languages when manipulated within programs	Knows that processors have instruction sets and that these relate to low-level instructions carried out by a computer.	Knows the purpose of the hardware and protocols associated with networking computer systems. 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. 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. Effectively designs and creates digital artefacts for a wider or remote audience. Considers the properties of media when importing them into digital artefacts. Documents user feedback, the improvements identified and the refinements made to the solution. Explains and justifies how the use of technology impacts on society, from the perspective of social, economic, political, legal, ethical and moral issues.
⇩	Designs a solution to a problem that depends on solutions to smaller instances of the same problem (recursion). Understands that some problems cannot be solved computationally.	Designs and writes nested modular programs that enforce reusability utilising sub-routines wherever possible. Understands the difference between 'While' loop and 'For' loop, which uses a loop counter. Understands and uses two dimensional data structures.	Performs operations using bit patterns e.g. conversion between binary and hexadecimal, binary subtraction etc. Understands and can explain the need for data compression, and performs simple compression methods. Knows what a relational database is, and understands the benefits of storing data in multiple tables.	Has practical experience of a small (hypothetical) low level programming language. Understands and can explain Moore's Law. Understands and can explain multitasking by computers.	Understands the hardware associated with networking computer systems, including WANs and LANs, understands their purpose and how they work, including MAC addresses	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.

COMPUTER SCIENCE ASSESSMENT OBJECTIVES and THRESHOLD CONCEPTS

ASSESSMENT OBJECTIVES

There are three Assessment Objectives in OCR GCSE (9–1) in Computer Science. These are detailed in the table below. Learners are expected to:

Assessment Objectives	
AO1	Demonstrate knowledge and understanding of the key concepts and principles of Computer Science.
AO2	Apply knowledge and understanding of key concepts and principles of Computer Science.
AO3	Analyse problems in computational terms: • To make reasoned judgements • To design, program, evaluate and refine solutions.

THRESHOLD CONCEPTS:

The focus of the new programme of study moves towards **programming** and other aspects of computer science. Programming has been part of the ICT national curriculum for some time but has frequently been overlooked or treated superficially. However, there is more to computer science than programming. Computer science incorporates techniques and methods for solving problems and advancing knowledge, and includes a distinct way of thinking and working that sets it apart from other disciplines. The role of programming in computer science is similar to that of practical work in other sciences – it provides motivation and a context within which ideas are brought to life.

Computational thinking is core to the programme of study. It is the process of *recognising* aspects of computation in the world that surrounds us, and *applying* tools and techniques from computing to understand and reason about both natural and artificial systems and processes. Computational thinking provides a powerful framework for studying computing, with wide application beyond computing itself. It allows pupils to tackle problems, to break them down into solvable chunks and to devise **algorithms** to solve them.

In summary, the key **threshold concepts** in computational thinking involve:

1. **Decomposition**
2. Pattern Recognition
3. **Abstraction**
4. Pattern Generalisation
5. Algorithm Design.

- **Decomposition** is breaking a problem down into its components, each of which can be tackled individually and further decomposed.
- **Pattern recognition** is looking for similarities in the behaviours and states of the system you are trying to model.
- **Abstraction** helps you only use the detail absolutely necessary for the functioning of the system.
- **Pattern generalisation** allows us to define concepts in their simplest form and to re-use the definition for all instances of that concept.
- **An algorithm** is a precise method for solving a given problem.

THE CONCEPTS

(1) A good Computer Science student can break a problem down into its components, each of which can be tackled individually and further decomposed.
(2) A good Computer Science student can look for similarities in the behaviours and states of the systems they are trying to model.
(3) A good Computer Science student uses only the detail absolutely necessary for the functioning of the system.
(4) A good Computer Science student can define concepts in their simplest form and to re-use the definition for all instances of that concept.
(5) A good Computer Science student can recognises that a**n algorithm** is a precise method for solving a given problem.