YEAR 8	Autumn Term (Cycle 1)	Spring Term (Cycle 2)	Summer Term (Cycle 3)
Students will know and remember…	**Representations: From clay to silicon:** Use of binary representations How representations are used to store, communicate and process information How representations are appropriate for different tasks Use of characters to represent sequences of symbols How to measure the length of a representation as the number of symbols it contains How symbols are carried on physical media How to explain what binary digits (bits) are, in terms of familiar symbols such as digits or letters How to measure the size or length of a sequence of bits as the number of binary digits that it contains How to describe how natural numbers are represented as sequences of binary digits How to convert a decimal to binary and vice versa How to convert between different units and multiples of representation size How to provide examples of the different ways that binary digits are physically represented in digital devices	**Introduction to Python:** How to describe what algorithms and programs are and how they differ That a program written in a programming language needs to be translated in order to be executed by a machine How to write simple Python programs that display messages, assign values to variables and receive keyboard input How to locate and correct common syntax errors How to describe the semantics of assignment statements How to use simple arithmetic expressions in assignment statements to calculate values How to receive input from the keyboard and convert it to a numerical value How to use relational operators to form logical expressions How to use binary expressions (**if, else** statements) to control the flow of program execution How to generate and use random integers How to use multi-branch selection (**if, elif, else** statements) to control the flow of program execution How to describe how iteration (**while** statements) controls the flow of program execution How to use iteration (**while** loops) to control the flow of program execution How to use variables as counters in iterative programs How to combine iteration and selection to control the flow of program execution How to use Boolean variables as flags	**Computing Systems:** That a general-purpose computing system is a device for executing programs That a program is a sequence of instructions that specify operations that are to be performed on data The difference between a general-purpose computing system and a purpose-built device How to describe the function of the hardware components used in computing systems How to describe how the hardware components used in computing systems work together in order to execute programs That all computing systems, regardless of form, have a similar structure (architecture) How to analyse how the hardware components used in computing systems work together in order to execute programs How to define what an operating system is, and recall its role in controlling program execution How to describe the NOT, AND and OR logical operators and how they are used to form logical expressions How to use logic gates to construct logic circuits and associate these with logical operators and expressions How to describe how hardware is built out of increasingly complex logic circuits That, since hardware is built out of logic circuits, data and instructions alike need to be represented using binary digits

			How to provide broad definitions of ‘artificial intelligence’ and ‘machine learning’ How to identify examples of artificial intelligence and machine learning in the real world How to describe the steps involved in training machines to perform tasks (gathering data, training, testing) How to describe how machine learning differs from traditional programming How to associate the use of artificial intelligence with moral dilemmas

So that they can…	**Representations: from clay to silicon:**	**Introduction to Python:**	**Computing Systems:**
	Understand the use and characteristics of representations Be prepared for their encounters with binary representations in the context of computing Understand how to encode, transmit and decode short messages Understand text representations using sequences of symbols and distinguish between them Understand the way in which symbols are embodied in physical media Associate binary digits with familiar sets of symbols such as letters and decimal digits Solve simple problems that reinforce the connection between alphanumeric information and its binary representation Consider why binary digits are predominantly used in conjunction with computing systems Build upon their familiarity with using a decimal numbering system Draw analogies with how numbers can be represented using binary Convert between binary and decimal Understand bytes and the prefixes used for measuring representation size such as kilo-, mega-, giga- and tera-.	Write and execute their first programs in Python Understand the basics of displaying messages, assigning values to variables, and receiving input from the keyboard Move on from their previous programming knowledge through Scratch Understand what algorithms and programs are and how they are different Build an understanding of what it means to express instructions in a formal language and how these instructions can eventually be executed by a machine Gain a deeper understanding of assignments and explicitly address some of the common misconceptions around the semantics of assignment statements Understand how to use arithmetic expressions and receive numerical input from the keyboard Construct their own short programs Understand how to use selection and randomness to develop a number guessing game Use multi-branch selection Comprehend when and how selection should be used Understand the mechanics of how iteration works Import and use functions from modules so they can understand how a text-based language can be more powerful than block-based languages Combine iteration and selection to create a times tables practice game	Compare calculating machines from the past to modern general-purpose computers Connect the abstract idea of a program to the applications they use Execute programs themselves Discover how all computing systems, regardless of form or capabilities, make use of the same components: a processor, memory, storage, input and output and devices, and communication components Understand how each of the above components work together in order to execute programs Understand the operating system, which is responsible for managing the complexity of modern computing devices Master the use of logical expressions in software Bridge the gap between logic and circuits and make the direct link between them Understand the hierarchy of a computing system, from programs to the hardware responsible for executing the programs Define the term ‘artificial intelligence’ Investigate machine learning and its relationship with conventional programming Gain insight into what training a model involves and the ethical considerations that are tied into building any system that makes decisions