Curriculum overview

Subject	Computing	**Year group**	8
Vision statement:	At Landau Forte our curriculum exists to ensure all students regardless of background and ability have the opportunity to unlock their potential. We are committed to students being challenged from their previous key stage learning experiences. Our broad and balanced curriculum is ambitious, coherently planned and sequenced, and will provide the platform for preparing students with the foundations for examination success. Our Curriculum Intent has been informed by a wide variety of researchers and is steeped in evidence based research. Christine Counsell summarises the aspiration of our curriculum to empower all learners creating a pathway to success in university, their career and life: *'A curriculum exists to change the pupil, to give the pupil new power. One acid test for a curriculum is whether it enables even lower attaining or disadvantaged pupils to clamber into the discourse and practices of educated people, so that they gain powers of the powerful.'* As well as excellent academic success we aim to ensure our students leave us as polite and well-rounded young adults. Our new core values of Compassion, Courage and Curiosity are currently being embedded throughout our curriculum offer to ensure we continue to meet our social, emotional, spiritual and moral obligations.		
Curriculum intent:	Computing will be central to everything students do in their future lives. This subject gives students the opportunity to utilise technology to enhance the way they live and work. It will also be used as a lens to develop their understanding of the world around them. ***In essence, computing should be seen as an underpinning subject that facilitates new learning and thinking in all other areas. The computer should be a tool that pupils use in the same way as a calculator or a pen.*** As outlined within the National Curriculum: "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 students 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, students are equipped to use **information technology** to create programs, systems and a range of content. Computing also ensures that students 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. The need to use technology with care and compassion should be considered throughout all lessons.		

Threshold Concepts (TCs):	1. We count in tens as it is convenient for us using our hands; computers count in two's as it is convenient for them using switches 2. Computer programs change what they do based how they are used so we have to make programs where we don't know what information will be provided 3. Computers make it look like they are doing many things all at once when really they are doing one thing at a time very quickly 4. Solutions to problems in a computer require many repeated steps and we should find ways to shortcut this even if it requires more thought in the short-term 5. Drawing on a computer screen is like drawing on a 2D graph 6. Problems have usually been solved before (pattern recognition)
KS2 National Curriculum summary:	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. Pupils should be taught to: • design, write and debug programs that accomplish specific goals, including controlling or simulating physical systems; solve problems by decomposing them into smaller parts • use sequence, selection, and repetition in programs; work with variables and various forms of input and output • use logical reasoning to explain how some simple algorithms work and to detect and correct errors in algorithms and programs • understand computer networks, including the internet; how they can provide multiple services, such as the World Wide Web, and the opportunities they offer for communication and collaboration • use search technologies effectively, appreciate how results are selected and ranked, and be discerning in evaluating digital content • select, use and combine a variety of software (including internet services) on a range of digital devices to design and create a range of programs, systems and content that accomplish given goals, including collecting, analysing, evaluating and presenting data and information • use technology safely, respectfully and responsibly; recognise acceptable/unacceptable behaviour; identify a range of ways to report concerns about content and contact
Learner skills:	CRITICAL THINKING ORACY COLLABORATION SELF QUIZZING ADAPTABILITY ORGANISATION

	Term 1 Aug-Oct	Term 2 Nov-Dec	Term 3 Jan-Feb	Term 4 Mar-Apr	Term 5 Apr-May	Term 6 Jun-Jul
The Big Question	**How do we tell computers what to do?**					
Big picture questions:	**How do computers count?**	**How do computers understand what we want them to do?**	**How do computers do so many things?**	**How are websites built?**	**How do computers draw?**	**How can we build complex computer solutions?**
Content (Linked to TCs):	TC1 • How representations can have been used in the past • That representations are used to store, communicate, and process information • Why different representations are appropriate for different tasks • That characters can be represented as sequences of symbols and list examples of character coding schemes • How to measure the length of a representation as the number of symbols that it contains • How symbols can be carried on physical media • What binary digits (bits) are, in terms of familiar symbols such as digits or letters	TC2 • What algorithms and programs are • That a program written in a programming language needs to be translated in order to be executed • 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 • 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 in expressions • How to use binary selection (if, else	TC3 • That a general-purpose computing system is a device for executing programs • That a program is a sequence of instructions that specify operations to be performed • The difference between a general-purpose computing system and a purpose-built device • The functions of hardware components • How hardware components used in computing systems work together in order to execute programs • That all computing systems have a similar 'architecture' • What an operating system is, and recall its role in controlling program execution • What the NOT, AND, and OR logical operators	TC4 • What HTML is • How to use HTML to structure static web pages • How to modify HTML tags using inline styling to improve the appearance of web pages • How to display images within a web page • How to apply HTML tags to construct a web page structure from a provided design • What CSS is • How to use CSS to style static web pages • What the benefits of using CSS to style pages are instead of in-line formatting • What a search engine is • How search engines 'crawl' through the World Wide Web and how they select and rank results	TC5 • How to draw basic shapes with different properties • How to manipulate individual objects • How to manipulate groups of objects • How to combine paths by applying operations • How to convert objects to paths • How to draw paths • How to edit path nodes • How to combine multiple tools and techniques to create a vector graphic design • What vector graphics are • Where using vector graphics would be appropriate • How to peer assess project work • How to improve project work based on feedback	TC6 • When a problem needs to be broken down • How to implement and customise GUI elements to meet the needs of the user • That events can control the flow of a program • How to use user input in an event-driven programming environment • How to use variables in an event-driven programming environment • How to identify and fix common coding errors • How to pass the value of a variable into an object • Why the need to establish user needs when completing a creative project is important • How to apply decomposition to break down a larger problem

	• How to measure the size or length of a sequence of bits as the number of binary digits that it contains • How natural numbers are represented as sequences of binary digits • How to convert a decimal number to binary and vice versa • How to convert between different units and multiples of representation size • The different ways that binary digits are physically represented in digital devices	statements) to control the flow of program execution • How to generate and use random integers • How to use multi-branch selection to control the flow of program execution • 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	are and how they are used to form logical expressions • How to use logic gates to construct logic circuits • How hardware is built out of increasingly complex logic circuits • That data and instructions alike need to be represented using binary digits	• How search engines select and rank results when searches are made • How to use search technologies effectively • What the impact of search technologies have been and the issues that arise by the way they function and the way they are used • How to create hyperlinks to allow users to navigate between multiple web pages • How to implement navigation to complete a functioning website		into more manageable steps • How to use user input in a block-based programming language • How to use variables in a block-based programming language • How to reflect and react to user feedback • How to use a block-based programming language to include sequencing and selection • How to evaluate the success of the programming project

Vocabulary Instruction:	Representations, symbols, storage, communication, processing, characters, coding (encoding/decoding), coding scheme, representation size or length, physical medium, binary digits, digital systems, decimal numbers, binary numbers, conversion (between number systems), representation size, units, multiples, prefixes	Algorithm, program, programming language, program translation and execution, interpreter, programming environment, input, output, variables, assignment, operators, expressions, integer and string type, execution, walk-through, selection, relational (or comparison) operators, logical (or Boolean) expressions, conditions, randomness, iteration, flags	Computer, system, device, program, software, instructions, data, hardware, processor, memory, storage, communication, input and output, architecture, operating system, logical operators (NOT, AND, OR), logical expressions, truth values (true, false), truth tables, logic gates, logic circuits	HTML, tags, formatting, image, tag, attribute, directory, render, CSS, style, formatting, head, body, attribute, search term, keywords, hyperlink, crawler, spider, index, query, ranking, connective, clause, operator, AND, OR, NOT, quote search, hyperlink, website, navigation, hyperlinks	Vector, raster, bitmap, paths, pixels, rectangle, ellipse, segment, arc, polygon, star, fill, stroke, select, move, resize, rotate, duplicate, flip, z-order, paths, operations, select, group, ungroup, align, distribute, union, difference, intersection, equidistant, object, node, freehand, path node, path segment, handles, monochrome, logo, illustration, icon, algorithms, formulae, scalable, svg (Scalable Vector Graphic)	Decomposition, mobile, app (application), properties, event-driven programming, variables, sequence, workspace, properties, ids, parameters, errors, event handler, input, checkbox
Assessment:	Knowledge check Topic test	Knowledge check Topic test	Knowledge check Topic test **Summative Assessment 1**	Knowledge check Topic test	Knowledge check Topic test	Knowledge check Topic test **Summative Assessment 2**
Key/Historical misconceptions in this unit:	Data is stored as we 'see' it not how the computer needs to use it.	Programs are written with the users responses directly coded rather than use of variables to hold data that is input. Computers will overlook mistakes in the code we produce and the way we write it.	Computers can do many things all at once. Computers can't make decisions about what to do.	Webpages are pictures that are sent ready-made rather than being defined and reconstructed. All the websites in the world appear in search engines automatically.	All drawing in a computer is done by hand.	Problems have to be solved all in one 'attempt'. We have to design computer programs to follow one set of instructions in a particular order.

Sequencing:	**We have chosen to sequence the year 8 curriculum like this because…** All three aspects of computing, (digital literacy, IT, computer science,) need to be covered in equal measure to ensure a common baseline that cannot be assured with previous education settings. These topics represent the fundamentals for the rest of the students' time at QEMS and need to be in place at an early age. There is a focus this year on the specifics of how we get computers to do the things we want. We want students to understand that solving computer-based problems is a combination of lots of smaller, easier-to-solve problems. If the smaller problems are specific enough then it is likely that these solutions already exist in some for and the only challenge is to find them and join them back up together. The technicalities of how computers work are also explored at various points in the year and the reasons why they are done in the way that they are.
Values	This scheme of work promotes the school values of Compassion, Curiosity and Courage by: *Compassion:* Students develop an awareness about how they present messages. They should understand that the way things are presented can have an impact on the perception of the message. *Curiosity:* Students begin to understand the underpinning nature that programming has on many of the topics that we encounter. They start to draw parallels between topics and understand that programming can unlock many of the wheres and whys of computer operation. *Courage:* Students need to be resilient. They understand that work is never straight-forward and that, to do well, they will need to attempt things many times. If they can develop these skills and abilities they will develop their independence and be able to solve problems on their own using the resources available to them.
National Curriculum plus:	**In addition to teaching the statutory elements of the national curriculum, we also include** Consideration throughout the year about how the topics we are covering are linked to what needs to be done in the wider world. There are many links to use of technology in jobs and lives and some of the activities, (such as building a mobile phone app,) are chosen to provide contexts with which learners will be familiar. We also explore how we utilise the Internet to catalogue and retrieve information so that learners are able to more easily find what they need using the knowledge of how it is found.