CURIOSITY COMPASSION COURAGE

Curriculum overview

Subject	GCSE Computer Science	Year group	10
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.		

CURIOSITY COMPASSION COURAGE

Threshold Concepts (TCs):	1. Programs are versions of human-created algorithms that computers understand 2. Algorithms perform differently depending on the data they are acting upon 3. Programs can still break without giving the user an error message 4. Programs that work reliably all the time have been designed to stop errors entering them in the first place 5. ? 6. ? 7. ?
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

CURIOSITY COMPASSION COURAGE

	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	**What makes a 'good' program?**					
Big picture questions:	**How quickly can an algorithm find something?**	**How quickly can an algorithm sort something?**	**How can we stop a program from breaking?**	**How do we make a 'good' program?**		
Content (Linked to TCs):	TC1 • What an algorithm is • What decomposition is • What abstraction is • How to use a systematic approach to problem solving and algorithm creation • How to explain simple algorithms in terms of their inputs, processing and outputs • How to determine the purpose of simple algorithms • How the linear search algorithm works • How the binary search algorithm works	TC2 • How the merge sort algorithm works • How the bubble sort algorithm works • That more than one algorithm can be used to solve the same problem • How to compare the efficiency of algorithms explaining how some algorithms are more efficient than others in solving the same problem • Compare and contrast linear and binary search algorithms • Compare and contrast merge sort and bubble sort algorithms	TC3 • How to write simple data validation routines • How to write simple authentication routines • What testing is for in the context of algorithms and programs • How to correct errors within algorithms and programs • What test data is and describe the following types of test data: normal (typical), boundary (extreme), erroneous data • How to select and justify the choice of suitable test data for a given problem • That there are different types of error: syntax error, logic error • How to identify and categorise errors within algorithms and programs	TC4 • How to structure programs into modular parts with clear documented interfaces • How to include authentication and data validation systems / routines within computer programs • How to write, debug and test programs to develop skills to articulate how programs work and argue using logical reasoning for the correctness of programs in solving specified problems • How to design and apply test data (normal, boundary and erroneous) to the testing of programs • How to refine programs in response to testing outcomes	Targeted revision	

Vocabulary Instruction:	Algorithm, sequence, decomposition, sub-problems, abstraction, pseudo-code, flowcharts, inputs, processing and outputs, trace tables, linear search, binary search	Merge sort, bubble sort, efficiency of algorithms, time efficiency	Validation routines, validity, authentication routines , errors, test data, normal (typical), boundary (extreme), erroneous data, syntax error, logic error	Modular, interfaces, authentication, validation, debugging, test data, normal (typical), boundary (extreme), erroneous data, syntax error, logic error	All previous	
Assessment:	Knowledge check Topic test	**Trial assessment**	Knowledge check Topic test	**Trial assessment**	**Public examination**	
Key/Historical misconceptions in this unit:	There is only one technique for searching data. There is always one best way of doing a job.	There is only one technique for sorting data. The performance of an algorithm is based on the time you physically measure to complete your task.	All errors are caused by the user breaking the program. We need to be able to test every possible user input to say that our program is reliable.	If we design our program thoroughly enough then it will be able to handle all mistakes that the user makes.	n/a	

CURIOSITY COMPASSION COURAGE

	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	**What are the various ways that computers handle data?**					
Big picture questions:	**How do computers actually do their thinking?**	**How do computers handle audio-visual data?**	**How can we reduce the size of data that we have?**			
Content (Linked to TCs):	TC5 • What Von Neumann architecture is • What the role and operation of main memory is and the major components of a central processing unit (CPU) • What the performance effects are of adjusting CPU properties • How the Fetch-Execute cycle operates	TC6 • That a bit is the fundamental unit of information • That a byte is a group of 8 bits • That quantities of bytes can be described using prefixes • How images are represented digitally in computers • How to calculate bitmap image file sizes based on the number of pixels and colour depth • How sound is represented digitally in computers • How to calculate sound file sizes based on the sampling rate and the sample resolution	TC7 • What data compression is and why it is used • How data can be compressed using Huffman coding • How to interpret and construct Huffman trees • How data can be compressed using run length encoding (RLE)	Exam technique	Targeted revision	
Vocabulary Instruction:	Hardware, software, processor(s), memory, I/O devices, applications, security, Von Neumann architecture, arithmetic logic unit (ALU), control	Data, instructions, bit pattern, information, bit, byte, prefix, kilo, mega, giga, tera, analogue, digital, sampling,	Compression, Huffman coding, trees, uncompressed, run length encoding (RLE), frequency	All previous	All previous	

	unit, clock, bus, cache, registers, clock speed, processor cores, cache size, cache type, Fetch-Execute cycle, fetch, decode, execute	amplitude, sampling rate, sample resolution,				
Assessment:	Knowledge check Topic test	**Trial assessment**	Knowledge check Topic test	**Trial assessment**	**Public examination**	
Key/Historical misconceptions in this unit:	Computers pause when they aren't being used. Computers work as fast as the CPU can go.	Real world data can be represented exactly using binary values. Higher quality data is always better and should be the default for all uses.	Data compression always lets us get back to the data we started with.	n/a	n/a	

Sequencing:	**We have chosen to sequence the year 10 curriculum like this because…** Again, the year is split between learning about computer programming and how computers work. This time the focus is on the application of the knowledge accrued during Year 10. Programming wise, the concept of an algorithm is defined more fully. The learners then progress on to thinking about the utility of algorithms based on the jobs that they do. They have to explore both searching and sorting algorithms and then this is reviewed through comparison in performance. This brings to light that different algorithms are better at their jobs than others, but this is dependent on the starting conditions of the data they are acting on. The remainder of the year is given over to the extended practise of designing robust programs they achieve given gaols for different scenarios. This is excellent preparation for paper 1 questions in the exam. On the theory side, key concepts from Year 10 are reviewed again here. Once the core concepts are re-established the contexts are taken further and in greater depth. This serves an additional purpose of exploring previous content from a different perspective and also allows learners to see the links between topics. Again, this helps to develop a deeper understanding in preparation for their exams.
Values	**This scheme of work promotes the school values of Compassion, Curiosity and Courage by:** *Compassion:* Users of computers are creating things for people to use and read. They should therefore do this in a way that considers the impact of their actions and use this as a moderating voice. *Curiosity:* Students apply their learning to many practical examples. They are given problems to solve and use their prior learning to help arrive at new solutions. *Courage:* The nature of the work and the activities they have to complete develop the courage of students. They need to learn how to solve more complex problems by breaking them down into lots of smaller, easier-to-achieve tasks.
National Curriculum plus:	**In addition to teaching the statutory elements of the national curriculum, we also include** Programming is a big draw for students and there is an extended focus on trying to solve more than trivial problems. Many electing to do the subject see a benefit of learning how to program and it is important that they appreciate how their core skills are still applicable to even the most complex problems that can be solved. Whilst there is no formal requirement for this we consider this to be an important part of the learners' development in the subject.