

Year 11 Curriculum Map : Computing

	Autumn	Spring		Summer		
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.					
Unit	Algorithms and Problem solving	Computer systems	Ethical Legal and environmental impacts of technology	Exam Revision	Exam revision	/
Key Learning Outcomes	1. Computational thinking 2. Computational thinking 3. Searching and sorting algorithms. 4. Searching and sorting algorithms. Knowledge audit test. 5. Reading algorithms 6. Reading algorithms 7. Writing algorithms 8. Writing algorithms 9. Writing algorithms Knowledge audit test. 10. Trace tables 11. Trace tables 12. Exam question practice. 13. Exam question practice. 14. Assessment	1. D.I.R.T lesson 2. Hardware and software 3. Internal hardware 4. Internal hardware 5. Application software Knowledge audit test. 6. System software 7. Operating System 8. Secondary storage 9. The CPU 10. Von Neumann Architecture Knowledge audit test. 11. Logic gates 12. Logic gates 13. Embedded Systems 14. Exam question practice. 15. Assessment	1. D.I.R.T lesson 2. Ethics – Business 3. Ethics – The internet (Digital divide) 4. Legal – Privacy 5. Legal – Copyright and patents 6. Legal – Cyber crime 7. Environment – Cycle 8. Cloud storage 9. Driverless cars (AI) 10. Exam question practice. 11. Exam question practice. 12. Exam question practice. 13. Assessment & DIRT lesson	Revision dependent on areas for improvement.	Revision dependent on areas for improvement.	/
Prior knowledge	KS3: Pupils should be taught to design, use and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems. Pupils should be taught to 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. Pupils should be taught to understand simple Boolean logic [for example, AND, OR and NOT] and some of its uses in circuits and programming. Pupils should be taught to 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]. Pupils should be taught to understand the hardware and software components that make up computer systems, and how they communicate with one another and with other systems.					

	<p>Pupils should 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.</p> <p>Pupils should 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.</p>		
CEIAG Specific careers links	<p>Link to careers routes: Software developers, Cyber security, CPU engineer, Computer Scientist, Environmental scientist, IT Lawyers.</p> <p>Topics to be delivered by employers: Laws in Computing, CPU design, Software development, program design.</p> <p>Link to personal skills: problem solving, resilience, creativity, Numeracy, Logical thinking.</p>		
RRSA	<p>Article 16: Right to privacy</p> <p>Article 28: Right to education</p> <p>Article 29: Goals of education</p>	<p>Article 16: Right to privacy</p> <p>Article 28: Right to education</p> <p>Article 29: Goals of education</p>	<p>Article 16: Right to privacy</p> <p>Article 28: Right to education</p> <p>Article 29: Goals of education</p>
Cross curricular links	<p>Maths – Key programming terminology and concepts, programming operators, Collecting, and manipulating data.</p> <p>Science – Key programming terminology, uses of variables. Environmental impacts of technology, finite resources.</p> <p>DT – Designing for a purpose.</p>		
Useful websites/videos	<p>Learn python - https://www.w3schools.com/python/default.asp</p> <p>How computer memory works - https://www.youtube.com/watch?v=p3q5zWCw8J4&t=207s</p> <p>Ethical dilemma of driverless cars - https://www.youtube.com/watch?v=ixloDYVfKA0&t=182s</p>		
Wider Reading	<p>Computational fairy tales - https://anyflip.com/xvkk/deuw/basic</p> <p>The Road to Conscious Machines: The Story of AI by Michael Wooldridge; Pelican, 2021</p>		
Literacy Programme	<ul style="list-style-type: none"> • Decode it NOW, Review it now, Glossary pages for key terminology. • Sentence Starters • Guided practice/model answers • Immersive reader function in office 365. • Reciprocal reading tasks. 		
Independent Learning Tasks	<p>Knowledge organiser recall questions. Students are asked to complete these to prepare for knowledge audits and assessments. Know it, Think it, Grasp it questions.</p> <p>OneNote catch up tasks. Flipped learning reading tasks.</p>		