

## Curriculum Intent Statement for **Computer Science**

At Chase Terrace Academy we aspire for all of our students to achieve greater things than they ever thought possible.

We pride ourselves on being a warm and welcoming school that places community at the heart of everything we do. Our ambitious curriculum is enriching and inclusive, providing challenge and breadth for all. This empowers our students to become compassionate, confident and creative individuals who are resilient, respectful and equipped with a desire to take up a fulfilling role in society and the wider world.

In Computer Science we aspire to enrich students with a varied and deep understanding of computing developments, concepts and the impact of technology on our society and environment. Students learn a diverse range of skills such as programming in a range of languages and also study the theory behind the science of computing, the Internet and the ever growing importance of our personal security and privacy. Ultimately, we aim to give students the knowledge and experience they need to study Computing to degree level, to use technology in their day to day lives or careers and to manipulate technology and tools to compliment almost any future study or job.

### Year 12 and 13 Curriculum Implementation Plan (Computer Science)

Computer Science				
Knowledge and Skills – Students will study...	Reading, Literacy and Numeracy	Formative Assessment	Summative Assessment	Link to GCSE Content
Year 12:  1.4 Data types, data structures and algorithms - How data is represented and stored within different structures. Different algorithms that can be applied to these structures  1.1 The characteristics of contemporary processors, input, output and storage devices -	Reading: <ul style="list-style-type: none"> <li>Regular use of extended out of class reading and research tasks focused on particular topic areas</li> <li>Extensive research and reading required for the coursework element of the course</li> <li>Research and online reading and extracts</li> </ul>	Structured revision programme  One to one feedback and support due to small group sizes  Regular self assessment during exam question practise  Regular opportunities to revisit previous tasks and improve based on feedback	40 on going programming challenge tasks  Six end of unit assessments  Two formal mock exams  Extended coursework task – worth 25% of the final grade.  Final A Level Examinations	All GCSE content is built upon directly throughout the A-Level.

<p>Components of a computer and their uses</p> <p>1.3 Exchanging data - How data is exchanged between different systems</p> <p>1.5 Legal, moral, cultural and ethical issues - The individual moral, social, ethical and cultural opportunities and risks of digital technology. Legislation surrounding the use of computers and ethical issues that can or may in the future arise from the use of computers</p> <p>1.2 Software and software development - Types of software and the different methodologies used to develop software</p> <p>2.2 Problem solving and programming How computers can be used to solve problems and programs can be written to solve them (Learners will benefit from being able to program in a</p>	<p>Literacy:</p> <ul style="list-style-type: none"> <li>• Extended written responses across units</li> <li>• Extended written task during coursework write up</li> <li>• In depth research and referencing of sources</li> <li>• Use of spelling and grammar tools</li> <li>• Regular review of in class work focussed on level of written response</li> <li>• Modelling of appropriate level of written response</li> </ul>	<p>Verbal feedback on an individual basis</p> <p>Whole class feedback</p> <p>Microsoft Forms based quizzes and quick tests with visual feedback</p>		
	<p>Numeracy:</p> <ul style="list-style-type: none"> <li>• Algebra – variables and data types</li> <li>• Logic and decision making</li> <li>• Circuit design</li> <li>• Refining logical problems</li> <li>• Refining and applying algebraic simplification rules</li> <li>• Number systems including floating</li> </ul>			

<p>procedural/imperative language and object oriented language.)</p> <p>2.3 Algorithms - The use of algorithms to describe problems and standard algorithms</p> <p>2.1 Elements of computational thinking - Understand what is meant by computational thinking</p>	<p>point representation</p> <ul style="list-style-type: none"><li>• AND, OR, NOT, XOR</li><li>• Conditional statements</li><li>• Logic diagrams</li><li>• Truth tables</li><li>• Number conversions</li><li>• Number systems</li><li>• Encryption</li><li>• Programming skills</li></ul>			
--	--	--	--	--