

## 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 11 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
<ul style="list-style-type: none"> <li>the use of variables, constants, operators, inputs, outputs and assignments</li> <li>the use of the three basic programming constructs used to control the flow of a program:               <ul style="list-style-type: none"> <li>sequence</li> <li>selection</li> <li>iteration (count and condition controlled loops)</li> </ul> </li> <li>the use of basic string manipulation</li> <li>the use of basic file handling operations:</li> </ul>	Reading: <ul style="list-style-type: none"> <li>Regular use of on screen sources of information</li> <li>Research and online reading and extracts</li> </ul> Literacy: <ul style="list-style-type: none"> <li>Extended written responses across units</li> <li>In depth research and referencing of sources</li> <li>Use of spelling and grammar tools</li> </ul>	Structured revision programme  On screen reviews of student work  Regular self assessment at key stages against level descriptors  Regular opportunities to revisit previous tasks and improve based on feedback	One extended programming task  One end of unit assessment  Two formal mock exams  GCSE Examinations	1.1 – Systems Architecture 1.2 – Memory 1.3 – Storage 1.4 – Networks 1.5 – Topologies and protocols 1.6 – System Security 1.7 Systems Software 1.8 – Law, Ethics, Morals 2.1 – Algorithms 2.2 – Programming techniques 2.3 – Producing robust programs 2.4 – Computational Logic

<ul style="list-style-type: none"> <li><input type="radio"/> open</li> <li><input type="radio"/> read</li> <li><input type="radio"/> write</li> <li><input type="radio"/> close</li> <li>• the use of records to store data</li> <li>• the use of SQL to search for data</li> <li>• the use of arrays (or equivalent) when solving problems, including both one and two dimensional arrays</li> <li>• how to use sub programs (functions and procedures) to produce structured code</li> <li>• the use of data types:           <ul style="list-style-type: none"> <li><input type="radio"/> integer</li> <li><input type="radio"/> real</li> <li><input type="radio"/> Boolean</li> <li><input type="radio"/> character and string</li> <li><input type="radio"/> casting</li> </ul> </li> <li>• the common arithmetic operators</li> <li>• the common Boolean operators.</li>   <li>• defensive design considerations:           <ul style="list-style-type: none"> <li><input type="radio"/> input sanitisation/validation</li> <li><input type="radio"/> planning for contingencies</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <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>Numeracy:</p> <ul style="list-style-type: none"> <li>• Algebra – variables and data types</li> <li>• Logic and decision making</li> <li>• AND, OR, NOT</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>	<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>2.5 – Translators and facilities of languages</p> <p>2.6 – Data Representation</p>
--	--	---	--	---

<ul style="list-style-type: none"><li><input type="radio"/> anticipating misuse</li><li><input type="radio"/> authentication</li><li>• maintainability:</li><li><input type="radio"/> comments</li><li><input type="radio"/> indentation</li><li>• the purpose of testing</li><li>• types of testing:</li><li><input type="radio"/> iterative</li><li><input type="radio"/> final/terminal</li><li>• how to identify syntax and logic errors</li><li>• selecting and using suitable test data.</li></ul>				
--	--	--	--	--