



Year Group	Year 8					
Subject Intent	<p>Computing 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.</p> <p>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 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 learn a range of content.</p>					
Subject Implementation	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
Knowledge	<ul style="list-style-type: none"> ★ Design, use and evaluate computational abstractions that model the state and behaviour of real-world problems and physical systems ★ 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 		<ul style="list-style-type: none"> ★ create, reuse, revise and repurpose digital artefacts for a given audience, with attention to trustworthiness, design and usability ★ 		<ul style="list-style-type: none"> ★ Understand several key algorithms that reflect computational thinking [for example, ones for sorting and searching]; use logical reasoning to compare the utility of alternative algorithms for the same problem ★ Use 2 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 	
Skills	<p><u>Unit 1 - Spreadsheets</u></p> <p>→ Identify columns, rows, cells, and cell references in</p>		<p><u>Unit 2 - Developing for the web</u></p> <p>→ Describe what HTML is</p>		<p><u>Unit 3 - Introduction to Python programming - 2</u></p> <p>→ Use multi-branch selection (if, elif, else</p>	



	<p>spreadsheet software</p> <ul style="list-style-type: none"> → Use formatting techniques in a spreadsheet → Use basic formulas with cell references to perform calculations in a spreadsheet (+, -, *, /) → Use the autofill tool to replicate cell data → Explain the difference between data and information → Explain the difference between primary and secondary sources of data → Collect data → Analyse data → Create appropriate charts in a spreadsheet → Use the functions SUM, COUNTA, MAX, and MIN in a spreadsheet → Analyse data → Use a spreadsheet to sort and filter data → Use the functions AVERAGE, COUNTIF, and IF in a spreadsheet → Use conditional formatting in a spreadsheet → Apply all of the spreadsheet skills covered in this unit 	<ul style="list-style-type: none"> → Use HTML to structure static web pages → Modify HTML tags using inline styling to improve the appearance of web pages → Display images within a web page → Apply HTML tags to construct a web page structure from a provided design → Describe what CSS is → Use CSS to style static web pages → Assess the benefits of using CSS to style pages instead of in-line formatting → Describe what a search engine is → Explain how search engines 'crawl' through the World Wide Web and how they select and rank results → Analyse how search engines select and rank results when searches are made → Use search technologies effectively → Discuss the impact of search technologies and the issues that arise by the way they function and the way they are used → Create hyperlinks to allow users to navigate between multiple web pages → Implement navigation to complete a functioning website 	<p>statements) to control the flow of program execution</p> <ul style="list-style-type: none"> → Describe how iteration (while statements) controls the flow of program execution → Use iteration (while loops) to control the flow of program execution → Use variables as counters in iterative programs → Combine iteration and selection to control the flow of program execution → Use Boolean variables as flags
--	---	---	---



Subject Impact	<p>The unit takes learners from having very little knowledge of spreadsheets to being able to confidently model data with a spreadsheet. The unit uses engaging activities to progress learners from using basic formulas to writing their own COUNTIF statements.</p>	<p>In this unit, learners will explore the technologies that make up the internet and World Wide Web. Starting with an exploration of the building blocks of the World Wide Web, HTML, and CSS, learners will investigate how websites are catalogued and organised for effective retrieval using search engines. By the end of the unit, learners will have a functioning website.</p>	<p>The Y7 - Introduction to Python Programming - 1 unit is a prerequisite for this unit.</p> <p>The unit of work continues to form a journey that starts with simple programs involving input and output, and gradually moves on through arithmetic operations, randomness, selection, and now iteration. Emphasis is placed on tackling common misconceptions and elucidating the mechanics of program execution. A range of pedagogical tools is employed throughout the unit, with the most prominent being pair programming, live coding, and worked examples.</p>
Assessment	<ul style="list-style-type: none"> ★ 2 Formative Assessments, <i>for e.g. Class work, Homework, Presentation, Short Recall Test, Practical Project, Quiz</i> ★ 1 Summative assessment, <i>for e.g. Internal School Examinations, End of Topic/Unit Tests</i> <p>(Summative assessment covers content taught in Autumn Term)</p>	<ul style="list-style-type: none"> ★ 2 Formative Assessments, <i>for e.g. Class work, Homework, Presentation, Short Recall Test, Practical Project, Quiz</i> ★ 1 Summative assessment, <i>for e.g. Internal School Examinations, End of Topic/Unit Tests</i> <p>(Summative assessment covers content taught in Autumn and Spring Term)</p>	<ul style="list-style-type: none"> ★ 2 Formative Assessments, <i>for e.g. Class work, Homework, Presentation, Short Recall Test, Practical Project, Quiz</i> ★ 1 Summative assessment, <i>for e.g. Internal School Examinations, End of Topic/Unit Tests</i> <p>(Summative assessment covers content taught in the curriculum over the whole academic year)</p>