

Computing Year 8	Curriculum intent: The year 8 curriculum builds on the achievements of year 7 and consolidates that understanding through the use of interleaving the key concepts listed below and tasks to scaffold the students’ learning experiences. As in year 7, new concepts and techniques are acquired through a combination of teacher led tasks, class discussions and practical activities. These are designed to increase the student understanding of the concepts, allowing them to apply their new found understanding in a variety of different scenarios related to the concepts being taught in an engaging way. This approach not only allows for co-ordinated efforts of students to be encouraged but also makes use of numerous open ended tasks that allow all students to stretch their understanding of these concepts as far as they are able.																				
Topic	Networks				Web Development				Computational Thinking				Object Orientated Programming				Combining digital skills				Cryptography
Interleaving	Key knowledge from previously studied topics				Key knowledge from previously studied topics				Key knowledge from previously studied topics				Key knowledge from previously studied topics				Key knowledge from previously studied topics				Key knowledge from previously studied topics
Knowledge	Students will learn the hardware and software components that make up computer networks, and how computers communicate with one another and with other systems.				Students will learn about creating basic content for the web including HTML, CSS and JS. Students will learn about the structure of web pages, web sites and interactive content with a firm basis in making content accessible and easy to use. Students will also learn about the difference between well-formed and valid markup and how this can impact the end user and the applications they use.				Students will study common algorithms and how they are selected and applied to given contexts both in code and offline activities.				Students will learn how to solve a variety of more complex computational problems and about the object-orientated programming paradigm. Students will understand why the object orientated paradigm is important and how it relates to modelling objects in the real world. Students will be able to explain when it is appropriate to use Object Orientation and the advantages of doing so for the programmer and others involved in the development process.				Students will use a range of different applications to create and combine digital artefacts, making use of the most appropriate software for a varying set of related tasks.				Students will learn about a range of cryptographic methods and how to encode and decode data using them. Students will understand the importance of cryptography and the role it has played in the past as well as how it is used in the current day.
Understanding	Students will learn about different types of networks and explain the components involved in them. Students will examine how data travels across a network, learn about the different standard protocols and how networks provide both advantages and disadvantages to those using them.				Students will be able to write HTML code to create a simple web page and progress into applying styles and scripts to enhance a page. Students will also understand how humans and computers interact and evaluate how the principles of usability apply in a range of scenarios and how these ultimately affect engagement with their content, evaluating their work against these criteria throughout.				Students will learn about common searching and sorting algorithms and explain how these work through a variety of medium including presentations and demonstrations. Students will work through the steps of given algorithms and apply these to different sets and types of data, evaluating their suitability and whether an alternative approach may have been appropriate.				Students will adapt, test and generate programs in a text-based language with well factored elements and describe these elements using documentation. Students will generate code from given pseudocode and identify where traditional scripts may be improved through refactoring and redesign, implementing these solutions as part of their learning. Students will present their solutions as either pseudo code, diagrams or actual code depending on the task.				Students will select the most appropriate software to produce digital content and combine output from multiple applications (for example, mail-merging documents from a given data source, leaflets and flyers, etc.) for a set of fictional business requirements.				Students will encrypt and decrypt data using a range of methods and produce a report focussing on an element of cryptography and how it worked and impacted those who used it.
Skills	Computational Thinking	Computational Thinking	Computational Thinking	Computational Thinking	Computational Thinking	Computational Thinking	Computational Thinking	Computational Thinking	Problem Solving	Analysis, Evaluation and Implementation	Technical Vocabulary	Computational Thinking	Problem Solving	Analysis, Evaluation and Implementation	Technical Vocabulary	Computational Thinking	Problem Solving	Analysis, Evaluation and Implementation	Technical Vocabulary		
Assessment	Pupils will sit an end of unit test as their final assessment.				Pupils will complete a series of short activities throughout the unit and present their finished code and site for their final assessment.				Pupils will write and run a program and submit the code and screenshots of the program running in an Assessment Portfolio.				Pupils will create a program as their final submission task for this topic alongside documentation and description of the code.				Students will present the artefacts they have generated and explanations as to why the artefacts have been generated using that particular piece of software.				Pupils will present their report, including the disadvantages and advantages of the selected cryptographic method.