Marshland HIGH SCHOOL

Marshland Curriculum Overview Computing

Curriculum Intent
Our curriculum is designed to fully satisfy the requirements of the national curriculum and to prepare pupils for the next stage of their academic or professional career. We prepare pupils to be good digital citizens, develop their confidence using technology in a responsible manner (digital literacy) and support them to keep safe online. Our curriculum enables students to apply computational thinking, enabling them to engage in jobs in the future that will help influence and change the world. As students progress through their academic journey they develop the concepts of computer science including abstraction and decomposition, applying logic, use of algorithms (use of sequence, selection and iteration) and presenting data in a range of ways.

How does our curriculum build upon previous learning?	
KS3	Students are introduced to the basics of Computing at KS2. Our KS3 curriculum builds on prior learning to develop knowledge from KS2 including designing, writing and debugging programs. Sequence, selection and iterations are built upon and basic skills recapped, delivered and extended in KS3 and KS4. At KS3 students become much more familiar with the vocabulary of Computer Science. KS2 introduces block-based programming which is developed further in Y7 with more complex block-based, which develops further into text-based programming. In Y8 HTML is introduced allowing students to develop accuracy in coding, building upon Y7 text-based coding to support them in Y9 (and beyond).
KS4	The KS4 curriculum continues to build depth of understanding for students. KS3 introduces students to Binary values, and at KS4 they extend their knowledge by learning how text, images and sound are represented by binary. KS3 knowledge directly supports their understanding of binary at KS4.

What do students *do* with this knowledge or these skills?	How do we help students secure this knowledge in long-term memory?
Computing knowledge and skills feed directly into other subjects, as well as develop future work and life skills, using computers efficiently and safely for desktop publishing, spreadsheets and presentation software. Knowledge acquired in coding each year is used to support more advanced coding as students progress through their schooling.	Significant elements of the Computing curriculum are based on prior learning therefore, it is essential that knowledge becomes secure in the long-term memory. Within lessons a range of techniques are deployed including: ● Knowledge recall quizzes ● Questions based from Knowledge Organiser (KS3) ● Use of the 50 facts questions (KS4) ● Use of the Marshland Computing Revision Book ● Targeted questions in class ● Use of spaced practice to aid securing knowledge

Marshland HIGH SCHOOL

How does our curriculum align to the national curriculum?
The KS3 curriculum has been carefully planned to cover all aspects of the national curriculum, which in turn underpins our own.

How do we check student understanding and monitor progress?
KS3 students create an e-portfolio with examples of their work completed in Computing lessons. They explain their work demonstrating their understanding of the various topics. Teachers look at the e-portfolios as students progress, correcting any misunderstandings that may develop. Every lesson contains a knowledge recall activity to enable the teacher to check understanding and progress. Questioning is routine in lessons to check and develop understanding further to monitor/inform the teacher if any area needs recapping. KS4 assessments take place which are analysed question by question to check understanding. Questioning in class monitors the understanding of current topics.

Curriculum sequencing

Year	Autumn	Spring	Summer
7	**Introduction** Prepares pupils to use the technology that will support their learning in Year 7 and beyond. Includes password security, email, Satchel One, google drive, input and output devices, how to print in school, how to search the internet and introduction to copyright and plagiarism. **E-Safety** How to keep safe online and when using a computer includes: ● how cyberbullying affects an individual and what to do if someone is being cyberbullied ● social media - the advantages and disadvantages of such technologies ● netiquette and leaving a digital footprint	**Spreadsheets** The basics of spreadsheets to analyse data, including basic formulas, functions and how to display information in graphs. **Cryptology and Flowal** How data is secured on a computer using cryptography. Pupils also look at how algorithms can be represented using flow charts.	**Scratch (Block based programming)** Computational thinking and block programming to create a game. Pupils learn about Cartesian coordinates, IF statements, forever loops and then move onto variables. Pupils learn about sequence, selection and iteration. **Introduction to text-based programming** Pupils' first introduction to text-based programming, learning about the syntax of python and developing programming skills that require accuracy and problem solving.

Curriculum sequencing			
Year	**Autumn**	**Spring**	**Summer**
8	**HTML** Learn to program for the web by using HTML to create a website including tags and how to structure a webpage. Skills include formatting text, use of colour (including RGB colour chart) tables and hyperlinks. **Animation and Video Editing** How animations are created using frames and layers including frame rate. This develops into using transitions to create a video for a target audience	**Web Authoring** Pupils move from text-based programming of a website to using an editor to develop a webpage. They learn more advanced features and focus on audience needs and purpose alongside accessibility features to make the website more usable **Python** Develop Python knowledge to further understand variables and data types. Build knowledge of iteration (loops) to make more efficient programs.	**Spreadsheets** Develop spreadsheet skills to ask 'what if' questions to a model. Moves onto naming cell ranges, sorting data, vlookups and other more advanced skills such as writing macros and protecting cells. **Database (Flat File)** It is important pupils understand how data is stored and this unit focuses on how to create and use a database to search and find the information that they require.
9	**Bitmap Editing** How to edit images for a particular audience. Pupils learn that images are not necessarily real as they may have been edited. Focuses on the types of files that can be used for images and the differences between them including file compression. **Coding - Data Representation** How a computer stores text, images and sound as binary values. How data is stored (compressed) using different methods. How analogue sound files are sampled into digital sound files.	**Database (Relational)** Develops pupils' understanding of databases and the use of relational databases to show how information can be retrieved from more than one related table. **Python** Knowledge of python is further developed by learning how to read and write to txt files. Develop skills in handling data in Python and being able to question the data to find answers. How to search and sort data effectively using Linear and Binary searches and bubble and merge sorts. How computers hold data in arrays	**Integrated Project** The final double unit brings together many aspects of computing where pupils develop an understanding of the system life cycle by planning, researching, designing, implementing and reviewing a large project. Using tools from KS3 to help them undertake this task such as the use of spreadsheets and databases to create financial models and store information. **Quiz** KS3 is completed by the creation of a quiz using facts and knowledge from content throughout KS3. Students use 'Scratch', developing a greater understanding of variables and If statements.

Curriculum sequencing			
Year	**Autumn**	**Spring**	**Summer**
10	**Programming Fundamentals** Inputs, outputs, error messages, if, elif and else, string manipulation, loops, validation **Units, Data Storage** Binary values, binary shifts, logic and truth tables, ASCII and Unicode character sets, file size and encryption, compression	**Algorithms** Flow Diagrams, pseudocode, problem-solving, trace tables **Computer Components** Input, process and output, components, cache, CPU, CU and ALU, secondary storage, Cloud computing, embedded Systems **Operating Systems** Purpose and functions, utility software, high/low level languages, assembly language	**Networks** Lan and Wan, connection media **Security** Threats, DOS, Brute Force and SQL injection, penetration testing, access levels and encryption, physical security, authentication and firewalls, types of risk, prevention **Impact of technology** Emerging trends, issues and threats, legal issues, copyright and patents, environmental, ethical issues **The internet** Protocols and Networks **Programming** Reading and writing to files, python lists, two dimensional arrays, writing algorithms, Little Man computer, SQL **Searching and sorting** Bubble, merge, insertion sort, linear and binary

Curriculum sequencing			
Year	**Autumn**	**Spring**	**Summer**
11	Recap all areas of the GCSE specification to remind students of revision focus areas. ● System architecture ● Memory and storage ● Units ● Data storage ● Networks and topologies ● The internet, wired and wireless, encryption ● Network security and layers ● Operating systems ● Utility software ● Computational thinking ● Designing, creating and refining algorithms ● String manipulation ● SQL ● Functions and procedures ● Integrated development environment ● High and low-level languages ● Exam style questions ● Searching and sorting - linear, binary, merge, bubble and insertion ● Logic and truth tables	Preparation for Computer Science Exam by relating theory to exam style questions. ● System architecture ● Primary storage ● Secondary storage ● Units ● Compression ● Wired and wireless protocols ● Identifying and preventing vulnerabilities ● Ethical, legal and cultural issues	Preparation for Computer Science Exam ● Identifying and fixing errors in code ● Pseudocode and flowchart revision
Rationale for this sequencing	The sequence of work is carefully planned to ensure that Y7 students have the basic skills required to effectively learn and acquire knowledge. In addition to this, students are instructed in the safe use of computers so that they understand e-safety right from the start. As confidence, knowledge and skills develop, Y7 students progress from following instructions to working on units that require more understanding and application of skills, such as the coding towards the end of the year. In Y8 students develop accuracy when coding HTML. This progresses into creating and using digital artefacts. Sequencing allows students to use HTML tags learned in module one, alongside elements created in module two to develop a website using software in module three leading to the Web Design unit. The Y9 integrated project brings the curriculum together as students apply skills and knowledge acquired throughout to complete it. Students use the project lifecycle of initiation, plan, execute and review. Whilst guided on tasks, they independently undertake the project using the skills and knowledge taught.		

How does our curriculum prepare students for the transition to post-16 pathways?
Programming taught in computing directly supports skills required for A Level Computing. The primary programming language is python which is used as one of the programmes at A Level.