Computer Science

Subject Leader: Mrs S Thomas **Email:** sthomas1@taptonschool.co.uk

Curriculum Intent: To give all our students the opportunity to learn 'powerful knowledge' through a curriculum with computational thinking at its core. Our curriculum is designed with a balance of the three strands of Computer Science, Information Technology and Digital Literacy with the aim of enabling all our students to be active participants in an increasingly digital society.

	Core Knowledge	Procedural Knowledge
Autumn	**Topics:** Computer systems 2: The CPU Computational thinking: With Bebras - 3 Computer systems 3 : Communicating and exchanging data protocols and networks	**Students will:** Describe how instructions are stored and executed within a computer system. Describe the basic components of the CPU. Describe the roles and purpose of each component of the CPU in computation. Apply decomposition, abstraction, and algorithmic thinking to help solve problems. Define computer networks, the internet and 'protocols'. Identify the key hardware used in networks. Understand different types of networks and topologies and identify advantages and disadvantages of each. Explain how data travels between computers using protocols and packets. Explain the difference between the internet, its services, and the World Wide Web. List examples of the hardware required to network computing devices.
Spring	**Topics:** Data representation 2: Binary, character sets and images Algorithms the essentials: Trace tables, flow charts, searching and sorting algorithms Physical computing 2: micro:bits with Python. Programming with Python 2: Key concepts (sequence, selection and iteration), functions and procedures, Arrays, Lists and Numbers.	**Students will:** Carry out operations on binary numbers (binary addition, conversion between binary and decimal, convert between hexadecimal, denary and binary) Able to look up values in character sets and convert between binary numbers and characters. Demonstrate how pixels are used to represent images. Calculate bit depth in an image. demonstrate sound sampling frequency and its impact. Identify the inputs, processes, and outputs for a problem. Articulate the difference between an algorithm and a computer program, Create, interpret, correct, and complete algorithms using: Flowcharts. Use trace tables to walk through code. Students can articulate/describe 2 key search algorithms, linear and binary, and 2 key sort algorithms , bubble and insertion. Make use of data structures. Design & develop modular programs that use procedures and functions using Python. Apply debugging using IDEs.

Summer	**Topics:** The future of jobs 2. Data modelling: Modelling, analysing data with spreadsheets: Machines and me: AI and machine learning. The IOT and smart devices Computer Systems 4: Boolean Logic & Circuits. Using IOT: programming and gathering data with IOT Data science: handling data, patterns and trends and algorithms for data Data in action: data and health, data and the environment	**Students will:** Identify the potential paths into roles in the technology sector. Define IoT and identify devices use in everyday life. Understand how Smart home, cities and agriculture use IoT to improve our world. Suggest ways in which IoT might fail and security issues related to internet connectivity. Use simple Boolean logic (for example, AND, OR and NOT) to set conditions, create circuits. Be able to write truth tables for common logic gates. Use a textual language, to solve a problem; make appropriate use of data structures design and develop modular programs that use procedures and functions. Analyse problems in computational terms through practical experience of solving such problems, including designing, writing and debugging programs. Undertake creative projects that involve selecting, using, and combining multiple applications, across a range of devices, including collecting and analysing data and meeting the needs of known users. Create, reuse, revise and repurpose digital artefacts for a given audience, with attention to trustworthiness, design and usability. Program an IOT device to collect data. Import data in various formats, use formula, formatting and filtering in spreadsheets and tableau. Be able to collect data, apply filters and visualisation tools to analyse data using spreadsheets. Use charts and other visualisation techniques to demonstrate patterns and trends in spreadsheets. Identify the difference between correlation and causation.

Homework: Students will have two pieces of homework per half term. Homework will comprise a combined terminology revision exercise and quiz each half term to aid students' development of the extensive technical language use in Computer Science.

Assessment:
The use of progress tasks in lessons.
Summative end of topic multiple choice quizzes.

Autumn Assessment:
Students will be assessed on Topics from Year 7, Year 8, and the Autumn Term of Year 9.
The assessment will be online and last for 60 minutes. The format will be a mixture of multi-choice questions, text-based questions and written exam-style questions. Students will complete the assessment in their Computer Science class. A revision guide will be available on Class Charts.

Spring Assessment:
Students will be assessed on Topics from Year 7, Year 8 and Autumn, Spring and Summer Term in Year 9.
The assessment will be online and last for 60 minutes. The format will be a mixture of multi-choice questions, text-based questions and written exam-style questions. Students will complete the assessment in their Computer Science class. A revision guide will be available on Class Charts.

Links to Personal Development: Enabling Students to recognise online risks to their own wellbeing.
Students to recognise the dangers of inappropriate use of mobile technology and social media.
Build students confidence, resilience, understanding of ethics, cultural capital and knowledge.
Prepare learners for future success in education, employment and training, so that they can keep themselves mentally healthy and be economically successful.

Promote inclusion: Computer Science opportunities are for everyone
How is my knowledge developed further at GCSE? Computer Science will encourage you to: understand and apply the fundamental principles and concepts of Computer Science, including abstraction, decomposition, logic, algorithms, and data representation. Analyse problems in computational terms through practical experience of solving such problems, including designing, writing, and debugging programs. Think creatively, innovatively, analytically, logically and critically. Understand the components that make up digital systems, and how they communicate with one another and with other systems and understand the impacts of digital technology to the individual and to wider society.