FRAMEWORK FOR LEARNING

CHS South

CREATIVE An education where imagination, curiosity and resilience enable us to ignite our learning.

HAPPY A shared belief that optimism, empathy and responsibility are the foundations for a respectful, safe and inclusive community.

SUCCESSFUL Individuals who are ready to learn, practise being reflective, and are motivated to become champions.

SUBJECT

COMPUTER SCIENCE

INTENT

Studying Computer Science will help develop problem-solving, critical thinking and analytical skills. Computer Science is found in nearly all jobs and careers. Studying Computing will provide students with a versatile foundation for many different careers and allows students to develop interchangeable and transferable skills inside and outside of IT. Our students are now living in a digital age where more of their lives become intertwined with digital technologies. It is important that students understand this technology and are able to use it effectively. In Computer Science, students will develop knowledge and understanding of key computing topics that will prepare them for their future studies in Computing. They will:

Key Stage 4:

1. Develop their capability, creativity and knowledge in computer science, digital media, and information technology.
2. Develop and apply their analytic, problem-solving, design, and computational thinking skills.
3. Understand how changes in technology affect safety, including new ways to protect their online privacy and identity, and how to report a range of concerns.

YEAR GROUP	YEAR 10					
RATIONAL / NARRATIVE	The GCSE course has a mixture of theory topics (Paper 1 – Computer Systems) and practical programming skills (Paper 2 – Computational Thinking, Algorithms and Programming). Within the course students must be given the opportunity to undertake a programming task or tasks during their course of study. In year 10 students will study a mixture of the content from both papers to help them develop the appropriate knowledge and skills. For paper 1 they will be examining the systems architecture, memory, storage, and system software which allow them to gain an understanding of how computers work. Also in Year 10 they will explore algorithms and programming skills to help develop their knowledge and understanding of a high-level programming language. **Year 10 Computer Science** **1.1 Systems architecture** – students will learn about the components of the CPU and their purpose during the FDE cycle. **1.2 Memory and storage** – students will learn about the difference between memory and storage and how data is stored in a computer system. **1.5 System software** – students will learn about the functions and features of the different system software and their role in the computer system. **2.1 Algorithms**- students will learn about the principles of computational thinking and be able to design, create and refine algorithms for a specific purpose. They will also learn about the different types of searching and sorting algorithms. **2.2 Programming fundamentals** – students will develop a range of programming techniques and skills using a high-level programming language. **2.4 – Boolean logic**- students will learn about the different types of logic operators and be able to apply these to solve problems					
TERM	AUTUMN 1	AUTUMN 2	SPRING 1	SPRING 2	SUMMER 1	SUMMER 2
KNOWLEDGE	**1.1.1 Architecture of the CPU** • The purpose of the CPU • Common CPU components and their function • Von Neumann architecture **1.1.2 CPU performance** • How common characteristics of CPUs affect their performance. **1.1.3 Embedded systems** • The purpose and characteristics of embedded systems	**2.1.2 Designing, creating and refining algorithms.** • Identify common errors. • Trace tables **2.1.3 Searching and sorting algorithms.** • Standard searching algorithms. • Standard sorting algorithms. **1.2.1 Primary storage (Memory)** • The need for primary storage • The difference between RAM and ROM	**1.2.2 Secondary storage** • The advantages and disadvantages of different storage devices and storage media relating to these characteristics: • Capacity • Speed • Portability • Durability • Reliability • Cost **1.2.3 Units** • The units of data storage. • How data needs to be converted into a binary format to be	**1.2.4 Data storage Sound** • How sound can be sampled and stored in digital form • The effect of sample rate, duration, and bit depth on: • The playback quality • The size of a sound file **1.2.5 Compression** • The need for compression • Types of compression: • Lossy • Lossless **2.2.1 Programming fundamentals** • The use of variables, constants, operators, inputs, outputs and	**2.2.2 Data types** • The use of data types: • Integer • Real • Boolean • Character and string • Casting **2.2.3 Additional programming techniques** • The use of basic string manipulation • The use of basic file handling operations • The use of records to store data • The use of SQL to search for data. • The use of arrays (or equivalent) when	**2.2.3 Additional programming techniques** • Random number generation **2.4.1 Boolean logic** • Simple logic diagrams using the operators AND, OR • and NOT • Truth tables • Combining Boolean operators using AND, OR and • NOT • Applying logical operators in truth tables to solve • Problems **1.5.1 Operating systems**

	• Examples of embedded systems **2.1.1 Computational thinking** • Principles of computational thinking: • Abstraction • Decomposition • Algorithmic thinking **2.1.2 Designing, creating and refining algorithms.** • Identify the inputs, processes, and outputs for a problem. • Structure diagrams • Create, interpret, correct, complete, and refine algorithms using: • Pseudocode • Flowcharts • Reference language/high-level programming language • Identify common errors.	• The purpose of ROM in a computer system • The purpose of RAM in a computer system • Virtual memory **1.2.2 Secondary storage** • The need for secondary storage • Common types of storage. • Suitable storage devices and storage media for a given application	processed by a computer • Data capacity and calculation of data capacity requirements **1.2.4 Data storage** **Numbers** • How to convert positive denary whole numbers to binary numbers • How to add two binary integers together and explain overflow errors which may occur. • How to convert positive denary whole numbers into 2-digit hexadecimal numbers and vice versa • How to convert binary integers to their hexadecimal equivalents and vice versa • Binary shifts **Characters** • The use of binary codes to represent characters. • The term 'character set' • The relationship between the number of bits per character in a character set, and the number of characters which can be represented. **Images** • How an image is represented as a	assignments • The use of the three basic programming constructs used to • control the flow of a program: • Sequence • Selection • Iteration (count- and condition-controlled loops) • The common arithmetic operators • The common Boolean operators AND, OR and NOT	solving problems, including both one-dimensional (1D) and two-dimensional arrays (2D) • How to use sub programs (functions and procedures) to produce structured code	• The purpose and functionality of operating systems: • User interface • Memory management and multitasking • Peripheral management and drivers • User management • File management **1.5.2 Utility software** • The purpose and functionality of utility software • Utility system software: • Encryption software • Defragmentation • Data compression

			series of pixels, represented in binary • Metadata • The effect of colour depth and resolution on: • The quality of the image • The size of an image file			
SKILLS	• Identifying • Describing • Explaining • Abstraction • Decomposition • Designing Algorithms • Creating Algorithms • Refining Algorithms • Writing in Pseudocode • Exam techniques	• Designing Algorithms • Creating Algorithms • Refining Algorithms • Identifying and correcting errors • Using algorithms to search for data. • Using algorithms to sort data. • Identifying • Describing • Explaining • Exam techniques	• Calculating file size • Converting data between binary and denary (vice versa) • Converting to Hexadecimal from binary (vice versa) • Converting to Hexadecimal from denary (vice versa) • Adding binary numbers together • Calculating text file size • Calculating image file size • Identifying • Describing • Explaining • Exam techniques	• Calculating sound file size • Assigning variables and constants • Assigning operators • Using inputs and outputs • Creating programs with selection • Creating programs with iteration • Using arithmetic operators • Using Boolean Operators • Analysing the task • Designing and writing programs using high level programming language. • Exam techniques	• Analysing the task • Designing and writing programs using high level programming language. • Assigning data types • Using string manipulation • Using file handling • Using SQL to search for data. • Creating arrays • Using sub programs • Exam techniques	• Using random number generation • Interpreting logic gates • Draw logic gate diagrams. • Completing truth tables • Applying logic operators in truth tables • Identifying • Describing • Explaining • Exam techniques
ASSESSMENT	**Marking Point 1:** OCR GCSE Computer Science exam questions on system architecture **Marking Point 2:** OCR GCSE Computer Science exam questions on computational thinking and algorithms	**Marking Point 1:** OCR GCSE Computer Science exam questions on algorithms **Marking Point 2:** OCR GCSE Computer Science exam questions on searching and sorting algorithms	**Marking Point 1:** OCR GCSE Computer Science exam questions on memory and storage **Marking Point 2:** Progress Test	**Marking Point 1:** OCR GCSE Computer Science exam questions on computational thinking and algorithms **Marking Point 2:** OCR GCSE Computer Science exam questions on computational thinking and algorithms	**Marking Point 1:** OCR GCSE Computer Science exam questions on computational thinking and algorithms **Marking Point 2:** OCR GCSE Computer Science exam questions on computational thinking and algorithms	**Marking Point 1:** OCR GCSE Computer Science exam questions on computational thinking and algorithms **Marking Point 2:** Progress Test

HOME LEARNING	**Home Learning 1:** Seneca Learning – Von Neuman Architecture **Home Learning 2:** Seneca Learning – Factors affecting CPU Performance and Exam Style Questions **Home Learning 3:** Seneca Learning – Computational Thinking and Algorithms	**Home Learning 1:** Seneca Learning – Interpreting, Correcting and Completing Algorithms **Home Learning 2:** Seneca Learning – Searching Algorithms **Home Learning 3:** Seneca Learning – Sorting Algorithms and Exam Style Questions	**Home Learning 1:** Revision Progress Test **Home Learning 2:** Seneca Learning – Memory **Home Learning 3:** Seneca Learning – Secondary Storage **Home Learning 4:** Seneca Learning – Units of Data	**Home Learning 1:** Seneca Learning – Number Representation **Home Learning 2:** Seneca Learning – Images and Sound	**Home Learning 1:** Seneca Learning – Compression and End of Topic Test **Home Learning 2:** Seneca Learning – Programming Fundamentals **Home Learning 3:** Seneca Learning – Data Types	**Home Learning 1:** Revision Progress Test **Home Learning 2:** Seneca Learning – Additional Programming Techniques **Home Learning 3:** Seneca Learning – Boolean Logic
READING, WRITING, TALK, NUMERACY	**Reading:** Students will read a range of different text as well as online resources. This half term students will focus on developing their skills in breaking down information and learning new vocab. **Writing:** Students will develop a range of different writing skills focusing on expository and answering exam questions. Some of the exam questions will be extended writing. **Oracy:** Students will focus on develop their listening and responding skills (Social and Emotional) and their use of appropriate language (Linguistic) **Numeracy:** Students will use a range of numeracy skills. They will use comparisons and calculations in their	**Reading:** Students will read a range of different text as well as online resources. This half term students will focus on developing their skills in learning new vocab, predict and infer. **Writing:** Students will develop a range of different writing skills focusing on summarising and answering exam questions. Some of the exam questions will be extended writing. **Oracy:** Students will focus on developing their clarity and summarising skills (Cognitive). They will also continue to develop their listening and responding. **Numeracy:** Students will use a range of numeracy skills. They will use operators within their algorithms to compare and calculate data.	**Reading:** Students will read a range of different text as well as online resources. This half term students will focus on developing their skills in asking questions, learning new vocab and infer. **Writing:** Students will develop a range of different writing skills focusing on descriptive and reflective writing. **Oracy:** Students will focus on developing their use of appropriate vocabulary choice (Linguistic). They will also develop working with others (Social and Emotional). **Numeracy:** Students will use a range of numeracy skills in order to calculate binary numbers, hexadecimal. Students will also add together 2 binary numbers.	**Reading:** Students will read a range of different text as well as online resources. This half term students will focus on developing their skills in relating to their own experience, infer and asking questions. **Writing:** Students will develop a range of different writing skills and develop their summarising skills further to ensure that they can explain the steps that need to be taken to convert data. **Oracy:** Students will focus on developing their self-regulation and clarifying and summarising skills (Cognitive). **Numeracy:** Students will use a range of numeracy skills. They will be required to calculate the data capacity of different file data types.	**Reading:** Students will read a range of different text as well as online resources. This half term students will focus on developing their skills in empathise, relating to experience and predict. **Writing:** Students will develop a range of different writing skills focusing on compare and contrast when answering exam questions. **Oracy:** Students will continue to develop their working with others and listening and responding skills (Social and Emotional). They will also focus on developing their reasoning skills (Cognitive). **Numeracy:** Students will use a range of numeracy skills. Students will need to use operators in their programs to compare data and make decisions.	**Reading:** Students will read a range of different text as well as online resources. This half term students will focus on developing their skills in relating to their own experience, infer and asking questions. **Writing:** Students will develop a range of different writing skills focusing on descriptive writing to be able to explain the different system software. Students will also continue to work on extended writing questions on this topic. **Oracy:** Students will continue to develop their social and emotional skills and their linguistic skills. Focusing on listening and responding and appropriate language choices.

	algorithms. They will also create a range of algorithms.					**Numeracy:** Students will use a range of numeracy skills. Students will use random number generator in their program to make decisions and make decisions.
TIER 2 VOCABULARY	- Identify - State - Explain - Complete - Justify - Describe - Define - Discuss - Write - Draw	- Identify - State - Explain - Complete - Justify - Describe - Define - Discuss - Write - Draw	- Identify - State - Explain - Complete - Justify - Describe - Define - Discuss - Write - Draw - Convert - Add - Give - Show - Calculate	- Identify - State - Explain - Complete - Justify - Describe - Define - Discuss - Write - Draw - Convert - Add - Give - Show - Calculate	- Identify - State - Explain - Complete - Justify - Describe - Define - Discuss - Write - Draw - Convert - Add - Give - Show - Calculate	- Identify - State - Explain - Complete - Justify - Describe - Define - Discuss - Write - Draw - Convert - Add - Give - Show - Calculate
TIER 3 VOCABULARY	- Central Processing Unit - Arithmetic Logic Unit - Control Unit - Cache - Registers - Embedded systems - Abstraction - Decomposition - Algorithm - Structure Diagram	- Searching Algorithm - Sorting Algorithm - Memory/ Primary Storage - Random Access Memory - Read Only Memory - Virtual Memory - Secondary Storage	- Characteristics - Units of Data - Denary - Binary - Hexadecimal - Binary Shift - Character Set - Pixels - Metadata - Colour depth	- Sound - Sample rate - Compression - Lossy - Lossless - Operators - Sequence - Selection - Iteration - Boolean	- Integer - Real - Casting - String manipulation - File Handling - Arrays - Procedure - Function	- Boolean Logic - Truth Tables - Logical operators - Operating system - Peripheral Management - Utility software - Encryption
PSPSMC, BRITISH VALUES AND DIVERSITY	**Personal:** Developing the valuable transferable skill of logical thinking. **Social:** Paired programming opportunities. **British value:** Consideration of the working environment of a programmer **Moral:** To be able to consider the end user and their needs when designing and creating programs. **Cultural:** Considering different cultures and backgrounds when creating and writing different programs.		**Personal:** Developing the valuable transferable skill of critical thinking. **Social:** Sharing ideas and being able to explain key topics. **British value:** Understanding how programs are created to comply with laws in data protection. **Moral:** Giving peer feedback in a respectful manner. **Cultural:** Understand and consider different cultures and backgrounds when representing data and information in a computer system.		**Personal:** Developing the valuable transferable skill of critical thinking. **Social:** Be able to present valid viewpoint to the class on topics **British value:** Understanding of the laws that govern computer systems and how they are design to protect people. **Moral:** Understand the impact that computer legislation has in keeping people safe.	

	Diversity: Examine Lillian Gilbreth work in flowcharts and how they impact computer science today. Also explore the work of Ada Lovelace and Al-Khwarizmi in their field of algorithms.	**Diversity:** examine key people involved in the development and creating of python software with focus on women.	**Physical:** Understand the design of computer programs. **Cultural:** Understand and consider different cultures and backgrounds and how their access to technology can impact their lives. **Diversity:** examine how Barbara Liskov helped to design and create the data types that we use today.