CHS Curriculum Intent

SUCCESSFUL: Learners who gain deep and powerful knowledge in preparation for life; combining academic rigour, curiosity and creative flair.

CREATIVE: Learners who are imaginative, optimistic and inventive; finding their voice to become effective communicators prepared for lifelong adaptability

HAPPY: Learners who are confident, resilient, well-rounded citizens; they understand the world's communities and are ready to discover their place in it.

CHS Curriculum Area Framework for Learning – Year 10

SUBJECT	Computer Science
INTENT	"I believe that at the end of the century, the use of words and general educated opinion will have altered so much that one will be able to speak of machines thinking without expecting to be contradicted" Alan Turning • Studying Computing at Chorlton High School allows our students to be prepared for a digital world, digital careers and how to use digital systems safely and responsibly. • Students will be taught the key skills to develop their digital literacy and awareness of computational thinking. • We aim to help our learners to become happy, successful and creative through our curriculum with students exploring a range of ways that the technology they interact with on a daily bases works and operates. Students are also able to develop their awareness through applying some of these in skillful tasks and activities.

Department: Computing & Technology 2020-2021

Subject: GCSE Computer Science *(OCR J277 (from 2020)) – new specification*

Year Group	10				
Rationale/ Narrative	During Year 9 students who opted to study Computing, spent their first term studying a common curriculum to develop their knowledge and understanding of key Computer Science and Digital Information Technology concepts. Students studied concepts including Cyber Security, User Interfaces, Hardware and Programming. Students who followed the Computer Science route after the Autumn term continued then explored a deeper context of Computer Systems, building upon their existing knowledge with Computer Systems, specifically learning about a wide scope of topics including 1.1 Systems Architecture, 1.2 Memory and storage, 1.3 Computer networks, connections and protocols. Students further developed their knowledge and understanding of programming whilst studying key 2.2 programming fundamentals such as the use of variables, inputs, outputs and assignments using the three basic programming constructs used to control the flow of a program: sequence, selection and iteration (count-and condition-controlled loops) and common arithmetic operators. Students demonstrated their programming aptitude through a range of complex challenges which required them to test, identify and resolve syntax and logic errors. Finally, students started to explorer a broader range of programming topics relating to data representation (Boolean logic, simple llogic diagrams using the operators AND, OR and NOT, Truth tables`, characters, images, sound and compression), whilst also considering the Ethical, legal, cultural and environmental impact of digital technology on wider society and legislation relevant to Computer Science. The main focus for Year 10 is in building confidence and skill levels with Computational thinking, algorithms and programming principles to encourage them to think creatively, innovatively, analytically, logically and critically, applying their understanding to a wide range of problems. Students will apply their skills to complete a major Programming Project in the Summer term and will therefore revisit programming skills regularly throughout the year (suggested at least one hour every fortnight, however during the Spring term this will be much more often). Note: OCR have updated the specification for the GCSE (9-1) Computer Science qualification (J277), and therefore this cohort of students will begin to study to the new specification from September 2020 as they will be examined via the new specification in Summer 2022.				
	Autumn 1	**Autumn 2**	**Spring 1**	**Spring 2**	**Summer 1**	**Summer 2**
KNOWLEDGE	**2.1 – Algorithms** **2.1.1 Computational thinking** • Abstraction • Decomposition Algorithmic thinking **2.1.2 Designing, creating and refining algorithms**	**2.1.3 Searching and sorting algorithms** • Binary/Linear search • Bubble sort • Merge sort • Insertion sort	**2.2.3 Additional programming techniques** • Basic string manipulation • Basic file handling operations: • Open	**2.3 Producing robust programs** **2.3.1 Defensive design** • Defensive design considerations: o Anticipating misuse	**2.5 – Programming languages and IDEs** **2.5.1 Languages** • Characteristics/purpose of different levels of programming language: • High-level	**Programming project (Netflix type program)** - Analysis - Design - Development - Testing, evaluation, and conclusions

¨ 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¨ - Trace tables **Programming Skills**	**2.2 – Programming fundamentals** **2.2.2 Data types** - Integer - Real - Boolean - Character and string - Casting **2.2.1 Programming fundamentals** - Use of variables, constants, operators, inputs, outputs and assignments¨ - Basic programming constructs used to control the flow of a program: - Sequence - Selection - Iteration (count- and condition-controlled loops)¨ - Common arithmetic operators¨ - Common Boolean operators AND, OR and NOT	- Read - Write - Close - The use of records to store data - The use of SQL to search for data¨ - Arrays (one-dimensional and two-dimensional) - Sub programs (functions and procedures) to produce structured code - Random number generation	○ Authentication - Input validation - Maintainability: - Use of sub programs - Naming conventions - Indentation - Commenting **2.3.2 Testing** - The purpose of testing - Types of testing: - Iterative - Final/terminal - Identify syntax and logic errors - Selecting and using suitable test data: - Normal - Boundary - Invalid - Erroneous - Refining algorithms	- Low-level • The purpose of translators • The characteristics of a compiler and an interpreter **2.5.2 The IDE** **Common tools and facilities available in an IDE** - Editors - Error diagnostics - Run-time environment - Translators Extended programming practice (covering any skills that may need improvement) **2d. Practical Programming Project (Netflix type program)** The programming task(s) must allow them to develop skills within the following areas when programming: • Design• Write• Test• Refine - Analysis - Design - Development - Testing, evaluation, and conclusions	**Structured revision based on exam analysis/ revisit Data Representation (Distance Learning)** - Units - Numbers - Characters - Images - Sound - Compression – *topics also revisited throughout the academic year via home learning tasks.*
SKILLS	- Evaluation skills - Metacognitive practice - Exam technique - Identifying and selecting information - Breaking down key information - Programming skills	- Evaluation skills - Metacognitive practice - Exam technique - Identifying and selecting information - Breaking down key information **Programming skills**:	- Evaluation skills - Metacognitive practice - Identifying and selecting information - Breaking down key information	- Evaluation skills - Metacognitive practice - Exam technique - Identifying and selecting information - Breaking down key information	- Evaluation skills - Metacognitive practice - Computational thinking - Identifying and selecting information - Breaking down key information	- Evaluation skills - Metacognitive practice - Exam technique - Identifying and selecting information

		- Identifying and using variables - Using different types of data appropriately - Using operators - Using inputs - Using outputs - Using sequence - Using selection - Using iteration (for loops) - Using iteration (while loops)	- Exam technique **Programming skills**: - Using basic string manipulation - Using records to store data - Using SQL to search for data - Using arrays (one-dimensional and two dimensional) - Using sub programs (functions and procedures) - Using random number generation		**Programming skills**: - Identifying and using variables - Using different types of data appropriately - Using operators - Using inputs - Using outputs - Using sequence - Using selection - Using iteration (for loops) - Using iteration (while loops) - Using basic string manipulation - Using records to store data - Using SQL to search for data - Using arrays (one-dimensional and two dimensional) - Using sub programs (functions and procedures) - Using random number generation	Breaking down key information **Programming skills**: See Summer 1
ASSESSMENTS	**Baseline Assessment: via MS Forms to identify misconceptions from topics learnt during Year 9 (in school/via distance learning) which will need to be revisited. – * AUTUMN KEY ASSESSED PIECE 1** *(*additional assessment piece due to requirement for distance*	- Autumn Progress test (Paper 1 and Paper 2) with exam questions based on Algorithms/Programming Fundamentals and Year 9 topics including Distance Learning. – **AUTUMN KEY ASSESSED PIECE** 4 - OCR GCSE Computer Science Paper 2 Past	OCR GCSE Computer Science Paper 2 Past questions\2.2 Programming Fundamentals Q4 (*players array) and Q5 (medicine) – **SPRING KEY ASSESSED PIECE 1** *(*Q4 also on Year 11's Framework for*	Spring Progress test (Paper 2) with exam questions based on Algorithms, programming fundamentals, robust programs and testing and Year 9 topics including Distance Learning. – **SPRING KEY ASSESSED PIECE** 4	- OCR GCSE Computer Science Paper 2 Past questions\2.5 Programming languages and IDEs Q-2 **SUMMER ASSESSED PIECE 1** - Practical Programming Project (Netflix type program) – Analysis &	Summer Progress test (Paper 1 & 2) with exam questions covering topics learnt throughout Year 9 and 10. – **SPRING KEY**

	learning provision during Spring/Summer term.) • **OCR GCSE Computer Science Paper 2 Past questions\2.1.2 Designing, creating and refining Algorithms Past Papers Q1 [6 marks] - AUTUMN KEY ASSESSED PIECE 2.**	questions\OCR GCSE Computer Science Paper 2 Past questions\2.2 Programming Fundamentals Q3 (Drinks Machine) **– AUTUMN KEY ASSESSED PIECE 5** *(also on Year 11's Framework for Learning to be replaced with a different AP for this cohort next year).* • End of unit test via MS Forms: 2.1 Algorithms	*Learning to be replaced with a different AP for this cohort next year).*	• End of unit test via MS Forms: 2.2 P**programming fundamentals** • End of unit test via MS Forms: 2.3 producing robust programs	Design – **SUMMER KEY ASSESSED PIECE 2**	**ASSESSED PIECE 3** • Practical Programming Project (Netflix type program) – Dev & Test – **SUMMER KEY ASSESSED PIECE 4**