



CHORLTON HIGH SCHOOL: CURRICULUM

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	<p style="text-align: center;">“Those who can imagine anything, can create the impossible.” — Alan Turing</p> <p>As we live in an ever-changing digital world, Computer Science impacts everything from gaming, to medicine, banking, communications and connectivity. Most individuals and businesses rely on sophisticated computer systems and software throughout their daily lives and operations. Learners today are preparing for careers that might not even exist yet, and we hope our students can be one of the Computer Scientists who makes the world better, faster and more inter-connected!</p> <p>Studying Computer Science at Chorlton High School, will provide our learners with the tools, knowledge and skills to develop their computational thinking, programming, problem solving and analytical skills. Our learners can then apply these to a variety of programming projects. Studying Computer Science will also allow our learners to access a wider range of topics including digital literacy, e-safety, computer systems, networks, data representation and the positive and negative impact of computing on modern society.</p>

Department: **Computing & Technology 2020-2021**

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

Year Group	Year 10
Rationale/ Narrative	<p>During Year 9 students who opted to study Computing studied an introduction to computer systems, Modern Technology and its impact, Legal impacts and Ethical considerations relating to technology, Cyber Security and programming.</p> <p>Students continued to then explore 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 and 1.3 Computer networks, connections and protocols. Students further developed their knowledge and understanding of programming whilst studying 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.</p>



CHORLTON HIGH SCHOOL: CURRICULUM

Alongside these fundamental programming skills, students started to explore a broader range of programming topics relating to data representation (Boolean logic, simple logic diagrams using the operators AND, OR and NOT, Truth tables, Trace Tables, High-Level languages and low level languages, translators as well as considering how to ensure their code is maintainable).

During the Autumn Term, students will revisit Paper 1: Computer Systems and focus their studies on Data Storage, specifically looking at how to convert between binary, denary and hexadecimal, alongside binary addition and binary shifts. Following on from this, students will then look at how characters, images and sound are stored using binary, leading into the compression (both lossy and lossless) of these file types. Students will finish Autumn 1 by looking at 1.5 System Software, covering both the operating system and utility software whilst considering the ethical, legal, cultural and environmental impact of digital technology, including legislation regarding the Data Protection Act, Computer Misuse Act, Copyright Design and patents Act as well as software licenses (open source and proprietary). The main focus for Year 10 will be 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).

Regarding the above, students will focus specifically on Searching and Sorting Algorithms, a range of additional programming techniques (file handling, the use of records to store data, SQL, 1D & 2D arrays as well as building upon pre-existing knowledge regarding sub-programs through looking at functions). Students will understand a range of defensive design techniques that should be considered when programming, including anticipating misuse & authentication and validation, before moving onto looking at how and why programs should be tested.

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	<p>Data Storage</p> <p>Numbers</p> <ul style="list-style-type: none"> Binary to Denary & Vice Versa Binary to Hexadecimal & Vice Versa Binary Addition (Overflow) Binary Shifts <p>Characters</p> <ul style="list-style-type: none"> Use of binary codes to represent characters The Term 'Character Set' Relationship between number of bits per character in a character set 	<p>Networks, Protocols & Layers</p> <ul style="list-style-type: none"> DNS Hosting IP addressing and MAC addressing Standards Common Protocols The concept of Layers <p>Systems Software</p> <p>Operating System</p> <ul style="list-style-type: none"> Purpose and functionality of operating systems User Interface Memory Management & Multi-tasking 	<p>Additional programming techniques</p> <ul style="list-style-type: none"> Basic string manipulation Basic file handling operations: <ul style="list-style-type: none"> Open Read Write Close The use of records to store data The use of SQL to search for data' Arrays (one-dimensional and two-dimensional) 	<p>Producing robust programs</p> <p>Defensive design</p> <ul style="list-style-type: none"> Defensive design considerations: <ul style="list-style-type: none"> Anticipating misuse Authentication Input validation Maintainability: Use of sub programs Naming conventions Indentation Commenting <p>Testing</p> <ul style="list-style-type: none"> The purpose of testing Types of testing: <ul style="list-style-type: none"> Iterative 	<p>Programming languages and IDEs</p> <p>Languages</p> <ul style="list-style-type: none"> Characteristics/purpose of different levels of programming language: <ul style="list-style-type: none"> High-level Low-level The purpose of translators The characteristics of a compiler and an interpreter <p>The IDE</p> <p>Common tools and facilities available in an IDE</p>	<p>Programming project (Netflix type program)</p> <ul style="list-style-type: none"> Analysis Design Development Testing, evaluation, and conclusions <p>Structured revision based on baseline assessment carried out on topics covered during Y9 remote learning/ revisit</p>



CHORLTON HIGH SCHOOL: CURRICULUM

	<ul style="list-style-type: none"> • ASCII • Unicode <p>Images</p> <ul style="list-style-type: none"> • How images are represented in pixels, in binary • Metadata • The effect of colour depth on quality and size <p>Sound</p> <ul style="list-style-type: none"> • How sound can be sampled and stored in digital • The effect of sample rate, duration and bit depth on quality and size <p>Networks Recap</p> <ul style="list-style-type: none"> • Recap content regarding networks that was delivered during Y9, preparing for advanced networking topics. 	<ul style="list-style-type: none"> • Peripheral Management and Drivers • User management • File management <p>Utility Software</p> <ul style="list-style-type: none"> • The purpose and functionality of utility software • Utility system software • Encryption software • Defragmentation • Data Compression <p>Utility Software</p> <ul style="list-style-type: none"> • Data protection Act 2018 • Computer Misuse Act 2019 • Copyright Designs and patents Act 1988 • Software Licenses (open source and proprietary) 	<ul style="list-style-type: none"> • Sub programs (functions and procedures) to produce structured code • Random number generation 	<ul style="list-style-type: none"> ○ Final/terminal • Identify syntax and logic errors • Selecting and using suitable test data: <ul style="list-style-type: none"> ○ Normal ○ Boundary ○ Invalid ○ Erroneous • Refining algorithms 	<ul style="list-style-type: none"> • Editors • Error diagnostics • Run-time environment • Translators <p>Extended programming practice (covering any skills that may need improvement)</p> <p>Practical Programming Project (Netflix type program)</p> <p>The programming task(s) must allow them to develop skills within the following areas when programming: • Design • Write • Test • Refine</p> <ul style="list-style-type: none"> • Analysis • Design • Development • Testing, evaluation, and conclusions 	
<p>SKILLS</p>	<ul style="list-style-type: none"> • Evaluation skills • Metacognitive practice • Exam technique • Identifying and selecting information • Breaking down key information • Programming skills 	<ul style="list-style-type: none"> • Evaluation skills • Metacognitive practice • Exam technique • Identifying and selecting information • Breaking down key information 	<ul style="list-style-type: none"> • Evaluation skills • Metacognitive practice • Identifying and selecting information • Breaking down key information • Exam technique <p>Programming skills:</p> <ul style="list-style-type: none"> • Using basic string manipulation • Using records to store data • Using SQL to search for data 	<ul style="list-style-type: none"> • Evaluation skills • Metacognitive practice • Exam technique • Identifying and selecting information • Breaking down key information <p>Programming skills:</p> <ul style="list-style-type: none"> • Defensive Design Techniques • Input sanitization • Writing maintainable code 	<ul style="list-style-type: none"> • Evaluation skills • Metacognitive practice • Computational thinking • Identifying and selecting information • Breaking down key information <p>Programming skills:</p> <ul style="list-style-type: none"> • All previously mentioned programming skills pulled into one project. 	<ul style="list-style-type: none"> • Evaluation skills • Metacognitive practice • Exam technique • Identifying and selecting information • Breaking down key information <p>Programming skills:</p> <ul style="list-style-type: none"> • All previously mentioned programming skills pulled into one project.



CHORLTON HIGH SCHOOL: CURRICULUM

			<ul style="list-style-type: none"> Using arrays (one-dimensional and two dimensional) Using sub programs (functions and procedures) Using random number generation 	<ul style="list-style-type: none"> Testing programs with numerous data types 		
ASSESSMENTS	<p>Key Assessment Piece:</p> <ul style="list-style-type: none"> Baseline Assessment: via MS Forms to identify misconceptions from topics learnt during Year 9. Data Representation: Numbers (<i>Exam questions</i>) Classwork piece Data representation Image & Sound – (<i>Exam questions</i>) Classwork piece 	<p>Computer Science Progress Checkpoint</p> <p>Students will have a progress checkpoint assessment to assess their knowledge and understanding of the topics covered in this unit and across the course (2 papers)</p> <ul style="list-style-type: none"> Networks – Protocols & Layers (<i>Exam questions</i>) Classwork piece Utility Software - (<i>Exam questions</i>) Classwork piece 	<p>Key Assessment Piece:</p> <ul style="list-style-type: none"> OCR GCSE Computer Science Paper 2 Past questions\2.2 Programming Fundamentals Q4 (*players array) <p>Key Assessment Piece:</p> <p>OCR GCSE Computer Science Paper 2 Past questions\2.1.2 Designing, creating and refining Algorithms Past Papers Q1 [6 marks]</p> <p>Key Assessment Piece:</p> <ul style="list-style-type: none"> OCR GCSE Computer Science Paper 2 Past questions\2.2 Programming Fundamentals – Sub-Programs 	<p>Computer Science Progress Checkpoint</p> <p>Students will have a progress checkpoint assessment to assess their knowledge and understanding of the topics covered in this unit and across the course (2 papers)</p> <p>Key Assessment Piece:</p> <p>OCR GCSE Computer Science Paper 2 Past questions – Defensive Design Considerations (Misuse, authentication & sanitization)</p> <p>Key Assessment Piece:</p> <p>OCR GCSE Computer Science Paper – Purpose of Testing</p>	<p>Key Assessment Piece:</p> <ul style="list-style-type: none"> OCR GCSE Computer Science Paper 2 Past questions\2.5 Programming languages and IDEs Q-2 <p>Key Assessment Piece:</p> <ul style="list-style-type: none"> Practical Programming Project (Netflix type program) – Analysis & Design 	<p>Computer Science Progress Checkpoint</p> <p>Students will have a progress checkpoint assessment to assess their knowledge and understanding of the topics covered in this unit and across the course (2 papers)</p> <p>Key Assessment Piece:</p> <ul style="list-style-type: none"> Practical Programming Project (Netflix type program) – Dev & Test –