

Subject Curriculum Map: Computer Science Year 10 2024-25

Exam board: OCR – 100% Terminal Examination

Curriculum intent: Year 10 Computer Science will build on the knowledge, understanding and skills established from year 9, such as advanced programming skills and the role of the CPU. The content will provide a solid basis of understanding and engage learners and get them thinking about real world application. The curriculum will also enable learners to develop computational thinking skills built on a sound base of conceptual learning and understanding. During the course of the year students will think creatively, innovatively, analytically, logically and critically.

Curriculum Implementation- The course is delivered as 5 fortnightly lessons. We have built in common assessment points to allow all classes to move through the plan at a similar pace (with stretch and challenge built in for the higher achieving students) to better allow for movement of students through the years leading up to the terminal examinations. Students will gain a deeper understanding of the components of the computer system, such as the main parts of the CPU. Each half term, students will cover theory topics, such as binary and networks and programming techniques, where they will learn a new skill in python, starting from the basics and then moving on to more advanced techniques by the summer term. During Summer 2, the students will take part in a practical task where they develop the skills to design, write, test, and refine program(s) in Python, to solve a problems.

Curriculum impact: Students will understand and apply the fundamental principles and concepts of Computer Science, including abstraction, decomposition, and logic, algorithms, and data representation. They will develop skills and understanding in computational thinking: algorithms, programming techniques, producing robust programs, computational logic, translators and data representation. The skills and knowledge developed within this curriculum will support the students when completing practical Programming tasks. They will analyse problems in computational terms through practical experience of solving such problems, including designing, writing and applying mathematical skills relevant to Computer Science. Students will also understand the components that make up digital systems, and how they communicate with one another and with other systems. The curriculum will also allow the students to understand the impacts of digital technology to the individual and to wider society.

Year 10 Computer Science	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
Themes, Concepts & Ideas	1.2.3 Units of data storage 1.2.4 Binary & hexadecimal 1.2.4 Characters 2.1 Computational Thinking Flowcharts	1.2.4 Images & Sound 1.2.5 Compression 2.1 Searching and sorting algorithms (showing the stages)	1.1.1 Architecture of the CPU 1.1.2 CPU performance 2.2.1 Programming fundamentals (constructs & data types)	1.1.3 Embedded systems 1.2.1 Primary storage (Memory) 1.2.2. Secondary storage 2.2.3 Sub programs	1.3.1 Networks and topologies 2.2.3 1D arrays	1.3.2 Wired & wireless networks 1.6.1 Ethics and legislation Programming tasks
Knowledge and understanding	See below	See below	See below	See Below	See Below	See Below
Subject specific skills	Explain how the computer understands binary. Explain how the characters on the keyboard are converted to binary. Be able to convert binary numbers to denary, Binary to hexadecimal and vice versa. Describe the terms abstraction, decomposition and algorithmic thinking and how these skills are	Students will explain how images and sound are represented in binary. Describe the difference between lossy and lossless compression. Show the stages of the searching and sorting algorithms (binary, linear, bubble, merge, insertion)	Identify and explain the role of the main components of the CPU. Describe the factors that affect the performance of the CPU (cores, cache, clock speed). Identify different data types such as strings, integers, Boolean. Write simple programs in python, using the 3 programming constructs.	Explain the features of an embedded system and give examples around the home. Describe the difference between RAM and ROM. Compare and contrast the features of secondary storage devices. Choose a suitable storage device for a given scenario.	Explain the difference between a LAN and a WAN. Describe the different topologies, such as star and mesh. Understand how our school network is arranged. The use of 1D arrays when solving problems.	Compare benefits and drawbacks of wired vs wireless connections. Recommend one or more connections for a given scenario. IP and MAC addressing Explain the principles of protocols and layers. Discuss the impacts of digital technology on wider society including, ethical,

	<p>key to problem solving.</p> <p>Create simple flowcharts for algorithms.</p>		<p>Sequence, selection, iteration.</p>	<p>Write simple programs using procedures and functions.</p>		<p>legal, cultural, environmental and privacy issues.</p> <p>Legislation relevant to CS, such as the DPA, CMA and different software licences.</p>
1.SMSC	<p>Social: Discussing problems and how we break them down in everyday situations.</p>	<p>Social, moral and Cultural: Collaboration and lots of group work when searching and sorting lists.</p>	<p>Social, cultural: Understanding why some computers perform better than others.</p>	<p>Social, cultural: Working together with practical tasks comparing different storage devices, deciding together which would be the most suitable for different scenarios.</p>	<p>Social, cultural: Understanding how our school is connected together. How other cultures communicate. Students may work together to solve problems and to produce programs in python.</p>	<p>Social, cultural: Understanding different internet connections and why some areas are unable to have an internet connection. Students may work together to solve problems and to produce programs in python.</p>
2.Skills For life	<p>Collaborative working, building relationships and responding to feedback – both formal and verbal.</p>	<p>Using different methods to search and sort information. Communication skills when working together to follow the algorithms.</p>	<p>Collaborative working, building relationships and responding to feedback – both formal and verbal.</p>	<p>Collaborative working, building relationships and responding to feedback – both formal and verbal.</p>	<p>Collaborative working, building relationships and responding to feedback – both formal and verbal</p>	<p>Problem solving and resilience when solving practical problems.</p>
3.FBV	<p>Values of respect, collaboration and group work will be developed.</p>	<p>Collaboration with teams, patience and timing. Demonstrating tolerance and respect for others.</p>	<p>Collaboration with teams, patience and timing. Demonstrating tolerance and respect for others.</p>	<p>Collaboration with teams, patience and timing. Demonstrating tolerance and respect for others.</p>	<p>Collaboration with teams, patience and timing. Demonstrating tolerance and respect for others.</p>	<p>Collaboration with teams, patience and timing. Demonstrating tolerance and respect for others.</p>
Stretch & challenge	<p>Higher ability will be able to implement</p>	<p>High ability will justify which</p>	<p>High ability students will explain how</p>	<p>High ability to write functions</p>	<p>High ability to manipulate a 1D</p>	<p>High ability to justify connection methods referring</p>

	iteration and sub procedures into their flowcharts.	sorting algorithm would be most appropriate for different lists.	factors that affect the performance of networks can be improved or possibly overcome.	and procedures with parameters.	array in python using iteration.	to context and applying knowledge.
Key assessment focus, suggested assessments	<u>Autumn 1 assessment point</u>	<u>Autumn 2 Assessment point</u>	<u>Spring 1 Assessment point</u>	<u>Spring 2 Assessment point</u>	<u>Summer 1 Assessment point</u>	<u>Summer 2 Assessment point</u> Programming task
Special events						
Visits/extra-curricular	Chess club Coding/homework club	Chess club Coding/homework club	Chess club Coding/homework club	Chess club Coding/homework club	Chess club Coding/homework club	Chess club Coding/homework club
Homework/Independent Learning	Practise tasks on binary/hexadecimal calculations.	Independent research- when and where do searching and sorting algorithms take place?	Homework tasks on key terms.	Homework tasks on key terms.	Research and activities into network topologies. Investigation into businesses and their networks.	Students will work on their programming project to meet deadlines.