Curriculum Map
Year 10

Computer Science

Learning Aims and Curriculum Intent:

The OCR GCSE in Computer Science is designed to encourage students to develop their understanding and application of the core concepts in computer science. Students also analyse problems in computational terms and devise creative solutions by designing, writing, testing, and evaluating programs.

Component 01: Computer systems. Introduces students to the central processing unit (CPU), computer memory and storage, data representation, wired and wireless networks, network topologies, system security and system software. It also looks at ethical, legal, cultural, and environmental concerns associated with computer science.

Component 02: Computational thinking, algorithms, and programming. Students apply knowledge and understanding gained in component 01. They develop skills and understanding in computational thinking: algorithms, programming techniques, producing robust programs, computational logic, and translators.

Practical programming. Students are to be given the opportunity to undertake a programming task(s) during their course of study which allows them to develop their skills to design, write, test and refine programs using a high-level programming language. Students will be assessed on these skills during the written examinations, in particular component 02 (section B).

Term	Content, Key Questions and Knowledge	Skills	Assessment
Michaelmas	**1.1 Systems Architecture** How do computers actually work? This unit looks at what goes on inside a computer and examines the operation of the CPU and the fetch decode execute cycle. The components of the CPU, what affects its performance, Von Neumann architecture. Detail looks at the components of the processor (PC, ALU, clock, CU, registers, cache). Embedded systems are also investigated. **2.1 Algorithms** Understanding the principles of computational thinking. This unit investigates the main concepts of computational thinking which involves abstraction, decomposition and algorithmic thinking and knowing how to use pseudocode, flowcharts and OCR reference language to represent algorithms. We will also look at the different searching (linear, binary) and sorting (bubble, merge, insertion) algorithms as well as their advantages and disadvantages. **1.2 Memory and Storage** How do computers remember and store data? This unit looks at the different types of memory a computer has and how these are used. The concepts of volatile and non-volatile memory and why we need both. Detail investigations of secondary storage technology such as magnetic, optical and SSD are conducted so the pupils can understand their advantages/disadvantages. The different types of files (text, images, sound) are also taught as well as how to calculate their sizes. This leads to understanding why and how compression are required and utilised. **2.2 Programming Fundamentals** How to program in a high-level language. In this unit, Python language is used to teach fundamental constructs such as sequence, selection and iteration as they are used to write programs/applications. Arithmetic and logic operators are investigated. The pupils are taught about variables, constants and the different data types they store. Basic file handling methods (open, read, write, close) are learned so data can be imported or exported to external documents. One-dimension and two-dimensions arrays are looked at and how data can be retrieved, updated from records using SQL commands.	- Be able to define a computer system and what a Von Neumann architecture computer is. Can identify the components inside a CPU and explain how they work together to carry out a F-D-E cycle. - Can define the key terms in computational thinking and can use different methods to describe an algorithm. Know how to search through a set of data using linear and binary methods and can show the steps involved in sorting a set of data using bubble, merge and insertion sorts. - Can explain clearly the different between volatile and non-volatile memories and can describe the characteristics of primary memory. Be able to recommend which secondary storage device to use based on user's requirements. Can calculate the sizes of text, image and sound files and can recommend the correct compression method for these files. - Can program using all the constructs and know when to use global/local variables effectively. Can write code to import/export data externally and know SQL commands to describe how data can be retrieved from a database.	**End-of-topic test.** **Test at the end of the Michaelmas term.**

Lent	**1.3 Computer Networks, Connections and Protocol** Understand what a network is made of, and how it works. This unit investigates LAN, WAN networks and the hardware requirements. Different topologies (star, mesh) are investigated, and the pupils are taught what happens when a browser makes a request and the role of a Domain Name Server. Protocols, IP and MAC addresses, wired/wireless connection, packet switching, layers stack are important concepts being taught in this unit. **2.3 Producing Robust Programs** Understand how a program is made robust and the concept of defensive design. Pupils learn how authentication, validation and testing are all considerations they must look at when writing programs. Different methods/techniques are used to verify users, check user inputs as well as testing and debugging are looked at in this unit. **1.4 Network Security** Know how to safeguard from malware and other types of computer threads This units looks at forms of attack like virus, worms, Trojan horse, brute-force attacks, DOS, SQL injection and the methods to protect ourselves from these (firewalls, anti-malware, encryption etc.). Physical security is also taught such as locked room, passwords, biometric passes. **2.4 Boolean Logic** Know the different logic gates, their truth tables and how apply them in different settings. Understand about the AND, OR and NOT gates as well as their truth tables. Pupils are also expected to be able to combine these gates to create more complex logics.	- Can describe and give examples of LAN and WAN networks and be able to explain the different hardware requirements. Can describe clearly the steps involved when a URL is entered into a browser regarding DNS. Can describe the components of an IP/MAC address and state when each one is used. Can list the benefits of having different protocol layers. - Can write programs in a robust method using the many methods of authentication, data validation and testing. - Be able to list ways of safeguarding a network of devices using firewall, anti-malware software and encryption. Can describe methods of physically protect data such as secure locked room, biometric pass and CCTV. - Can use AND, OR, NOT gates to complete truth tables and can draw logic diagrams given a set of conditions.	**End-of-topic test.** **Test at the end of the Lent term.**
Trinity	**1.5 Systems Software** Understand the Operating System of a computer and the different utility software. This units looks at the roles of the OS in providing an interface and managing different aspects of the computer such as memory, processor, files and users. Utility programs like encryption, defragmentation and compression applications are investigated in detail. **2.5 Programming Languages and IDE** Know the characteristics and purpose of different levels of programming languages. Understand high-level and low-level languages like Python and machine code respectively. This unit also looks at types of translators and how they are used to convert programs from high-level to low-level. Their advantages/disadvantages are investigated in detail. **1.6 Ethical, Cultural and Environmental Impacts of Technology** How technology impacts on society concerning ethical, cultural, environmental and legal issues. Understand the impacts of technology such as robot and AI can have on different societies around the world and know about legislations that are used to guard against certain legal impact. This unit also looks at proprietary and open-source software and their different usage situations.	- Can state the main functions of an operating system and can describe clearly what each function does. Can explain what a utility software is and what it does. Be able to describe clearly what a compression, defragmentation or encryption program does and know of typical misconceptions. - Be able to define what a high-level/low-level programming language is and give example of each. Can state the advantages/disadvantages of each. Can explain clearly the types of translators and their advantages/disadvantages. Be able to define what an IDE is and can state the benefits of an IDE. - Can associate technology or new/recent development in computer related fields to their impacts on ethical, cultural, environmental and legal issues. Be able to write clear and meaningful essays on these issues.	**End-of-topic test.** **End-of-year examinations.**

What consolidation looks like in this subject	Only for A-Level and GCSE, delete for KS3.	
Examples of Homework	Practice topic questions, watching *Craig n Dave* videos and making notes, doing past GCSE exam questions. Reading from assigned computer science textbook and making notes.	
Key terminology	Click here for a comprehensive list of terminologies (and their definitions) on the Yr10 school intranet page. Scroll down to the **Yr10 Revision** box to see the list.	
Super-curricular enrichment and scholarly extension	• **Read:** Assigned CS Textbook (**OCR GCSE Computer Science (9-1), J277**) from MCI Resources • **Watch: BBC Click, CraignDave, BBC Technology** • **Listen:** • **Visit:**	
Useful websites	**https://www.youtube.com/@craigndave** **BBC Click** **https://isaaccomputerscience.org/**	
Who can I contact?	**Head of Department**	Mark Jalowiecki - msj@forest.org.uk
	Teachers	Vinh Kieu - vk@forest.org.uk