Our Lady’s Catholic College

Computer Science – Key Stage 4 Curriculum Overview

Our Lady’s Catholic College, Morecambe Road, Lancaster, LA1 2RX

GCSE COMPUTER SCIENCE

Our vision @ Our Lady's Catholic College:

Computer Science is a dynamic and rapidly growing field of study and has quickly become an integral part of modern society. Through our curriculum we aim to develop students' computational thinking and encourage their creativity, enabling them to decompose complex problems and find efficient solutions. We equip students with a deep understanding of computational theory and emerging technologies and provide them with a rich experience in practical programming so that they can bring their ideas to life. We want our students to leave us as digitally literate individuals who are well-prepared to engage with and thrive in, an ever-changing technological world.

Our Intent:

Computer Science has been designed to maximise progression. Topics covered revisit prior learning, as well as enabling students to deepen their understanding of the core concepts of computer science. Topics will be taught from both papers concurrently, maximising chances to revisit more challenging content in retrieval practice. New content and topics are also introduced throughout year 10. Computer programming is taught throughout year 10 to overcome the forgetting curve when students have lengthy periods of time without utilising these skills. All students will be able to access the main content of all lessons and all students will be taught to the top with scaffolding, adaptive teaching and stretch and challenge provided where necessary.

Curriculum Implementation:

When delivering programming content, teachers will model coding where the core skills are introduced. Students will then be given a similar task to complete after the demonstration and may be given example code to copy and then edit. During Students will then be given a real world concept scenario that require them to apply the skills learnt. This will help solidify this knowledge, and also help students understand when and where to use programming techniques in solving computational problems.

Lesson sequences have been carefully chosen to ensure that students have the required background knowledge to fully understand and apply skills in relation to the topic. Therefore, the lesson sequence may not match that of the exam specification, however all content is covered. This is especially important for paper 2, where students need concrete understanding of the key programming techniques before applying these to produce GCSE Algorithms.

Lessons follow a consistent format beginning with a retrieval practice activity in the form of Revise, Recap, Review. This will normally involve students answering 3 questions from last lesson, followed by 2 questions from previous study and one more challenging question. Each activity will involve students being posed questions interleaved over multiple units delivered throughout the year. Students are encouraged to work independently through the provision of scaffolding where required. Computing lessons often involve the application or practical/technical skills. These will be modelled to students using the I do, we do, you do approach. Students will be assessed at the end of each unit. Following assessment, students will complete a follow up activity based upon the individual areas to be improved that have been identified.

Curriculum Impact:

Students will have deepened their understanding of the computer system and know how instructions are fetched and executed by the CPU. Students will be more confident in their problem solving abilities and will be able to apply computational thinking strategies to solve problems. The year 10 curriculum will foster students' intellectual curiosity around topics delivered.

YEAR 10 OVERVIEW

Term	Overview	Building On	Cultural Capital	
1	**SLR 1.1 Systems architecture** In System Architecture students will look at the Fetch, Decode and Execute cycle and how this is applied within processors using Von Neumann Architecture. They study the individual registers that make up a processor and how their individual roles support the processing of data and instructions. **SLR 1.2 Memory and storage** Students begin by learning some of the most important fundamentals of Computer Science, such as how data is stored and represented in different parts of the computer and for different types of data.	**SLR 1.1 Systems architecture** Main memory including RAM and Cache: • The purpose of the CPU • Inputs and outputs • FDE cycle **SLR 1.2 Memory and storage** • The use and manipulation of Binary numbers • Storage Devices • Data units	**SLR 1.1 Systems architecture** The CPU is responsible for all the processing in every piece of technology, and therefore arguably the most integral part of the computer. Von Neumann and his architecture had a ground-breaking influence in the world of Computer Science. The changes to the way data and programs were stored meant that the industry could move on in leaps and bounds. **SLR 1.2 Memory and storage** This topic is arguably the foundation for every other topic in Computer Science. It forms the basic principles that relate to networks, programming, security and logic. It provides the necessary understanding you will need in order to make sense of how computers work overall.	
	Topics	Assessment	Know	Know How
	• Introduction to the course • SLR 1.1 Systems architecture ○ 6 lessons • SLR 1.2 Memory and storage – Part 1 ○ 5 lessons • Plus 5 dedicated programming lessons	SLR 1.1 Student workbook SLR 1.1 End-of-topic test	**SLR 1.1 Systems architecture** - What is the purpose of the CPU? - What actions occur at each stage of the fetch-decode=execute cycle. - The role of each component of the CPU. - The purpose of each register and what it stores. - How the common characteristics such as clock speed, cache and number of cores affect the performance of a CPU. - Purpose and examples of embedded systems.	**SLR 1.1 Systems architecture** - Using the simulator Little man computer can help understand the flow of data inside the CPU - To interpret the technical specifications of a CPU and how different changes will affect performance. - To draw a diagram which represents the Von Neuman Architecture which shows how the CPU works and communicates with the memory to execute instructions.
	Topics	Assessment	Know	Know How
2	• SLR 1.2 Memory and storage – Part 1 ○ 2 lessons • SLR 1.2 Memory and storage (Part 2) ○ 12 lessons • Plus 6 dedicated programming lessons	SLR 1.2 Student workbook – Part 1 SLR 1.2 End-of-topic test – Part 1 SLR 1.2 Student workbook (Part 2) SLR 1.2 End-of-topic test (Part 2)	**SLR 1.2 Memory and storage** - Why there is a need for primary storage - The difference between RAM and ROM - The purpose of ROM in a computer system - The purpose of RAM in a computer system - How virtual memory works - The use of secondary storage - Describe common types of storage - Explain the characteristics of different types of storage - How data such as numbers, characters, sound and images are stored in a computer - How character sets are logically ordered. - The two different types of compression.	**SLR 1.2 Memory and storage** - To Identify suitable secondary storage needed for given scenarios - To calculate required storage capacity for a given set of files. - To Calculate the file sizes of sound, image and text files: - *Sound file size = sample rate x duration (s) x bit depth* - *Image file size = colour depth x image height (px) x image width (px)* - *Text file size = bits per character x number of characters* - To convert denary values to binary digits

			- Advantages and disadvantages of compressing files.	- To convert denary values to hexadecimal values and vice versa - To convert binary digits to denary - To convert binary digits to hexadecimal and vice versa - To calculate the resolution of an image.
	Overview	**Building On**	**Cultural Capital**	
3	**SLR 1.3 Computer networks, connections and protocols** **(a)** Students will learn the purpose of function of Computer Networks, from small LAN Networks to the largest Network we use - the Internet. This includes an understanding of the different hardware used and the functions of different topologies used today. **(b)** Students then look at the different protocols (or rules) that are followed at different levels within the Network layers to enable error-free communication.	**SLR 1.3 Computer networks, connections and protocols** • Storage units • Data representation • Compression • Using computers (sending emails) • Connection methods such as Wi-Fi • Intro to Networks in Year 8	**SLR 1.3 Computer networks, connections and protocols** We use and rely on Computer Networks every day. In all industries they are used as a way of communicating and sharing information and data. Networking is used by businesses worldwide in order to share ideas and collaborate, in addition to setting up new IT infrastructure and accessing essential platforms. Many competent individuals are required to support this ever-developing network.	
	Topics	**Assessment**	**Know**	**Know How**
	• SLR 1.3 Computer networks, connections and protocols ○ 12 lessons • Plus 3 dedicated programming lessons	Online MCQ	**SLR 1.3 Computer networks, connections and protocols (a)** - The purpose & types of Networks. - The factor that affect network performance. - The difference between Client Server & Peer to Peer networks. - The networking hardware required. - How the Internet functions. - The different types of Network Topologies. - Wired v Wireless connectivity. **SLR 1.3 Computer networks, connections and protocols (b)** - Common communication protocols. - Data encryption. - Different network layers. - The role of IP & MAC addressing.	**SLR 1.3 Computer networks, connections and protocols (a)** - To comparing the pros and cons of different types of Network - To consider different factors when designing a Network. **SLR 1.3 Computer networks, connections and protocols (b)** - To determine the difference between network connection methods and suggest advantages and disadvantages of connection types - To recommend the most appropriate methods of connection.
4	• SLR 1.3 Computer networks, connections and protocols ○ 2 lessons • SLR 1.4 Network security ○ 10 lessons • Plus 3 dedicated programming lessons	SLR 1.3 Student workbook SLR 1.3 End-of-topic test		
	Overview	**Building On**	**Cultural Capital**	
	SLR 1.4 Network Security The Network Security unit teaches about the threats that are posed to computer systems and networks. Students will then investigate ways to prevent attacks and mitigate damages received by systems. **SLR 1.5 System software** System Software delves into the built in programs found on computer systems so that	**SLR 1.4 Network Security** • Cyber Crime & Security • Legislation • Protection of personal data • User Management **SLR 1.5 System software** • Memory and RAM	**SLR 1.4 Network Security** Cyber security is a major issue in the world today, with many new developments happening every year that Computer Scientists need to stay ahead of. **SLR 1.5 System software** Everyone uses Systems software in some form or another. Understanding how it works and why it's needed allows us to see the links between the software we use and the hardware all around us.	

	students may develop an understanding of their purpose and function. As part of this they will explore operating systems and utility programs.	• Peripheral devices • Data compression and encryption • Magnetic Storage devices		
	Topics	**Assessment**	**Know**	**Know How**
5	• SLR 1.4 Network Security ○ 2 lessons • SLR 1.5 System software ○ 6 lessons • Plus 5 dedicated programming lessons	SLR 1.4 Student workbook SLR 1.4 End-of-topic test SLR 1.5 Student workbook SLR 1.5 End-of-topic test	**SLR 1.4 Network Security** - Forms of Attacks. - Social Engineering & SQL Injection. - Common attack prevention methods. - The reasons for Firewalls and anti-malware software. - The need for user access levels. **SLR 1.5 System software** - The Purpose of systems software. - Memory management. - The Purpose and functionality of utility software. - The functions of operating systems. - File and peripheral management software. - Encryption software. - Types of user interface. - User management & Defragmentation.	**SLR 1.4 Network Security** - To identify and prevent vulnerabilities - To identify appropriate security methods to use in different situations - To use some basic SQL statements. **SLR 1.5 System software** - To identify the features of different User Interfaces - To apply the appropriate utility to a given scenario.
	Overview	**Building On**	**Cultural Capital**	
6	**SLR 1.6 Ethical, legal, cultural and environmental concerns** Within this unit students will learn how the developing use of technology can cause issues in the wider society. They will discuss Ethical, Legal, Cultural, Environmental and Privacy issues, and the laws that are associated	**SLR 1.6 Ethical, legal, cultural and environmental concerns** • Cybercrime and security • Legislation and law relating to hacking and computer misuse • Implications of using automation	**SLR 1.6 Ethical, legal, cultural and environmental concerns** This unit looks at the impact of technology on the world around us and allows us to consider both positives and negatives on all associated stakeholders. It ensures that we are forward thinking in a world where technology is ever changing.	
	Topics	**Assessment**	**Know**	**Know How**
	• SLR 1.6 Ethical, legal, cultural and environmental concerns ○ 9 lessons • 8 lesson text-based adventure game	SLR 1.6 Student workbook SLR 1.6 End-of-topic test	**SLR 1.6 Ethical, legal, cultural and environmental concerns** - Impact of technology on wider society - Ethical & Cultural Issues - Environmental & Privacy Issues	**SLR 1.6 Ethical, legal, cultural and environmental concerns** - To consider the positive and negative impacts of technology on society and its stakeholders. - To recommend a type of license for a given scenario including the benefits and drawbacks.

			- Legislation (Data Protection Act, Computer Misuse, Copyright Law & Software Licencing)	

YEAR 11 OVERVIEW

Term	Overview	Building On	Cultural Capital	
1	**SLR 2.1 Algorithms** Reading, understanding and changing algorithms are an essential Computer Science skill. Students will learn in this unit how to abstract key information out of problems, decompose problems into smaller sections and think "like a computer". They will learn to express algorithms in several ways such as, Flowcharts and Pseudocode. **SLR 2.2 Programming fundamentals** Programming Fundamentals builds on the introductory programming unit, first introduced in year 8. Students will explore Programming constructs, in addition to several key Python functions. This unit has spanned the entire course, alongside the learning of theory topics.	**SLR 2.1 Algorithms** • Computational thinking • Algorithms and flowcharts • Programming constructs • Data types **SLR 2.2 Programming fundamentals** • Use of an interpreter • Mathematical Operators • Computational and logical thinking	**SLR 2.1 Algorithms** The ability to think computationally is an important and useful life skill that can be transferred to many different situations and sectors. Problem solving provides opportunities to develop and improve solutions to make them more efficient. **SLR 2.2 Programming fundamentals** Learning to program is arguably one of life's most valuable skills. It enables students to see every day processes in a different light and enables them to question the way that the technology around us is designed and developed. The skills are transferable to many other careers, such as engineering and Mathematics.	
	Topics	**Assessment**	**Know**	**Know How**
	• SLR 2.2 Programming fundamentals ○ 9 lessons • SLR 2.1 Algorithms ○ 5 lessons • Plus 3 paper 2 exam revision lessons	SLR 2.2 Student workbook SLR 2.2 End-of-topic test	**SLR 2.1 Algorithms** - Abstraction - Decomposition - Algorithmic thinking - Algorithm design - Trace Tables - Common searching and sorting algorithms **SLR 2.2 Programming fundamentals** - Iteration; condition controlled and count controlled - Exponentiation - The use of Arrays	**SLR 2.1 Algorithms** - To identifying inputs, process and outputs - To create, interpret and correct given algorithms. **SLR 2.2 Programming fundamentals** - To identify For and While loops - To Interpret and adapt algorithms - To use a wide range of programming techniques to independently write programs for a range of purposes.

			- Using sub programs (functions and procedures) - Variables Vs Constants - Global and local variables - String manipulation techniques - Boolean operators - The use of records and SQL	
	Topics	Assessment	Know	Know How
2	• SLR 2.1 Algorithms ○ 13 lessons • Plus 7 paper 2 exam revision lessons	SLR 2.1 Student workbook SLR 2.1 End-of-topic test	**SLR 2.1 Algorithms** - Abstraction - Decomposition - Algorithmic thinking - Algorithm design - Trace Tables - Common searching and sorting algorithms	**SLR 2.1 Algorithms** - To identifying inputs, process and outputs - To create, interpret and correct given algorithms.
	Overview	Building On	Cultural Capital	
3	**SLR 2.3 Producing robust programs** This unit takes programming up a notch! Students will learn how to anticipate misuse within their programs, and implement input validation and sanitisation in order to improve their reliability. They will practice different methods of testing to ensure their programs are fully operational and are free of errors.	**SLR 2.3 Producing robust programs** • All previous knowledge of writing programs • Algorithms • Debugging programs • Testing programs with data • Data validity and reliability • Input methods • Use of commenting and indentation	**SLR 2.3 Producing robust programs** We use and rely on Computer Networks every day. In all industries they are used as a way of communicating and sharing information and data. Networking is used by businesses worldwide in order to share ideas and collaborate, in addition to setting up new IT infrastructure and accessing essential platforms. Many competent individuals are required to support this ever-developing network.	
	Topics	Assessment	Know	Know How
	• SLR 2.3 Producing robust programs ○ 8 lessons • Plus 7 paper 2 exam revision lessons	SLR 2.3 Student workbook SLR 2.3 End-of-topic test	**SLR 2.3 Producing robust programs** - Defensive design considerations; anticipating misuse and authentication - Maintainability such as use of sub programs; naming conventions, indentation and commenting - Types of testing; iterative and terminal - Using suitable test data - Identifying syntax and logic errors - Refining algorithms - Input validation - Purpose of testing.	**SLR 2.3 Producing robust programs** - To spot issues and predict program outcomes - To suggest test data to enter into programs - To understand which parts of a program will need to be explained by the use of comments.
	Overview	Building On	Cultural Capital	
4	**SLR 2.4 Boolean logic**	**SLR 2.4 Boolean logic**	**SLR 2.4 Boolean logic**	

	Students will be introduced to several logic gates found inside computer systems. They will then investigate how logic is used within computers in order to solve problems through the use of logic diagrams and truth tables. **SLR 2.5 Programming languages and IDEs** This unit explores the different generations of programming languages and the differences between them. It also teaches the different tools that are integrated within development environments in order to support programmers when creating code.	• Boolean operators • Datatypes **SLR 2.5 Programming languages and IDEs** - Knowledge of Python as a high level language - The use of Interpreters and compilers when writing programs -Some basic features of IDEs	Logic gates which are built into electronic circuits are used in every piece of technology; from computer games to home security systems. They are an integral part of how circuits work. **SLR 2.5 Programming languages and IDEs** Being able to use a range of different IDEs when writing programs is a useful skill as students may come across a variety of platforms. They may also go on to learn other languages so having an understanding of the features of high level languages would be an advantage.	
	Topics	**Assessment**	**Know**	**Know How**
	• SLR 2.4 Boolean logic ○ 5 lessons • SLR 2.5 Programming languages and IDEs ○ 6 lessons • Plus 4 paper 2 exam revision lessons	SLR 2.4 Student workbook SLR 2.4 End-of-topic test SLR 2.5 Student workbook SLR 2.5 End-of-topic test	**SLR 2.4 Boolean logic** - Logic gates AND, OR and NOT - Truth tables - Understanding logic diagrams - Combining gates to solve problems **SLR 2.5 Programming languages and IDEs** - Characteristics and purpose of different levels of programming language - Purpose of translators - Compilers and interpreters - Tools and facilities available in an IDE such as: editors, error diagnostics, run-time environment, translators.	**SLR 2.4 Boolean logic** - To create, complete or edit logic diagrams and truth tables for given scenarios - To work with more than one gate in a logic diagram. **SLR 2.5 Programming languages and IDEs** - To use a range of IDEs along with its tools and features - To use the error checking facility when debugging programs -The basic use of assembly code enables us to write programs.
	Topics	**Assessment**		
5	• Further programming experience with the exam revision unit. • Additional time to complete the programming challenges and super challenges presented throughout the SLR theory units and from the dedicated programming resources.	Past papers Smart Revise app Seneca		