

Grace Academy Coventry

Curriculum Map 2024/25 - GCSE Computer Science - Year 10

DATES	AUTUMN TERM 1	AUTUMN TERM 2	SPRING TERM 1	SPRING TERM 2	SUMMER TERM 1	SUMMER TERM 2
Focus	1.1 SYSTEMS ARCHITECTURE	1.2 MEMORY AND STORAGE	1.3 Ethical, legal, cultural and environmental impacts of digital technology	1.4 - Programming fundamentals 2.5 - Boolean logic	1.4 - Programming fundamentals 2.5 - Boolean logic	1.4 - Python Programming Project with Trinket
Key Knowledge	1.1.1 Architecture of the CPU 1.1.2 CPU Performance 1.1.3 Embedded systems	1.2.1 Primary storage (memory) 1.2.2 Secondary storage 1.2.3 Units 1.2.4 Data storage 1.2.5 Compression	1.3.1 Ethical, legal, cultural and environmental impact Data Protection Act Computer misuse act Copyrights designs and patents act Creative commons	1.4.1 Programming fundamentals 1.4.2 Data types 1.4.3 Additional programming techniques 5.1.1 Boolean logic	1.4.1 Programming fundamentals 1.4.2 Data types 1.4.3 Additional programming techniques 5.1.1 Boolean logic	1.4.1 Programming fundamentals 1.4.2 Data types 1.4.3 Additional programming techniques 5.1.1 Boolean logic
Key Skills	To understand how the CPU works. In particular the Fetch decode Execute cycle, Von Newman Architecture, and how hardware works in unison with software.	To understand the difference between primary storage (main memory) and secondary storage. To learn how memory and storage is measured, where data is stored and how files are converted for storage.	To discuss and debate the ethical, legal and environmental impacts of technology. To learn the legislation that must be followed in an online environment.	To understand the use of variables, constants, operators, inputs, outputs and assignments To learn and use the three basic programming constructs used to control the flow of a program:	To understand the use of variables, constants, operators, inputs, outputs and assignments To learn and use the three basic programming constructs used to control the flow of a program:	To understand the use of variables, constants, operators, inputs, outputs and assignments To learn and use the three basic programming constructs used to

				<ul style="list-style-type: none"> o Sequence o Selection o Iteration (count- and condition-controlled loops) <p>To understand the following additional programming techniques:</p> <ul style="list-style-type: none"> o File handling o SQL to search o Arrays <p>To understand the simple logic diagrams using the operators AND, OR and NOT as well as their truth tables.</p> <p>Combining Boolean operators using AND, OR and NOT.</p> <p>Applying logical operators in truth tables to solve problems.</p>	<ul style="list-style-type: none"> o Sequence o Selection o Iteration (count- and condition-controlled loops) <p>To understand the following additional programming techniques:</p> <ul style="list-style-type: none"> o File handling o SQL to search o Arrays <p>To understand the simple logic diagrams using the operators AND, OR and NOT as well as their truth tables.</p> <p>Combining Boolean operators using AND, OR and NOT.</p> <p>Applying logical operators in truth tables to solve problems.</p>	<p>control the flow of a program:</p> <ul style="list-style-type: none"> o Sequence o Selection o Iteration (count- and condition-controlled loops) <p>To understand the following additional programming techniques:</p> <ul style="list-style-type: none"> o File handling o SQL to search o Arrays <p>To understand the simple logic diagrams using the operators AND, OR and NOT as well as their truth tables.</p> <p>Combining Boolean operators using AND, OR and NOT.</p> <p>Applying logical operators in truth tables to solve problems.</p>
Individual Tasks/Assessment	• Assessment Points	• Assessment Points	• Assessment Points	• Assessment Points	• Assessment Points	• Assessment Points

	<ul style="list-style-type: none"> Exam questions Multiple-choice questions Short-answer questions 	<ul style="list-style-type: none"> Exam questions Multiple-choice questions Short-answer questions 	<ul style="list-style-type: none"> Exam questions Multiple-choice questions Short-answer questions 	<ul style="list-style-type: none"> Exam questions Multiple-choice questions Short-answer questions Extended writing tasks 	<ul style="list-style-type: none"> Exam questions Multiple-choice questions Short-answer questions Extended writing tasks 	<ul style="list-style-type: none"> Exam questions Multiple-choice questions Short-answer questions Extended writing tasks
Links to KS2/KS3	Computer systems (Y8) Memory and storage (Y8) Data representation (Y9)	Memory and storage (Y8) Data representation (Y9)	Safety and Responsibility (Y7)	Edublocks (Y7) Python Basics (Y8) Python Programming (Y9)	Edublocks (Y7) Python Basics (Y8) Python Programming (Y9)	Edublocks (Y7) Python Basics (Y8) Python Programming (Y9)
End Point	<p>Understand what actions occur at each stage of the fetch-execute cycle.</p> <p>The role/purpose of each component and what it manages.</p> <p>The purpose of each register, what it stores (data or address).</p> <p>The effects of changing any of the common characteristics on system performance, either</p>	<p>Understanding the need of primary and secondary storage in a computer system.</p> <p>The purpose of virtual memory.</p> <p>The need for data to be stored in a binary format.</p> <p>To understand how file sizes are calculated and then converted into binary.</p>	<p>Knowledge of a variety of examples of digital technology and how this impacts on society</p> <p>An ability to discuss the impact of technology based around the issues listed</p> <p>The purpose of each piece of legislation and the specific actions it allows or prohibits</p> <p>Open source vs proprietary licences.</p>	<p>Practical use of the techniques in a high-level language within the classroom.</p> <p>Understanding of each technique.</p> <p>Practical use of the data types in a high-level language within the classroom.</p> <p>Practical use of the additional programming techniques in a high-level language within the classroom.</p>	<p>Practical use of the techniques in a high-level language within the classroom.</p> <p>Understanding of each technique.</p> <p>Practical use of the data types in a high-level language within the classroom.</p> <p>Practical use of the additional programming techniques in a high-level language within the classroom.</p>	<p>Practical use of the techniques in a high-level language within the classroom.</p> <p>Understanding of each technique.</p> <p>Practical use of the data types in a high-level language within the classroom.</p> <p>Practical use of the additional programming techniques in a high-level language within the classroom.</p>

	individually or in combination.			<p>Knowledge of the truth tables for each logic gate.</p> <p>Understanding of how to create, complete or edit logic diagrams and truth tables for given scenarios.</p> <p>Ability to work with more than one gate in a logic diagram.</p> <p>Recognition of each gate symbol.</p>	<p>Knowledge of the truth tables for each logic gate.</p> <p>Understanding of how to create, complete or edit logic diagrams and truth tables for given scenarios.</p> <p>Ability to work with more than one gate in a logic diagram.</p> <p>Recognition of each gate symbol.</p>	<p>Knowledge of the truth tables for each logic gate.</p> <p>Understanding of how to create, complete or edit logic diagrams and truth tables for given scenarios.</p> <p>Ability to work with more than one gate in a logic diagram.</p> <p>Recognition of each gate symbol.</p>
--	---------------------------------	--	--	---	---	---

Grace Academy Coventry Curriculum Map 2025/26 - GCSE Computer Science - Year 11

DATES	AUTUMN TERM 1	AUTUMN TERM 2	SPRING TERM 1	SPRING TERM 2	SUMMER TERM 1	
Focus	2.1 COMPUTER NETWORKS, CONNECTIONS AND PROTOCOLS	2.2 Network security 2.3 Systems software 2.3.1 Operating systems 2.3.2 Utility software	2.3 - Algorithms	2.4 - Programming languages and Integrated Development Environments 2.5 - Producing robust programs	Revision & Terminal Examinations	
Key Knowledge	2.1.1 Networks and topologies 2.1.2 Wired and wireless networks, protocols and layers	2.2.1 Threats to computer systems and networks 2.2.2 Identifying and preventing vulnerabilities	2.3.1 Computational thinking 2.3.2 Designing, creating and refining algorithms 2.3.3 Searching and sorting algorithms	2.4.1 Languages 2.4.2 The Integrated Development Environment (IDE) 2.5.1 Defensive design 2.5.2 Testing	Revision & Terminal Examinations	
Key Skills	To understand the purpose of a network and the characteristics of each type of network topology. Understanding the difference between a	The social engineering methods that are used by cybercriminals to steal data. To understand the different types of malware that exist and	To understand the principles of computational thinking: o Abstraction o Decomposition o Algorithmic thinking	To understand the characteristics and purpose of different levels of programming language: o High-level languages	Revision & Terminal Examinations	

	<p>wired and wireless network, the hardware used, the protocols that are followed and the layers that make up a network.</p>	<p>how to find vulnerabilities/ protect against these attacks.</p>	<p>To create, interpret, correct, complete, and refine algorithms using:</p> <ul style="list-style-type: none"> o Pseudocode o Flowcharts o Reference language/high-level programming language <p>To learn and use the process of searching and sorting algorithms.</p>	<p>o Low-level languages</p> <p>To learn the purpose of translators and the characteristics of compilers/interpreters.</p> <p>Learn the common tools and facilities available in an Integrated Development Environment (IDE).</p> <p>To understand the principles of defensive design by anticipating misuse, using input validation and providing authentication when creating programs.</p> <p>To ensure the maintainability of programs through the use of sub programs, naming conventions, indentations and commenting.</p> <p>The importance of iterative and final testing.</p>		
--	--	--	--	--	--	--

Individual Tasks/Assessment	<ul style="list-style-type: none"> • Assessment Points • Exam questions • Multiple-choice questions • Short-answer questions • Extended writing tasks 	<ul style="list-style-type: none"> • Assessment Points • Exam questions • Multiple-choice questions • Short-answer questions 	<ul style="list-style-type: none"> • Assessment Points • Exam questions • Multiple-choice questions • Short-answer questions • Extended writing tasks 	<ul style="list-style-type: none"> • Assessment Points • Exam questions • Multiple-choice questions • Short-answer questions • Extended writing tasks 	Revision & Terminal Examinations	
Links to KS2/KS3	<p>Networks (y7) Cybersecurity (Y9)</p>	<p>Cybersecurity (Y9)</p>	<p>Computational thinking (Y7) Algorithms (Y8)</p>	<p>Debugging (Y8) Edublocks (Y7) Python Basics (Y8)</p>		

<p>End Point</p>	<p>Understand the characteristics of LANs and WANs including common Understanding of different factors that can affect the performance</p> <p>Understand the concept of the Internet as a network of computer networks</p> <p>Compare benefits and drawbacks of wired versus wireless</p>	<p>Threats posed to devices/systems Knowledge/principles of each form of attack Understanding of methods to remove vulnerabilities</p> <p>Knowledge/principles of each prevention method:</p> <p>Understand what each function of an operating system does</p> <p>User management functions, e.g.: Allocation of an account Access rights Security, etc.</p> <p>File management, and the key features, e.g.: Naming Allocating to folders Moving files Saving, etc.</p> <p>Purpose of the identified utility software and why it is required</p>	<p>Produce simple diagrams to show: The structure of a problem Subsections and their links to other subsections.</p> <p>Understand the main steps of each algorithm.</p>	<p>The differences between high- and low-level programming languages.</p> <p>The need for translators</p> <p>The differences, benefits and drawbacks of using a compiler or an interpreter.</p> <p>Understanding of the issues a programmer should consider to ensure that a program caters for all likely input values.</p> <p>The difference between testing modules of a program during development and testing the program at the end of production.</p> <p>Understanding of how to deal with invalid data in a program.</p>		
-------------------------	---	--	--	--	--	--