Grace Academy Coventry Curriculum Map 2023/24 – 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 COMPUTER NETWORKS, CONNECTIONS AND PROTOCOLS	1.4 Network security	1.5 Systems software	1.6 Ethical, legal, cultural and environmental impacts of digital technology
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 Networks and topologies 1.3.2 Wired and wireless networks, protocols and layers	1.4.1 Threats to computer systems and networks 1.4.2 Identifying and preventing vulnerabilities	1.5.1 Operating systems 1.5.2 Utility software	1.6.1 Ethical, legal, cultural and environmental impact Data Protection Act Computer misuse act Copyrights designs and patents act Creative commons
Key Skills	To understand how the CPU works. In particular the Fetch decode	To understand the difference between primary storage (main	To understand the purpose of a network and the characteristics of	The social engineering methods that are used by cybercriminals to steal data.	To understand the purpose of an operating system within a computer.	To discuss and debate the ethical, legal and environmental impacts of technology.

	Execute cycle, Von Newman Architecture, and how hardware works in unisor with software.	memory) and secondary storage. To learn how memory and storage is measured where data is stored and how files are converted for storage.	each type of network topology. Understanding the difference between a wired and wireless network, the hardware used, the protocols that are followed and the layers that make up a network.	To understand the different types of malware that exist and how to find vulnerabilities/ protect against these attacks.	To learn the difference between utility software and application software.	To learn the legislation that must be followed in an online environment.
Individual Tasks/Assessment	End of Topic Exam Questions Multiple- choice questions Short-answer questions	End of Topic Exam Questions Multiple- choice questions Short-answer questions	End of Topic Exam Questions Multiple- choice questions Short- answer questions	End of Topic Exam Questions Multiple- choice questions Short-answer questions	End of Topic Exam Questions Multiple- choice questions Short-answer questions	End of Topic 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)	Networks (y7) Cybersecurity (Y9)	Networks (y7) Cybersecurity (Y9)	Computer Systems Y8 Memory and storage (Y8)	Safety and Responsibility (Y7)
End Point	Understand what actions occur at each stage of the fetch- execute cycle.	Understanding the need of primary and secondary storage in a computer system.	Understand the characteristics of LANs and WANs including common Understanding of	Threats posed to devices/systems Knowledge/principles of each form of attack Understanding of	Understand what each function of an operating system does User management	Knowledge of a variety of examples of digital technology and how this impacts on

The rol each ca what it The pur register (data o The eff changin common on syster perforn individu	le/purpose of omponent and manages. rpose of each r, what it stores or address). fects of g any of the characteristics rem nance, either ually or in ation.	The purpose of virtual memory. The need for data to be stored in a binary format. To understand how file sizes are calculated and then converted into binary.	different factors that can affect the performance Understand the concept of the Internet as a network of computer networks Compare benefits and drawbacks of wired versus wireless	methods to remove vulnerabilities Knowledge/principles of each prevention method:	functions, e.g.: Allocation of an account Access rights Security, etc. File management, and the key features, e.g.: Naming Allocating to folders Moving files Saving, etc. Purpose of the identified utility	society An ability to discuss the impact of technology based around the issues listed The purpose of each piece of legislation and the specific actions it allows or prohibits Open source vs
---	---	---	---	---	---	--

Grace Academy Coventry Curriculum Map 2023/24 – GCSE Computer Science – Year 11

DATES	AUTUMN TERM 1	AUTUMN TERM 2	SPRING TERM 1	SPRING TERM 2	SUMMER TERM 1
Focus	2.1 - Algorithms	2.2 - Programming fundamentals	2.3 - Producing robust programs	2.4 - Boolean logic	2.5 - Programming languages and Integrated Development Environments
Key Knowledge	2.1.1 Computational thinking2.1.2 Designing, creating and refining algorithms2.1.3 Searching and sorting algorithms	2.2.1 Programming fundamentals 2.2.2 Data types 2.2.3 Additional programming techniques	2.3.1 Defensive design 2.3.2 Testing	2.4.1 Boolean logic	2.5.1 Languages 2.5.2 The Integrated Development Environment (IDE)

Key Skills	To understand the principles of computational thinking: o Abstraction o Decomposition o Algorithmic thinking To create, interpret, correct, complete, and refine algorithms using: o Pseudocode o Flowcharts o Reference language/high-level programming language To learn and use the process of searching and sorting algorithms.	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: o Sequence o Selection o Iteration (count- and condition-controlled loops) To understand the following additional programming techniquess o File handling o SQL to search o Arrays	To understand the principles of defensive design by anticipating misuse, using input validation and providing authentication when creating programs. To ensure the maintainability of programs through the use of sub programs, naming conventions, indentations and commenting. The importance of iterative and final testing.	To understand the simple logic diagrams using the operators AND, OR and NOT as well as their truth tables. Combining Boolean operators using AND, OR and NOT. Applying logical operators in truth tables to solve problems.	To understand the characteristics and purpose of different levels of programming language: o High-level languages o Low-level languages To learn the purpose of translators and the characteristics of scompilers/interpreters. Learn the common tools and facilities available in an Integrated Development Environment (IDE).
Individual Tasks/Assessment	End of Topic Exam Questions Multiple- choice auestions	End of Topic Exam Questions Multiple- choice questions	End of Topic Exam Questions Multiple- choice guestions Short-	End of Topic Exam Questions Multiple- choice guestions	End of Topic Exam Questions Multiple- choice auestions
	Short-answer questions Extended writing tasks	Short-answer questions Extended writing tasks	answer questions Extended writing tasks	Short-answer questions Extended writing tasks	Short-answer questions Extended writing tasks

Links to KS2/KS3	Computational thinking	Edublocks (Y7)	Debugging (Y8)	Memory and storage	Edublocks (Y7)
	(Y7)	Python Basics (Y8)	Python Basics (Y8)	(Y8)	Python Basics (Y8)
	Algorithms (Y7)	Python Programming (Y9)	Python Programming (Y9)	Number systems (Y9)	Python Programming (Y9)
End Point	Understanding of these principles and how they are used to define and refine problems. Produce simple diagrams to show: The structure of a problem Subsections and their links to other subsections. Understand the main steps of each algorithm.	Practical use of the techniques in a high- level language within the classroom. Understanding of each technique. Practical use of the data types in a high- level language within the classroom. Practical use of the additional programming techniques in a high- level language within the classroom.	Understanding of the issues a programmer should consider to ensure that a program caters for all likely input values. The difference between testing modules of a program during development and testing the program at the end of production. Understanding of how to deal with invalid data in a program.	Knowledge of the truth tables for each logic gate. Understanding of how to create, complete or edit logic diagrams and truth tables for given scenarios. Ability to work with more than one gate in a logic diagram. Recognition of each gate symbol.	The differences between high- and low- level programming languages. The need for translators The differences, benefits and drawbacks of using a compiler or an interpreter.