## The Grange Academy Computer Science curriculum map



## Intent

The core of computing is computer science, in which pupils are taught the principles of information and computation, how digital systems work, and how to put this knowledge to use through programming. Buildings on this knowledge and understanding, pupils are equipped to use information technology to create programs, systems and a range of content. Key Skills - Computer Science also ensures that pupils become digitally literate – driven, and express themselves and develop their ideas through identifying issues and problem solving at a level suitable for the future workplace and as active participants in a digital world.

## Implementation

	Autumn Term I	Autumn Term II	Spring Term I	Spring Term II	Summer Term I	Summer Term II
Year 7	Unit 1: Understanding Computers  - E-Safety - Elements of a	Unit 1: Understanding Computers  - ASCII - Binary addition - Storage devices - Revision	Unit 1: Understanding Computers  - E-Safety - Elements of a Computer System - The CPU - Understanding binary - Convert Binary to Denary	Unit 1: Understanding Computers  - ASCII - Binary addition - Storage devices - Revision	Unit 1: Understanding Computers  - E-Safety - Elements of a Computer System - The CPU - Understanding binary - Convert Binary to Denary	Unit 1: Understanding Computers  - ASCII - Binary addition - Storage devices
Year 8	Unit 1: Understanding Computers  - E-Safety - Elements of a	- The internet - Connectivity - Topologies - Client-server networks - Encryption - Revision	Unit 1: Understanding Computers  - E-Safety - Elements of a	Unit 2: Network  - The internet - Connectivity - Topologies - Client-server networks - Encryption - Revision	Unit 1: Understanding Computers  - E-Safety - Elements of a Computer System - The CPU - Understanding binary - Convert Binary to Denary - ASCII - Binary addition - Storage devices	Unit 2: Network  - The internet - Connectivity - Topologies - Client-server networks - Encryption - Revision





	1		Γ		Γ	
Year	Flow Charts & Binary	Computer Systems &	Inside a Computer System	Programming 1	Programming 2	Programming 3
9	Flow Charts & Billary	Legislations	Inside a Computer System	Flogialililing 1	Frogramming 2	riogiailillilig 3
9	This unit will cover an	Legislations	This unit will cover an	This unit will cover an	This unit will cover an	This unit will cover an
	introduction to the core	This unit will cover an	introduction to the core	introduction to the	introduction to the core	introduction to the
	content that is covered in	introduction to the	content that is covered in	core content that is	content that is covered in	core content that is
	the Computer Science	core content that is	the Computer Science	covered in the	the Computer Science	covered in the
	GCSE. Flow Charts	covered in the	GCSE.	Computer Science	GCSE.	Computer Science
	Binary	Computer Science	GCSL.	GCSE.	GCSL.	GCSE.
	Biriary	GCSE.	Networks Logic Gates	GCJL.	Iteration	GCJL.
	Flow charts	Algorithms Hardware	Malware System Security	ROM & RAM	User Input	Subroutines
	Binary Code	Software Ethics	Von Neumann	Print/String Variables	Lists	String Manipulation
	Pseudo Code	Software Ethics	Von Neamann	Selection	File Handling	Operators
	Recalling Information	<ul><li>Problem Solving ●</li></ul>	Networks	Sciection	The Handing	Casting
	Problem Solving ●	Pseudo Code	• Internet	Computer Systems	• Iteration	Practical Python
	Group work	Computer Parts &	Logic gates	• ROM	• Loops	Challenge
	Organisation	Components	Calculations	• RAM	User input	Chancinge
	- Organisation	Recalling Information	Malware	Programming	Programming	• Subroutines
	Students will have a topic	Networks	◆ Virus Protection ◆	Writing Code	Writing Code	Writing string
	test at the end of the half	Group work	Security	Reading Code	Reading code	<ul> <li>Manipulating strings</li> </ul>
	term. This will assess their	<ul><li>Organisation</li></ul>	Von Neumann	<ul> <li>Writing Variables ●</li> </ul>	Writing and reading	Operations
	knowledge of the topics	- 10-11-11-11	Group Work	Using selection	lists	Casting
	they have learnt so far.	Students will have a	Organisation	9	File handling	Programming
	,	topic test at the end of		Students will have a		Calculations
	This unit links to the GCSE	the half term. This will	Students will have a topic	topic test at the end of	Students will have a topic	Problem Solving
	in computer science. The	assess their knowledge	test at the end of the half	the half term. This will	test at the end of the half	
	content in this unit will	of the topics they have	term. This will assess their	assess their knowledge	term. This will assess their	Students will have a
	help towards both	learnt so far.	knowledge of the topics	of the topics they have	knowledge of the topics	topic test at the end of
	component 1 - Computer		they have learnt so far.	learnt so far.	they have learnt so far.	the half term. This will
	Systems and component	This unit links to the				assess their knowledge
	2_Computational	GCSE in computer	This unit links to the GCSE	This unit links to the	This unit links to the GCSE	of the topics they have
	thinking, algorithms and	science. The content in	in computer science. The	GCSE in computer	in computer science. The	learnt so far.
	programming.	this unit will help	content in this unit will	science. The content in	content in this unit will	
		towards both	help towards both	this unit will help	help towards both	This unit links to the
		component 1 -	component 1 - Computer	towards both	component 1 - Computer	GCSE in computer
		Computer Systems and	Systems and component	component 1 -	Systems and component	science. The content in





		component	2_Computational	Computer Systems and	2_Computational	this unit will help
		2_Computational	thinking, algorithms and	component	thinking, algorithms and	towards both
		thinking, algorithms	programming.	2_Computational	programming.	component 1 -
		and programming.		thinking, algorithms		Computer Systems and
				and programming.		component
						2_Computational
						thinking, algorithms
						and programming.
Year	2.4 Boolean logic 1.2.4	1.2.4 Data storage 1.2.5	1.1.1 Architecture of the	1.1.3 Embedded	1.2.2 Secondary storage	1.3.1 Networks and
10	Data storage 2.1.1	Compression 2.1.2	CPU 1.1.2 CPU	systems 1.2.1 Primary	1.3.1 Networks and	topologies 1.3.2 Wired
	Computational thinking	Designing, creating and	Performance 2.2.2 Data	storage (Memory) 1.2.2	topologies 2.2.3 Additional	and wireless networks,
		refining algorithms	types 2.2.1 Programming	Secondary storage 2.2.1	programming techniques	protocols and layers
	This topic will introduce		fundamentals	Programming		
	to students how step by	In this topic students		fundamentals 2.2.3	Students will continue to	Students will look at
	step instructions are	will continue with their	Students will be	Additional programming	cover secondary storage	network topologies and
	needed. They will also	knowledge on data	introduced to the	techniques	and also look at network	also cover wired and
	learn about data storage	storage and also cover	architecture of the CPU		topologies. They will also	wireless networks.
	and boolean logic.	compression and how	and also look at the	In this topic students	continue to improve	
		to design and create	performance. Along with	will look at embedded	programming techniques.	<ul><li>Networks</li></ul>
	<ul> <li>Boolean logic</li> </ul>	algorithms.	this they will look at	systems along with		• Data
	● Storage		different data types. They	primary and secondary	<ul><li>Storage</li></ul>	<ul><li>◆ Wired Connections ◆</li></ul>
	<ul><li>Calculations</li></ul>	<ul><li>Storage</li></ul>	will also continue to build	storage.They will also	<ul><li>Networking</li></ul>	Wireless Connections
	<ul> <li>Contextualise content</li> </ul>	<ul><li>Compression</li></ul>	on programming	continue to improve	<ul> <li>Contextualise content</li> </ul>	<ul> <li>Contextualise</li> </ul>
	<ul> <li>Computational</li> </ul>	<ul><li>Design</li></ul>	fundamentals.	programming	<ul> <li>Computational</li> </ul>	content
	approach	<ul> <li>Creating Algorithms</li> </ul>		techniques.	approach	<ul> <li>Computational</li> </ul>
	<ul> <li>Clarify significant</li> </ul>	<ul> <li>Contextualise</li> </ul>	● CPU		<ul> <li>Clarify significant</li> </ul>	approach
	information	content	<ul><li>Performanc e of CPU ●</li></ul>	<ul><li>Systems</li></ul>	information	<ul><li>Clarify significant</li></ul>
	<ul> <li>Develop Computing</li> </ul>	<ul> <li>Computational</li> </ul>	Contextuali se content	● Storage	<ul> <li>Develop Computing</li> </ul>	information
	Language skills	approach	<ul> <li>Computatio nal</li> </ul>	Memory	Language skills	<ul> <li>Develop Computing</li> </ul>
	<ul> <li>Problem solving</li> </ul>	<ul> <li>Clarify significant</li> </ul>	approach	<ul> <li>Contextualise</li> </ul>	<ul> <li>Problem solving</li> </ul>	Language skills
	_	information	<ul> <li>Clarify significant</li> </ul>	content ●		<ul><li>Problem solving</li></ul>
	Е	Develop Computing	information	Computational		
		Language skills	Develop Computing	approach		
		<ul><li>Problem solving</li></ul>	Language skills	Clarify significant	Component 1 - Computer	
			<ul><li>Problem solving</li></ul>	information	Systems and component 2	
				Develop Computing	Computational thinking,	
	Component 1 - Computer		Component 1 - Computer	Language skills	algorithms and	Component 1 -
	Systems and component		Systems and component 2	Problem solving	programming	Computer Systems and
	<b>2</b> _Computational		Computational thinking,			component 2



## The Grange Academy Computer Science curriculum map

	thinking, algorithms and programming	Component 1 - Computer Systems and component 2_Computational thinking, algorithms and programming.	algorithms and programming.	Component 1 - Computer Systems and component 2 Computational thinking, algorithms and programming.		Computational thinking, algorithms and programming.
Year 11	Content: J277 OCR - Component 2	Content: J277 OCR - Component 2	Content: J277 OCR - Component 2	Content: J277 OCR - Component 2	Content: J277 OCR - Component 1 & 2	Content: J277 OCR - Component 1 & 2
	<ul><li>2.1 Algorithms</li><li>2.2 Programming</li><li>Fundamentals</li><li>2.3 Producing robust</li><li>programs</li><li>2.4 Boolean logic</li></ul>	2.5 Programming languages and integrated development environments  Practical Programming All students must be given the opportunity to undertake a programming task(s), either to a specification or to solve a problem (or problems). Students may draw on some of the content in both components when engaged in Practical Programming.	1.1 System Architecture 1.2 Memory and Storage 1.3 Computers networks, connections and protocols	1.4 Network Security 1.5 System Software 1.6 Ethical, legal, environmental impacts of digital technology	Component 1 revision Component 2 revision	Exam term where students must practice their practical programming, component 1 and 2.