



Curriculum Map cover sheet

Subject: Computing	Year Group 7	Key Stage 3
Links to prior learning Introduction to school network, basic digital literacy, simple algorithms (building on primary computing skills)	Retrieval Practice Starter activities revisiting prior lesson concepts, mini quizzes, recap of key vocabulary, homework	Adaptive teaching strategies Scaffolded programming tasks, differentiated worksheets, step-by-step instructions for new concepts
Literacy Focus on command words, technical vocabulary, and writing clear instructions and explanations	Oracy Group discussions on e-safety, presenting simple projects, peer feedback	Numeracy Binary number conversion, spreadsheet calculations, data analysis
Careers across the Curriculum Introduction to IT careers, case studies of tech professionals	Wider Curriculum opportunities Links to maths (data handling), science (experiment modelling), and English (digital storytelling)	Super Curriculum Please add link, can't find it
Assessment and feedback Formative assessment given frequently in the form of mini-whiteboard activities, cold-calling, group discussions etc... Summative assessment given once per half term and feedback given on this test and anything of note will be retaught, retrieval starters will be put into future lessons to retrieve knowledge that was a weakness. End of year test summarising all knowledge from the year will be sat in the last half term and again feedback will be given on this assessment		



Curriculum Map cover sheet

Subject: Computing	Year Group 8	Key Stage 3
Links to prior learning • Builds on Year 7 basics: file management, e-safety, introductory programming(scratch), and understanding of hardware/software. • Recaps digital literacy skills and basic algorithms.	Retrieval Practice Starter activities revisiting prior lesson concepts, mini quizzes, recap of key vocabulary, homework	Adaptive teaching strategies • Scaffolded worksheets and differentiated tasks (e.g., challenge tasks for advanced students, step-by-step guides for those needing support). • Use of visual aids (flowcharts, mind maps) and interactive tools.
Literacy • Focus on key computing terminology (e.g., sequence, selection, iteration). • Sentence starters for explaining code and digital concepts. • Reading and summarizing short articles about technology in society.	Oracy • Paired “explain your code” activities. • Group discussions on online safety and responsible use. • Presentations on technology’s impact or digital citizenship.	Numeracy • Binary numbers and simple binary conversions. • Counting loops, understanding variables, and basic logic. • Links to maths through pattern recognition and problem-solving.

Careers across the Curriculum • Introduce roles like web developer, IT technician, and social media manager. • Discuss how computing skills apply in creative industries (e.g., animation, music tech).	Wider Curriculum opportunities • Links with maths (logic, patterns), art (digital graphics), and PSHE (online safety). • Projects such as creating a simple game. • Trip to Bletchley park	Super Curriculum Please add link, can't find it
Assessment and feedback Formative assessment given frequently in the form of mini-whiteboard activities, cold-calling, group discussions etc... Summative assessment given once per half term and feedback given on this test and anything of note will be retaught, retrieval starters will be put into future lessons to retrieve knowledge that was a weakness. End of year test summarising all knowledge from the year will be sat in the last half term and again feedback will be given on this assessment		



Curriculum Map cover sheet

Subject: Computing	Year Group 9	Key Stage 3
Links to prior learning • Builds on Year 8 programming (e.g., Python), understanding of hardware, and digital literacy. • Recaps binary, logic, and safe/ethical use of technology.	Retrieval Practice Starter activities revisiting prior lesson concepts, mini quizzes, recap of key vocabulary, homework	Adaptive teaching strategies • Scaffolded worksheets and differentiated tasks (e.g., challenge tasks for advanced students, step-by-step guides for those needing support). • Use of visual aids (flowcharts, mind maps) and interactive tools. • Extension tasks for high-attaining students (e.g., more complex programming challenges).
Literacy • Emphasis on advanced computing vocabulary (e.g., abstraction, decomposition, protocol). • Writing code comments and technical documentation. • Research and reporting on current	Oracy • Debates on ethical issues (e.g., data privacy, AI in society). • Group presentations on computing innovations. • Paired debugging and code	Numeracy • More advanced binary and hexadecimal operations. • Algorithm efficiency and computational thinking. • Data analysis and interpretation (e.g., spreadsheets, simple

computing issues (e.g., AI, cybersecurity).	explanation sessions.	databases).
Careers across the Curriculum • Explore roles in cybersecurity, software engineering, and data science. • Virtual guest speakers or video interviews with professionals. • Investigate how computing is used in medicine, engineering, and entertainment.	Wider Curriculum opportunities • Links with maths (logic, patterns), art (digital graphics), and PSHE (online safety). • Projects such as creating a simple game. • Trip to Bletchley park • Explore the societal impact of technology (links with PD). • Use computing to support learning in other subjects (e.g., creating revision tools).	Super Curriculum Please add link, can't find it
Assessment and feedback Formative assessment given frequently in the form of mini-whiteboard activities, cold-calling, group discussions etc... Summative assessment given once per half term and feedback given on this test and anything of note will be retaught, retrieval starters will be put into future lessons to retrieve knowledge that was a weakness. End of year test summarising all knowledge from the year will be sat in the last half term and again feedback will be given on this assessment		

Term	Number of weeks				
Autumn first half term	1	Intro to Network/ E-Safety	E-Safety/ Cybersecurity	Python (Recap/ Loops/ Subroutines)	Test
	2				
	3				
	4				
	5				
	6				
	7				
Autumn second half term	1	Digital literacy	Algorithms/ Flowcharts Python (Up to Selection)	Spreadsheets	test
	2				
	3				
	4				
	5				
	6	Scratch			
	7				
Spring first half term	1	Scratch	Networks	Cyber security	test
	2				
	3				
	4				
	5				
	6				
Spring second half term	1	Hardware/Software	Social media marketing	Data Representation	
	2				
	3				
	4				
	5				
Summer first half term	1	Spreadsheets	What is AI?	IOE- Internet of Everything	test
	2				
	3				
	4				
	5				
	6				

[illegible]



Curriculum Map cover sheet

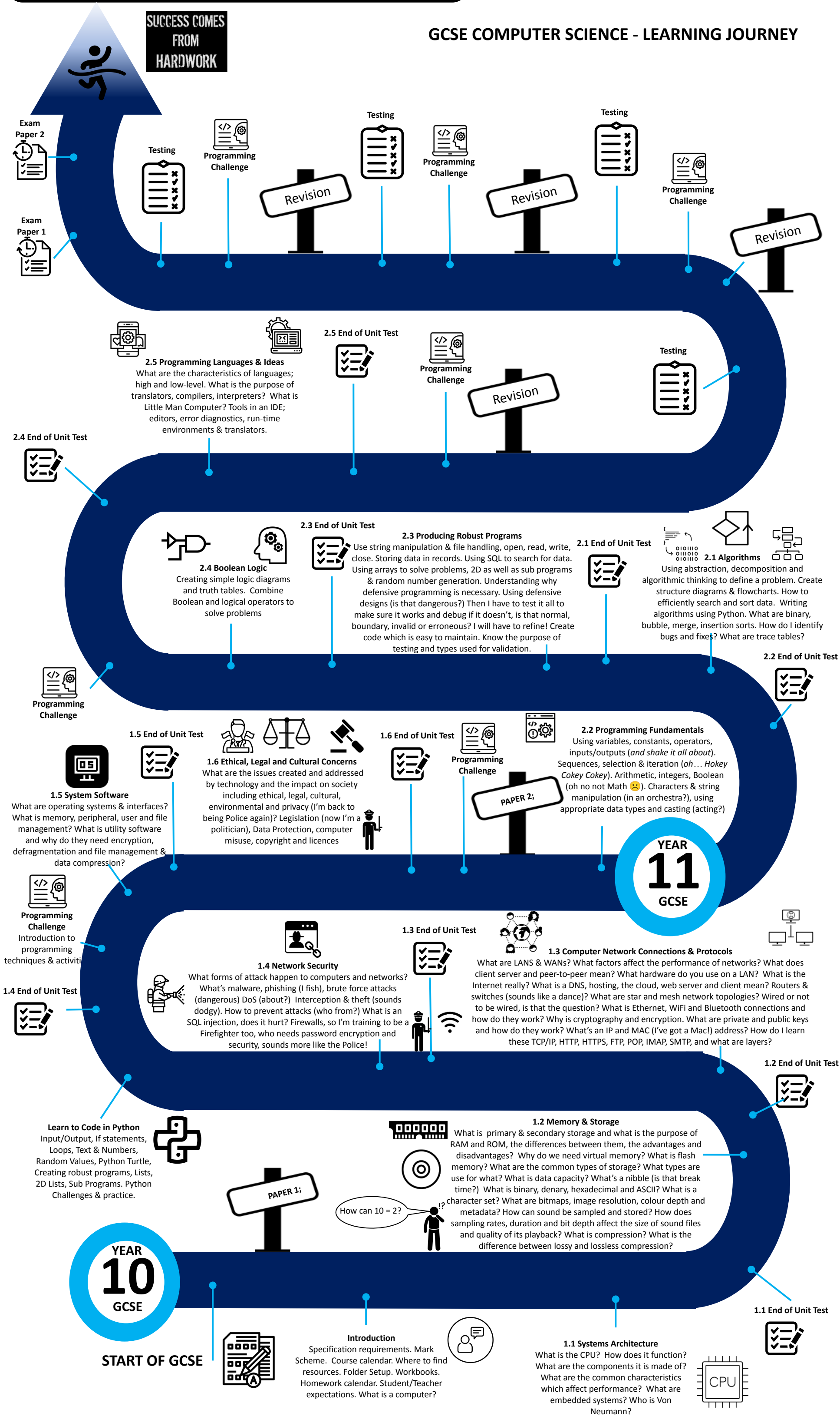
Subject: Computing	Year Group 10 and 11	Key Stage 4
Links to prior learning • Builds on KS3 computing: foundational programming (e.g., Python, Scratch), basic binary, hardware components, and digital literacy. • Prior exposure to algorithms, simple networks, and e-safety is essential for accessing GCSE topics such as systems architecture, programming fundamentals, and network security	Retrieval Practice • Frequent low-stakes quizzes on key terms (e.g., CPU components, types of memory, logic gates). • Regular code tracing, debugging, and algorithm recall exercises. • Spaced retrieval of concepts from earlier in the course, such as binary arithmetic and ethical/legal issues in computing	Adaptive teaching strategies • Scaffolded tasks (e.g., step-by-step algorithm design, differentiated programming challenges). • Visual aids (flowcharts, annotated diagrams) for complex processes like the fetch-execute cycle or network protocols. • Use of formative assessment data to revisit and reteach challenging concepts
Literacy • Explicit teaching and display of technical vocabulary (e.g., abstraction, decomposition, protocol, robust). • Structured writing tasks: explain algorithms, describe impacts of technology, comment code. • Reading and analyzing exam-style questions and mark schemes to	Oracy • Paired and group discussions on ethical, legal, and environmental issues in computing. • Verbal explanation of algorithms, code, and system designs. • Presentations on computing innovations or careers, and debates on topics such as data privacy	Numeracy • Binary and hexadecimal conversions, calculations with data sizes and storage. • Logic gate truth tables and Boolean logic operations. • Algorithm efficiency analysis and use of mathematical reasoning in programming tasks

develop comprehension and written communication skills		
Careers across the Curriculum • Exploration of roles: software developer, cybersecurity analyst, network engineer, data scientist. • Case studies highlighting computing in healthcare, finance, creative industries, and engineering. • Discussion of transferable skills (problem-solving, logical reasoning) relevant to a wide range of careers	Wider Curriculum opportunities • Links to maths (logic, binary, problem-solving), science (data logging, simulations), and PSHE (ethics, online safety). • Interdisciplinary projects, such as using programming for scientific data analysis or creating digital solutions for real-world problems. • Investigation of the societal impact of technology, supporting citizenship and media literacy • Trips to Bletchley Park • Clubs	Super Curriculum Please add link, can't find it
Assessment and feedback Formative assessment given frequently in the form of mini-whiteboard activities, cold-calling, group discussions etc... Summative assessment given once per half term and feedback given on this test and anything of note will be retaught, retrieval starters will be put into		

future lessons to retrieve knowledge that was a weakness.

End of year Mock summarising all knowledge from the year will be sat in the last half term and again feedback will be given on this assessment

GCSE COMPUTER SCIENCE - LEARNING JOURNEY



Curriculum Map cover sheet

Subject: Computing	Year Group 12 and 13	Key Stage 5
Links to prior learning - Builds on GCSE Computer Science (J277) knowledge: programming fundamentals (Python/Java), binary/hexadecimal, basic computer architecture, and understanding of networks, databases, and ethical issues. - Prior experience with algorithms, logic gates, and problem-solving is essential for accessing advanced topics such as computational thinking, data structures, and system architecture	Retrieval Practice - Regular low-stakes quizzes on key terms and concepts (e.g., processor components, Boolean algebra, data structures). - Spaced retrieval and interleaved practice of previously covered topics, including programming constructs, logic, and algorithms. - Code tracing, debugging, and exam-style questions to reinforce long-term retention and exam readiness	Adaptive teaching strategies - Scaffolded tasks for complex topics (e.g., step-by-step algorithm design, differentiated programming challenges). - Use of visual aids (e.g., annotated diagrams of the fetch-decode-execute cycle, flowcharts for algorithms). - Continuous formative assessment to identify and address misconceptions, with tailored interventions
Literacy - Explicit teaching of technical vocabulary (e.g., abstraction, concurrency, normalization, protocol). - Structured writing tasks: code annotation, technical reports, and evaluation of solutions. - Analysis of exam questions and mark schemes to develop precision in written communication	Oracy - Group discussions and debates on legal, moral, and ethical issues in computing. - Verbal explanations of algorithms, system designs, and project choices. - Presentations on research topics, project proposals, and technological innovations	Numeracy - Advanced binary and hexadecimal arithmetic, floating-point representation, and logic circuit design. - Algorithm efficiency analysis (Big O notation), use of truth tables, and Karnaugh maps. - Data analysis and manipulation in programming and database task

Careers across the Curriculum ● Exploration of careers: software engineer, systems architect, cybersecurity analyst, data scientist, AI/machine learning specialist. ● Case studies and guest speakers highlighting computing's role in sectors such as finance, healthcare, engineering, and creative industries. ● Emphasis on transferable skills (problem-solving, analytical thinking, project management) relevant to a wide range of professions	Wider Curriculum opportunities ● Links to mathematics (logic, set theory, problem-solving), science (simulation, data analysis), and business (project management, information systems). ● Interdisciplinary projects, such as using programming for scientific modelling or database design for real-world scenarios. ● Investigation of societal impacts of technology, supporting citizenship and media literacy ● Trips to Bletchley Park ● Clubs	Super Curriculum Please add link, can't find it
Assessment and feedback Formative assessment given frequently in the form of mini-whiteboard activities, cold-calling, group discussions etc... Summative assessment given once per half term and detailed, timely feedback with clear targets for improvement, addressing both theoretical and practical skills, and opportunities for reflection and iterative development will be given on these tests and anything of note will be retaught, retrieval starters will be put into future lessons to retrieve knowledge that was a weakness. End of year 12 Mock summarising all knowledge from		

the year will be sat in the last half term and again feedback will be given on this assessment.		
---	--	--

- **Assessment:**
 - Paper 1: Computer Systems (40%)
 - Paper 2: Algorithms & Programming (40%)
 - Programming Project (20%)

- **Assessment:**
 - Paper 1: Computer Systems (40%)
 - Paper 2: Algorithms & Programming (40%)
 - Programming Project (20%)

A-level COMPUTER SCIENCE - LEARNING JOURNEY

