

# KS5 Computer Science Curriculum Map

Term	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
	<b>Paper 1:</b> Programming recap <b>Paper 2:</b> Data Representation	<b>Paper 1:</b> Programming <b>Paper 2:</b> Hardware	<b>Paper 1:</b> Advanced programming <b>Paper 2:</b> Hardware	<b>Paper 1:</b> Programming tasks <b>Paper 2:</b> Consequences of computing and networks	<b>Paper 1:</b> Abstraction and automation <b>Paper 2:</b> Data structures	<b>Paper 1:</b> Object Orientated programming <b>NEA:</b> Analysis
<b>Year 12</b>	<b>Paper 1 Content</b> 1 to 4: Data Types Data Structures Programming Practice Selection Assignment Iteration Selection Assignment Iteration Nested selection and iteration is also included. 5: Operators Random Relational and Boolean 6 – 7: String Functions	<b>Paper 1 Content</b> 1 and 2: Intro to functions Programming Practice Global Variables 3 and 4: Exception handling 5-7: Consolidation on all aspects of programming learned so far..  <b>Paper 2 Content</b> 1: Internal and external hardware, Recap on what impacts performance, Software types, Operating systems.	<b>Paper 1 Content</b> 1: Trial 1 assessment 2 to 5: Abstract data types Dynamic data types Reading and writing to a file Reading and writing to a file - practice Binary files in Python and practice Dynamic / static data structures 6 and 7: Programming practice  <b>Paper 2 Content</b> 1: Trial 1 assessment	<b>Paper 1 Content</b> 1 to 5: Extended programming set tasks:  <b>Paper 2 Content</b> 2: Communication basics: Serial parallel and rates including latency, Synchronous and asynchronous data transmission, Protocols.  3: Networks: LAN WAN hardware and topologies Client server and peer to peer, Wireless.	<b>Paper 1 Content</b> 1: Abstraction and automation Breaking down problems Abstraction including levels Automation and questions 2 and 3: Finite state machines Intro Transition tables and differences Mealy Machines  <b>Paper 2 Content</b> 1: Queues and stacks	<b>Paper 1 Content</b> 1 and 2: Concepts of OOP and comparisons Class Diagrams and inheritance Practice Polymorphism Agregation methods 3 to 6: OOP practice  <b>NEA</b> 1: Project Preparation 2: NEA Analysis Section

	<p>8: Programming assessment</p> <p><b>Paper 2 Content</b></p> <p>1: Number systems theory, Putting it into practice, Review of number bases, More conversions.</p> <p>2: Program some conversions, Addition and multiplication, Two's complement, Fixed point, Floating Point</p> <p>3: More Floating Point, Underflow and Overflow and Normalisation and Precision, Consolidation.</p> <p>4: ASCII, Error Checking 1, Error Checking 2, Bitmap vs vector graphics.</p> <p>5: Sound, Methods of Compression</p> <p>Methods of Compression</p> <p>6: Coding compression methods</p>	<p>2: Levels of languages including writing assembly language (briefly).</p> <p>3: Logic gates introduction and new gates.</p> <p>4: Boolean algebra including simplification of statements.</p> <p>5: Continue Boolean algebra then Adders and D Type Flip Flop.</p> <p>6: Internal hardware</p> <p>7: Revision for trial 1 exam.</p>	<p>2: Review of assessment, consolidation work.</p> <p>3: Fetch Decode Execute cycle and the performance of processors.</p> <p>4: Processor instruction set including: Interrupts, Machine code instruction set, Direct and indirect addressing, Different operations, Assembly language instruction set</p> <p>5: External hardware: Intro to external devices Laser vs Inkjet printers Harddrives vs SSDs, DVD BluRay and CDs</p> <p>6: Unit revision and assessment</p>	<p>4: IP Addressing, NAT, Public and Private IP, Port forwarding, IPV4 and 6, DHCP and DNS, Structure of packets and packet switching</p> <p>5: Firewalls VPNs and Proxys</p>	<p>2: Queues and stacks including RPN.</p> <p>3: Graphs and trees</p> <p>4: Graphs and Trees including traversal</p> <p>5: Hashes, dictionaries and vectors</p> <p>6: Dijkstra's algorithm</p>	
--	---	--	--	---	--	--

	7: Introduction and Caesar CIPHER, Frequency analysis and Python programming, Transposition and Verman CIPHER, Encryption Essay 8: Unit revision and assessment					
--	--	--	--	--	--	--

Term	Autumn 1	Autumn 2	Spring 1	Spring 2	Summer 1	Summer 2
	Paper 1: trial paper preparation NEA: Design and programming	Paper 1: Exam paper preparation NEA: Programming and testing	Paper 1: Exam paper preparation Paper 2: Networks and regular expressions	Paper 1: Exam paper preparation Paper 2: Databases	Paper 1: Exam paper preparation Paper 2: Functional programming	
<b>Year 13</b>	<b>Paper 1 Content</b> <ul style="list-style-type: none"> <li>Recapping of previous learning about object-oriented programming in Python. <b>(1 lesson)</b></li> <li>Using 2025 skeleton code to prepare for trial 1. Checking student annotations.</li> <li>Class diagrams relating to the skeleton code</li> <li>Section D programming task</li> <li>Section C theory questions relating to skeleton code</li> <li>Section B programming task</li> <li>Section D programming task</li> <li>Section D programming task</li> </ul>	<b>Paper 1 Content</b> <ul style="list-style-type: none"> <li>Provide student with new current skeleton code- Time to become accustomed to preliminary material, skeleton code and add annotations. <b>(2 lessons)</b></li> <li>Class diagrams relating to the skeleton code</li> <li>Section D programming task</li> <li>Section C theory questions relating to skeleton code</li> <li>Section B programming task</li> <li>Section D programming task</li> <li>Section D programming task</li> </ul>	<b>Paper 1 Content</b> <ul style="list-style-type: none"> <li>Recapping earlier learning relating to algorithms and non-OOP programming concepts <b>(1 lesson)</b></li> <li>Recapping of previous learning about object-oriented programming in Python. <b>(1 lesson)</b></li> <li>Section D programming task</li> <li>Section C theory questions relating to skeleton code</li> </ul>	<b>Paper 1 Content</b> <ul style="list-style-type: none"> <li>Section D programming task</li> <li>Section C theory questions relating to skeleton code</li> <li>Section B programming task</li> <li>Section D programming task</li> <li>Section C theory questions relating to skeleton code</li> <li>Section B programming task</li> </ul>	<b>Paper 1 Content</b> <ul style="list-style-type: none"> <li>Recapping earlier learning relating to algorithms and non-OOP programming concepts <b>(1 lesson1)</b></li> <li>Recapping of previous learning about object-oriented programming in Python. <b>(1 lesson1)</b></li> <li>Section D programming task</li> <li>Section C theory questions</li> </ul>	

	<ul style="list-style-type: none"> <li>• Section C theory questions relating to skeleton code</li> <li>• Section B programming task</li> <li>• Section D programming task</li> <li>• Trial 1 prep</li> </ul> <p><b>NEA</b> Design and programming</p>	<ul style="list-style-type: none"> <li>• Section C theory questions relating to skeleton code</li> <li>• Section B programming task</li> <li>• Section D programming task</li> </ul> <p><b>NEA</b> Programming and testing</p>	<ul style="list-style-type: none"> <li>• Section B programming task</li> <li>• Section D programming task</li> <li>• Section C theory questions relating to skeleton code</li> <li>• Section B programming task</li> <li>• Trial 2 prep</li> </ul> <p><b>Paper 2 Content</b> 1: Recap FSM, turing machine, regular languages 2: Maths for regular expressions, Big O. 3: Internet 4: Internet security 5: Encryption - public/private key, Malware and protection 6: TCP-IP Model, Protocols - FTP, SSH, POP3 etc, Questions and confirming knowledge</p>	<ul style="list-style-type: none"> <li>• Section D programming task</li> <li>• Section C theory questions relating to skeleton code</li> <li>• Section B programming task</li> </ul> <p><b>Paper 2 Content</b> 1: Relational databases 2: SQL 3: Big data 4: Client server CRUD REST JSON XML Etc</p>	<p>relating to skeleton code</p> <ul style="list-style-type: none"> <li>• Section B programming task</li> <li>• Section D programming task</li> <li>• Section C theory questions relating to skeleton code</li> <li>• Section B programming task</li> <li>• Section D programming task</li> </ul> <p><b>Paper 2 Content</b> 1: Functional programming</p>	