

GSA Curriculum Map 2022-24: ICT, Business and Computing

Computer Science A-level Curriculum Map

We aim to develop all students into successful and competent programmers who:

- have the desire to develop and apply their analytical, problem-solving, design, and computational thinking skills within programming and Computing as a whole;
- become digitally literate – able to use, and express themselves and develop their ideas through information and communication technology – at a level suitable for the future workplace and as active participants in a digital world;
- understand the impacts of digital technology to the individual, wider society.

	Autumn 1 (HT1)	Autumn 2 (HT2)	Spring 1 (HT3)	Spring 2 (HT4)	Summer 1 (HT5)	Summer 2 (HT6)
Year 12 AQA Computer Science - 7517						
Topic Covered	4.5 Data Representation 4.1 Fundamentals of Programming	4.2 Fundamentals of Data Structures 4.8 Consequences of using Computers 4.1.2 Programming Paradigms 4.1.2.3 Object-oriented programming	4.1 Fundamentals of Programming 4.9 Fundamentals of Communications and Networking	4.6 Fundamentals of Computer Systems 4.1 Fundamentals of Programming	4.10 Fundamentals of Databases 4.1 Fundamentals of Programming	4.7 Organisation and Architecture 4.14 Non Exam Assessment - Analysis
Knowledge developed	4.5 - Data Representation Number systems Understand and convert between different number bases Units of information binary number systems and arithmetic information coding system e.g. ASCII Image representation,, sound and other data compression Encryption methods 4.1 Fundamentals of Programming Data types Programming concepts Arithmetic operations in	4.2 - Data Structures Data structures and abstract data types Graphs, Trees, Dictionaries, Hash Tables, Queues and stacks and vectors. 4.8 Consequences of using Computers Moral, Ethical, Legal, Social and Cultural implications of: <ul style="list-style-type: none"> • Surveillance • Right to privacy • Legal and ethical issues relating to the Internet • Access to 	4.1 Fundamentals of Programming Data types Programming concepts Arithmetic operations in a programming language Relational operations in a programming language Boolean operations in a programming language Constants and variables in a programming language String-handling operations in a programming language Random number generation in a programming language Exception handling Subroutines	4.1 Fundamentals of Programming Data types Programming concepts Arithmetic operations in a programming language Relational operations in a programming language Boolean operations in a programming language Constants and variables in a programming language String-handling operations in a programming language Random number generation in a programming language Exception handling Subroutines (procedures/functions)	4.1 Fundamentals of Programming Data types Programming concepts Arithmetic operations in a programming language Relational operations in a programming language Boolean operations in a programming language Constants and variables in a programming language String-handling operations in a programming language Random number generation in a programming language Exception handling Subroutines (procedures/functions)	4.7 Organisation and Architecture Internal hardware components of a computer Stored program concept Understand the processor and its components Fetch-Execute cycle and the role of registers within it Processor instruction sets Compare addressing modes machine-code/assembly language operations Interrupts Factors affecting processor performance Input and output devices secondary storage devices

	<p>a programming language</p> <p>Relational operations in a programming language</p> <p>Boolean operations in a programming language</p> <p>Constants and variables in a programming language</p> <p>String-handling operations in a programming language</p> <p>Random number generation in a programming language</p> <p>Exception handling</p> <p>Subroutines (procedures/functions)</p> <p>Parameters of subroutines</p> <p>Returning a value/values from a subroutine</p> <p>Local variables in subroutines</p> <p>Global variables in a programming language</p>	<ul style="list-style-type: none"> • encryption keys • Big Data: Impact on society • Data protection • Digital footprint • Legislation <p>4.1.2 Programming Paradigms</p> <p>4.1.2.3 Object-oriented programming</p> <p>programming paradigms</p> <p>procedural and OOP programming in Systems Development.</p>	<p>(procedures/functions)</p> <p>Parameters of subroutines</p> <p>Returning a value/values from a subroutine</p> <p>Local variables in subroutines</p> <p>Global variables in a programming language</p> <p>4.9 Fundamentals of Communications and Networking</p> <p>Communication methods and basics</p> <p>Network topologies, types of networking between hosts.</p> <p>Wireless networking including:</p> <ul style="list-style-type: none"> • Purpose of WIFI • Components of a Network • Wireless Security • CSMA • RTS/CTS • SSID 	<p>Parameters of subroutines</p> <p>Returning a value/values from a subroutine</p> <p>Local variables in subroutines</p> <p>Global variables in a programming language</p> <p>4.6 Fundamentals of Computer Systems</p> <p>relationship between hardware and software system software and application software.</p> <p>Role of the operating system programming languages and their classification into low-and high-level languages.</p> <p>Role of each of and assembler, Compiler and interpreter.</p> <p>Logic gates, boolean expressions, full adder and half adders</p> <p>f Boolean identities and De Morgan's laws to manipulate and simplify Boolean expressions.</p>	<p>Parameters of subroutines</p> <p>Returning a value/values from a subroutine</p> <p>Local variables in subroutines</p> <p>Global variables in a programming language</p> <p>4.10 Fundamentals of Databases</p> <p>conceptual data models and entity relationship modelling</p> <p>Relational databases</p> <p>Database design and normalisation techniques</p> <p>Program with Structured Query Language (SQL)</p> <p>Client server databases</p>	<p>4.14 Non Exam Assessment</p> <p>Produce a programming project and apply the following:</p> <p>produce a clear statement that describes the problem area and specific problem that is being solved/investigated</p> <p>Outline how they researched the problem state for whom the problem is being solved/investigated</p> <p>Provide background in sufficient detail for a third party to understand the problem being solved/investigated</p> <p>Produce a numbered list of measurable, "appropriate" specific objectives, covering all required functionality of the solution or areas of investigation</p> <p>Report any modelling of the problem that will inform the Design stage, for example a graph/network model of Facebook connections or an E-R model.</p>
Skills developed	<p>Number Systems</p> <p>Number Bases</p> <p>Binary Number System</p> <p>Information Coding</p> <p>Systems</p>	<p>Programming</p> <p>Data Manipulation</p> <p>Problem Solving</p> <p>Ethics</p>	<p>Communication</p> <p>Networking</p> <p>Programming</p>	<p>Hardware and Software</p> <p>Classifications of Programming Languages</p> <p>Problem Solving</p>	<p>Problem Solving</p> <p>Programming Languages</p>	<p>Hardware and Software</p> <p>Classifications of Programming Languages</p> <p>Problem Solving</p>
Links to Specification	<p>4.5</p> <p>4.1</p> <p>AQA Specification</p>	<p>4.2</p> <p>4.1</p> <p>4.8</p>	<p>4.1</p> <p>4.9</p>	<p>4.6</p> <p>4.1</p>	<p>4.10</p> <p>4.1</p>	<p>4.7</p> <p>4.14</p>
Flagship Link						
Cross Curriculum		<p>Personal Development -</p> <p>Media Literacy and Digital</p>		<p>A-Level Physics -</p> <p>Electricity and circuit</p>		<p>Design and Technology -</p> <p>Critical evaluation and</p>

Connections		Resilience in Y12 HT6. Links to 4.8 Consequences of using Computers. Cross-curricular links Further Maths: Autumn 1 - Complex Numbers, Further Maths: Spring 1 - Vectors		building in Year 12 HT3. Links to 4.6 Fundamentals of Computer Systems.		testing in HT5/6. Links to 4.14 Non Exam Assessment - Analysis.
Resources to support learning	4.5 4.1	4.2 4.1 4.8	4.1 4.9	4.6 4.1	4.10 4.1	4.7 4.14

	Autumn 1 (HT1)	Autumn 2 (HT2)	Spring 1 (HT3)	Spring 2 (HT4)	Summer 1 (HT5)	Summer 2 (HT6)
Year 13 AQA Computer Science - 7517 (Paper 1)						
Topic Covered	4.7 Organisation and Architecture 4.14 Non Exam Assessment	4.2 Fundamentals of Data Structures 4.14 Non Exam Assessment	4.1 Practice Electronic Assessment Paper 1 4.14 Non Exam Assessment 4.3 Fundamentals of Algorithms	4.4 Theory of Computation 4.14 Non Exam Assessment	Exam Revision Paper 1 & 2	Exam Revision Paper 1 & 2
Knowledge developed	<p>4.7 Organisation and Architecture Internal hardware components of a computer Stored program concept Understand the processor and its components Fetch-Execute cycle and the role of registers within it Processor instruction sets Compare addressing modes machine-code/assembly language operations Interrupts Factors affecting processor performance Input and output devices secondary storage devices</p> <p>4.14 - NEA Design Students are expected to articulate their design in a manner appropriate to the task and with sufficient clarity for a third party to understand how the key aspects of the solution/investigation are structured</p>	<p>4.2 - Data Structures Data structures and abstract data types Graphs, Trees, Dictionaries, Hash Tables, Queues and stacks and vectors.</p> <p>4.14 - NEA - Implementation Students are to complete the technical solution for the NEA program.</p>	<p>4.3 Fundamentals of Algorithms Reverse Polish – infix transformation Linear search Binary search Binary tree search Bubble sort Merge sort Dijkstra's shortest path algorithm</p> <p>4.14 -NEA Complete the technical solution, testing and evaluation of NEA Project</p>	<p>4.4 Theory of Computation Abstraction Information hiding Procedural abstraction Functional abstraction Data abstraction Problem abstraction/reduction Decomposition Composition Automation Finite state machines (FSMs) with and without output Maths for regular expressions Regular expressions Regular language Backus-Naur Form (BNF)/syntax diagrams Comparing algorithms Maths for understanding Big-0 notation Order of complexity Limits of computation Classification of algorithmic problems Computable and uncomputable problems Halting problem Turing machine</p> <p>4.14 -NEA Complete all NEA structure development and outstanding documentation</p>	Retrieval of prior knowledge	

Skills developed	Algorithms Problem Solving Decomposition Abstraction Programming Literacy	Algorithms Problem Solving Decomposition Abstraction Programming Literacy	Algorithms Problem Solving Decomposition Abstraction Programming Data Handling	Algorithms Problem Solving Decomposition Abstraction Programming Literacy	Exam Writing Techniques	Exam Writing Techniques
Links to Specification AQA Specification	4.7 4.14	4.2 4.14	4.3 4.1 4.14	4.11 4.14		
Flagship Link						
Cross Curriculum Connections	Design and Technology - Critical evaluation and testing in HT5/6. Links to 4.14 Non Exam Assessment - Design					
Resources to support learning	4.7 4.14	4.2 4.14	4.3 4.1 4.14	4.11 4.14		