Kings College Computer Science Department: GCSE Curriculum Overview 2026-27

Curriculum Intent: The Computer Science Department offers a broad curriculum that does the following:

- Explore how Computer Science has an impact on our day-to-day world.
- Understand how Computer Science has grown and developed to be the industry of the future.
- Develop skills in Computer Science to allow access to a broad range of careers.

Our aim is to create inquisitive and critical computer literate students who are able to navigate and understand the power and impact that Computing has on the modern world. We seek to empower students to interpret the code, programs and hardware and understand how these are constructed. We offer pupils a curriculum which is tailored to GCSE success while consistently using modern contemporary resources which are relevant to their lives.

We expose students to a range of wider cultural and technical knowledge from Computing industries including Architecture, Networks, Programming (Python) and the Ethics in Computing.

Year 9 gets 3 lessons per fortnight cycle.

Year 9	**Term 1 (September – December)**		**Term 2 (January – March)**		**Term 3 (April – July)**	
	Module 1(3rd Sep – 15th Oct) [6.5 Weeks / 9 lessons] Cyber security Programming fundamentals	**Module 2 (2nd Nov – 18th Dec) [7 weeks, 10 Lessons] Programming fundamentals**	**Module 3 (4th Jan – 12th Feb) [6 weeks, 9 lessons] Programming Fundamentals**	**Module 4 (22nd Feb – 24th Mar) [5 weeks, 7 lessons] Programming Project Revision**	**Module 5 [7 weeks 10 lessons] Programming Project Revision**	**Module 6 (GCSE) [8 weeks 12 Lessons] GCSE Computer science Systems Architecture**
Content	**Cyber Security** • Computer Laws (1) • Social Engineering (1) • Types of Cyber Attacks (1) • Types of Malwares (1) • Protection Mechanisms (1) Consolidate (1) **Dart(1)** **Artificial Intelligence (AI)** • Making effective use of AI(1) **[9/9]**	**Programming fundamentals** • Algorithmic Thinking (1) • Pseudocode (1) • Flow Charts (1) **Consolidate**(1) • Intro to Python (1) o print() & Data types o input() & Variables • Operators (1) November Assessment (1) (30 Nov – 4th Dec) • Sequence & Selection (1) • Iteration - for and while (1) • Basic Strings (1) **[10/10]**	• Arrays (1) • Sub-Programs (1) • File IO (1) **Dart(1)** Group Programming project (5) **[4/9]**	• Group Programming project • Individual Programming Project • Revision **[7/7]**	• Individual Programming Project • Revision	• Computer Systems Overview • Architecture of the CPU • CPU Performance • Embedded systems • Primary Storage • Secondary Storage
Feedback Points	End of unit assessment.	November Assessment	End of unit assessment.	End of unit assessment.	End of Year assessment.	End of year assessment.
Key Questions	1. How do we keep data safe? 2. How are we vulnerable to attack? 3. What types of attack are there? 4. AI. Is it safe?	1. An approach to problem solving. 2. How can we make programs more reliable? 3. How do we use Python to program? 4. What sort of data can it use? 5. What can we achieve with this software?	1. Using advanced techniques to create more useful programs.	1. Laws around computer science? 2. Do we have freedom of information? 3. How does Computing tech impact the environment? 4. How can fake news impact our lives? 5. Daling with fake news?	1. Practice programming a useful piece of software. Plan, write, debug, release a useful piece of software.	1. What is Computer Science? 2. What is the structure of a CPU based on Von Neumann's design? 3. Describe the purpose and function of the CPU. 4. What factors affect CPU performance? 5. Why both Primary and Secondary storage is needed.
Direct Vocab Instruction	Malware, cyber	Algorithm, Decomposition, Sequence, Variable, Selection	Iteration, Array sub-program	Law, Regulation, Ethics,		Architecture, Embedded
Standardised	Turing Lab Creative Python	Turing Lab Creative Python	Turing Lab Creative Python	Turing Lab Creative Python	Turing Lab Creative Python	Turing Lab Creative Python

Kings College Computer Science Department: GCSE Curriculum Overview 2026-27

Homework	Lessons based worksheets	Lessons based worksheets	Lessons based worksheets	Lessons based worksheets	Lessons based worksheets	Lessons based worksheets

Kings College Computer Science Department: GCSE Curriculum Overview 2026-27

Curriculum Intent: The Computer Science Department offers a broad curriculum that does the following:

- Explore how Computer Science has an impact on our day-to-day world.
- Understand how Computer Science has grown and developed to be the industry of the future.
- Develop skills in Computer Science to allow access to a broad range of careers.

Our aim is to create inquisitive and critical computer literate students who are able to navigate and understand the power and impact that Computing has on the modern world. We seek to empower students to interpret the code, programs and hardware and understand how these are constructed. We offer pupils a curriculum which is tailored to GCSE success while consistently using modern contemporary resources which are relevant to their lives.

We expose students to a range of wider cultural and technical knowledge from Computing industries including Architecture, Networks, Programming (Python) and the Ethics in Computing.

Year 10	**Term 1 (September – December)**		**Term 2 (January – March)**		**Term 3 (April – July)**	
	Module 1 (3rd Sep – 15th Oct) [6.5 Weeks / 9 lessons] Data Representation, Memory and Storage	**Module 2(2nd Nov – 18th Dec) [7 weeks, 11 lessons] Computer Networks, Connections and Protocols.**	**Module 3(4th Jan – 12th Feb) [6 Weeks, 13 Lessons] [6 Weeks, 15 Lessons] Network Security**	**Module 4(22nd Feb – 24th Mar) [5 weeks, 7 Lessons] Systems Software Impacts of digital technology**	**Module 5 (12th Apr – 28th May) [7 Weeks, 17 Lessons] Algorithms**	**Module 6 (07th Jun – 23rd Jul) [6 weeks excluding work experience week, 15 lessons] Python fundamentals Programming experience**
Content	**Data Storage units (1)** **Number Systems (5)** • Counting • Bin < > Hex • Den < > Bin • Den < > Hex • Dart **Binary Arithmetic (3)** • Binary Addition • Binary Shifts • Dart **Data Files (5)** • Text files & • Sound Files • Image Files • Data Compression Module test **[13/15]**	**Networking (8)** • LANs, Wans • Network Addressing • Networking Hardware • Network Topology • Wireless, Wired networks • Network Performance • Networking Standards • Network Layers • Dart Revision November Mocks (16th – 20th) (1) • Network Protocols • The internet **[13/17]**	**Threats to computer systems and networks (7)** • Malware • Social engineering • Brute Force Attack, DoS and DDoS Attack • Data interception and theft • QL data injection • Dart **Prevention methods (3)** • Pen Testing • Firewalls • User access Levels • Anti-Malware • Encryption • Physical security Module test **[10/14]**	**Operating System software (7)** • User interface • Memory management • Multitasking • User Management • File management • Peripheral management • Dart **Utility System Software (4)** • Encryption • Defragmentation • Data compression • Dart **Impacts of Digital Technology (4)** • Ethical & Legal • Privacy & Cultural • Environmental Issues Module test **[16/17]**	**Computational Thinking Concepts (3)** • Decomposition (1) • Abstraction (1) • Algorithmic thinking (1) • dart **Design, Create & refine Algos(5)** • Pseudocode • ERL • Flowcharts • Trace tables • Dart **Searching Algorithms (2)** • Linear search (1) • Binary Search (1) **Sorting Algorithms (4)** • Bubble sort • Insertion sort • Merge sort • Dart Module test **[14/17]**	**Programming Fundamentals 1 (10)** • Output, Data types & Math Operators • Input & Variables • Comparison & Boolean Operators • Selection - if & match • Iteration – for & while • String operations • Putting it all together 1 • Arrays (1D & 2D) • Functions & procedures • Basic File I/O • Putting it all together 2 • Programming practice 1 • Programming practice 2 Module test **[13/15]**
Feedback Points	End of unit assessment. (1)	November Assessment (1)	Mid unit and end of unit assessment.	Mid unit and end of unit assessment.	Mid unit and end of unit assessment.	End of Year Assessment
Key Concepts	• Converting between units • Counting in Binary • Converting between the number systems. • Calculating file storage needs. • Digitising sound.	• Type and purpose of networking hardware • Networking media • Features and advantages of WANs and LANs • MAC and IP addressing • Commonly used	• Identify main threats to computer systems. • Identify countermeasures to threats.	• Purpose of an OS • Types and purpose of utility software • How digital technology impact our lives.	1. Techniques used for problem solving. 2. Techniques for developing algorithms 3. Basic algorithms for searching through data • 4. Basic algorithms for sorting data	• The different generations of programming language? • The differences between high-level and low-level languages. • The benefits and drawbacks of high and low level languages.

	• How images are constructed. • Types of compression and need for compression.	communications protocols and how they enable computer communication.				
DVI	Embedded, Characteristic, Pixel, metadata, compress	Network, Media, Protocol, Standard	Malware, Utility, Interception, encryption	Operating System, Interface, Peripheral, Driver, Ethics, defragment, multitask, peripheral	Abstraction, Decomposition, Algorithm, Pseudocode	Sequence, Selection, iteration
Standardised Homework	1. TuringLab module review 2. Homework sheet	1. TuringLab module review 2. Homework sheet	1. TuringLab module review 2. Homework sheet	1. TuringLab module review 2. Homework sheet	1. TuringLab module review 2. Homework sheet	1. TuringLab module review 2. Homework sheet

Year 11	**Term 1 (1st Sep – 18th Dec) 1st &2nd Sep are insets.**		**Term 2 (January – March)**		**Term 3 (April – July)**	
	Module 1 (3rd Sep – 15th Oct) [6.5 Weeks / 15 lessons] Algorithms	**Module 2(2nd Nov – 18th Dec) [7 weeks, 17 lessons] Defensive Design Testing Boolean Logic**	**Module 3(4th Jan – 12th Feb) [6 Weeks, 14 Lessons] Computer Languages and IDEs Practical Programming**	**Module 4(22nd Feb – 24th Mar) [5 weeks, 11 Lessons] Practical Programming Revision**	**Module 5 (12th Apr – 31st May) [7 weeks, 8 Lessons] GCSE Computer Science Rev**	
Content	**Computational Thinking (3)** • Abstraction • Decomposition • Algorithmic thinking **Design, Create & refine Algos (4)** • Pseudocode & ERL • Flowcharts • Trace tables **Dart** **Searching Algorithms (2)** • Linear search • Binary Search **Sorting Algorithms (4)** • Bubble sort • Insertion sort • Merge sort **Dart** **[13/15]**	**Defensive Design(2):** • Anticipating misuse & Input validation • Authentication & Maintainability **Testing (4)** • Types of testing & Selecting and using suitable Test data • Syntax Vs Logic Errors & Refining algos • Create/Complete test plan. **revision (1)** **November Mocks (16th – 20th) (2)** **Boolean Logic (4)** • Basic Logic operators symbols & truth tables • Logic diagrams – 3 layers deep • Truth table to logic diagram **Language types (1)** • High & Low level Languages **Translators (1)** • Compilers and Interpreters **Features Provided by IDEs (3)** • Editor, Error diagnostics, Translators, Runtime environment • Debugging **Dart** **[16/17]**	**Programming Fundamentals 1 (10)** • Output, Data types & Math Operators • Input & Variables • Comparison, Boolean Operators • Selection - if & match • Iteration – for & while • Strings and string operations • Arrays (1D & 2D) • Functions & procedures • Basic File I/O • Putting it all together **Programming Fundamentals 2 (2)** • SQL 1 - Fields, Records & Queries • SQL 2 – More queries. Programming Practice Programming Practice **[14/14]**	Revision • Paper 1 • Paper 2 • Programming practice. February Mocks (1st Mar – 12th Mar) • Paper 1 • Paper 2 • Programming practice. • Paper 2 • Paper 1 • Programming practice. **[11/11]**	• Paper 1 • Paper 2 • Programming practice. • Paper 1 • Paper 2 • Programming practice. • Paper 1 • Programming practice. • Programming practice. **Monday 10th May (week 5)** **GCSE Paper 1** **Wednesday 19th May (week 5)** **GCSE Paper 2** **[9/9]**	
Feedback	Dart	Dart	Mid unit and end of unit assessment.	February mocks		

Points	End of unit assessment.	End of Unit assessment.				
Key Concepts	1. Techniques used for problem solving. 2. Techniques for developing algorithms 3. Basic algorithms for searching through data 4. Basic algorithms for sorting data	1. Writing reliable programs. 2. Structured way of testing data. 3. Connecting Boolean operators, symbols diagrams and truth tables. 4. Classification of programming languages	1. Basic building blocks for writing programs. 2. Basic database queries.			
Direct Vocab Instruction	Abstraction, Decomposition, Algorithm, Pseudocode	Authenticate, Syntax, Validate, refine, operator, compile, translate, Run Time	Variable, Selection, Iteration, Function Procedure, record			
Standardised Homework	Turing Lab module review	Turing Lab module review Programming task	Turing Lab module review Programming task			

Provision for at least 20 hours of practical programming

1. Python Programming
2. Plan, Develop, Test, Refine
3. Mini programming tasks provided throughout the year